



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL ECUADOR
SEDE AMBATO

ESCUELA DE INGENIERÍA DE SISTEMAS

Tema:

“DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL DE
EXTENSIÓN UNIVERSITARIA E INFORMACIÓN ESTADÍSTICA PARA LA
DIRECCIÓN DE ESTUDIANTES DE LA PUCESA”

Autor:

DANILO EDUARDO RUBIO PUIG

Director de Disertación:

Ing. Jorge Chávez Andrade

Ambato – Ecuador
Septiembre 2007

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE AMBATO

ESCUELA DE INGENIERÍA DE SISTEMAS

Tema:

DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL DE
EXTENSIÓN UNIVERSITARIA E INFORMACIÓN ESTADÍSTICA PARA LA
DIRECCIÓN DE ESTUDIANTES DE LA PUCESA

Autor:

DANILO EDUARDO RUBIO PUIG

Director de Disertación:

Ing. Jorge Chávez Andrade

Ambato – Ecuador
Septiembre 2007

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Danilo Eduardo Rubio Puig portador de la cédula de ciudadanía No. 180300888-5 declaro que los resultados obtenidos en la investigación que presento como informe final, previo la obtención del título de Ingeniero en Sistemas son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo de investigación propuesto, así como de la redacción de este documento, son y serán de exclusiva responsabilidad de mi persona.

Danilo Eduardo Rubio Puig

C.I. 180300888-5

AGRADECIMIENTO

Debido a la importancia que el desarrollo de este proyecto abarca, es meritorio hacer constar mi más profundo agradecimiento a los profesionales, administrativos, familiares y amigos que con su apoyo y conocimientos hicieron posible la realización del mismo. Mención especial merecen Ing. Telmo Viteri e Ing. Santiago Acurio, Directores de la Escuela de Sistemas, Ing. Diego Santacruz, Administrador del Centro de Cómputo de la Universidad. e Ing. Jorge Chávez Andrade, Asesor de Tesis, Ing. Msc. María Elena Sandoval, Directora de Estudiantes, Ing. Rosita Palacios, Directora Administrativa; quienes compartieron no solo conocimientos, sino también valores y amistad que fortalecen la formación de todo buen profesional.

Además no puedo dejar de agradecer a todas aquellas personas que formaron parte de mi carrera universitaria directores, profesores, compañeros y amigos, quienes me brindaron su apoyo y amistad, contribuyendo de una u otra forma en mi formación profesional.

Mi sincero deseo es que este proyecto sirva para fortalecer y enaltecer el prestigio bien ganado que posee la Universidad.

D.E.R.P.

DEDICATORIA

El presente trabajo está dedicado muy especialmente a mis padres, cuyo amor y apoyo motivaron mi interés en recompensar todo su esfuerzo realizado al formarme en una persona de bien; además, a mis hermanos que siempre han sabido brindarme su soporte y cariño.

INDICE GENERAL

| | |
|---|-----------|
| DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD | 2 |
| AGRADECIMIENTO | IV |
| DEDICATORIA | V |
| INDICE GENERAL | VI |
| 1 CAPÍTULO I. PROYECTO DE INVESTIGACIÓN | 1 |
| 1.1 Antecedentes..... | 1 |
| 1.2 Significado del Problema | 2 |
| 1.3 Definición del Problema..... | 2 |
| 1.4 Planteamiento del Tema..... | 3 |
| 1.5 Delimitación del Tema | 3 |
| 1.6 Hipótesis..... | 4 |
| 1.7 Objetivos | 4 |
| 1.7.1 Objetivo General..... | 4 |
| 1.7.2 Objetivos Específicos | 5 |
| 1.8 Metodología de Trabajo..... | 5 |
| 1.9 Justificación..... | 6 |
| 2 CAPÍTULO II. MARCO TEÓRICO | 8 |
| 2.1 Dirección de Estudiantes de la PUCESA | 8 |
| 2.1.1 Actividades de Extensión Universitaria | 8 |
| 2.1.2 Políticas de Extensión Universitaria..... | 8 |
| 2.1.3 Procedimiento para solicitar horas de Extensión Universitaria en el Departamento de Pastoral Universitaria y la Dirección de Estudiantes..... | 11 |

| | | |
|------------|---|-----------|
| 2.2 | Ingeniería de Software | 13 |
| 2.2.1 | Definición..... | 13 |
| 2.2.2 | Características del software..... | 13 |
| 2.2.3 | Aplicaciones de Software..... | 17 |
| 2.2.4 | Calidad de Software..... | 20 |
| 2.2.4.1 | Definición de Calidad | 20 |
| 2.2.4.2 | Definición de Calidad de Software..... | 22 |
| 2.3 | Bases de Datos | 24 |
| 2.3.1 | Definición..... | 24 |
| 2.3.2 | Organización de una Base de Datos | 25 |
| 2.3.3 | Abstracción de la Información | 28 |
| 2.3.3.1 | Nivel físico | 28 |
| 2.3.3.2 | Nivel conceptual..... | 29 |
| 2.3.3.3 | Nivel de visión | 30 |
| 2.3.4 | Modelos de Base de Datos..... | 30 |
| 2.3.4.1 | Modelos lógicos basados en objetos..... | 31 |
| 2.3.4.1.1 | Modelo Entidad-Relación | 31 |
| 2.3.4.2 | Modelos lógicos basados en registros | 32 |
| 2.3.4.2.1 | Modelo jerárquico | 32 |
| 2.3.4.2.2 | Modelo de red..... | 33 |
| 2.3.4.2.3 | Modelo relacional..... | 33 |
| 2.3.4.3 | Modelos físicos de datos..... | 34 |
| 2.3.5 | Sistema de Gestión de Base de Datos (SGBD)..... | 34 |
| 2.4 | Base Datos MySql | 35 |
| 2.4.1 | Introducción | 35 |
| 2.4.2 | Comandos en MySQL | 37 |
| 2.4.3 | Conectar al Servidor MySQL..... | 38 |
| 2.4.4 | Creación de Base de Datos | 39 |
| 2.4.5 | Carga y Recuperación de Registros de una Tabla..... | 41 |
| 2.4.6 | Operadores Relacionales | 42 |
| 2.4.7 | Borrar Registros de una Tabla | 43 |
| 2.4.8 | Modificar Registros de una Tabla | 45 |
| 2.4.9 | Clave Primaria de una Tabla | 45 |
| 2.4.10 | Campo entero con autoincremento | 46 |
| 2.4.11 | Tipos de datos | 48 |
| 2.4.11.1 | Tipos de Datos Texto..... | 50 |

| | | |
|------------|---|-----------|
| 2.4.11.2 | Tipos de Datos Numéricos..... | 51 |
| 2.4.11.3 | Tipos de Datos Fechas y Horas | 53 |
| 2.4.12 | Valores por defecto..... | 55 |
| 2.4.13 | Valores inválidos | 56 |
| 2.4.14 | Operadores Lógicos | 58 |
| 2.5 | Programación Web..... | 60 |
| 2.5.1 | Lenguajes de Programación Web..... | 61 |
| 2.5.1.1 | Lenguajes del Lado del Cliente | 64 |
| 2.5.1.2 | Lenguajes del Lado del Servidor | 69 |
| 2.5.1.3 | Lenguajes del Lado Cliente-Servidor..... | 76 |
| 2.6 | Lenguaje de Programación PHP | 78 |
| 2.6.1 | Aplicaciones y Ventajas | 80 |
| 2.6.2 | Introducción al Lenguaje PHP..... | 82 |
| 2.6.3 | Variables | 83 |
| 2.6.4 | Tipos de Datos | 83 |
| 2.6.4.1 | Enteros..... | 84 |
| 2.6.4.2 | Números en punto flotante..... | 84 |
| 2.6.4.3 | String | 85 |
| 2.6.4.4 | Arrays | 85 |
| 2.6.4.5 | Objetos..... | 86 |
| 2.6.4.6 | Juggling..... | 86 |
| 2.6.5 | Operadores | 87 |
| 2.6.5.1 | Aritméticos..... | 87 |
| 2.6.5.2 | Comparación | 88 |
| 2.6.5.3 | Lógicos | 88 |
| 2.6.6 | Estructuras de Control | 89 |
| 2.6.6.1 | Estructura IF..... | 89 |
| 2.6.6.2 | Estructura SWITCH..... | 90 |
| 2.6.6.3 | Bucle FOR..... | 91 |
| 2.6.6.4 | Bucle WHILE..... | 92 |
| 2.6.6.5 | Bucle DO..WHILE..... | 92 |
| 2.6.6.6 | Break | 92 |
| 2.6.6.7 | Continue | 93 |
| 2.6.7 | Funciones en PHP | 93 |
| 2.6.8 | PHP y MySQL | 94 |
| 2.6.8.1 | Conexión con MySQL | 94 |

| | | |
|------------|---|------------|
| 2.6.8.2 | Consultas a la base de datos..... | 98 |
| 2.6.8.3 | Otras Funciones de PHP y MySQL..... | 99 |
| 2.6.9 | Cookies..... | 112 |
| 2.6.10 | Variables de Sesión..... | 113 |
| 2.6.11 | Clases | 114 |
| 3 | CAPÍTULO III. DESARROLLO DEL PROYECTO | 116 |
| 3.1 | Análisis del Sistema | 116 |
| 3.1.1 | Identificación de las necesidades..... | 116 |
| 3.1.2 | Estudio de Viabilidad | 117 |
| 3.1.3 | Análisis Técnico | 119 |
| 3.1.4 | Análisis de Requisitos..... | 120 |
| 3.1.4.1 | Especificaciones del Sistema | 121 |
| 3.1.5 | Análisis Estructurado..... | 123 |
| 3.1.5.1 | Diagrama de Flujo de Datos Nivel 0 | 123 |
| 3.1.5.2 | Diagrama de Flujo de Datos Nivel 1 | 124 |
| 3.2 | Diseño del Sistema | 125 |
| 3.2.1 | Diseño de la Base de Datos | 126 |
| 3.2.2 | Diseño del Sitio Web y la Interfaz | 135 |
| 3.3 | Desarrollo del sistema | 140 |
| 3.3.1 | Creación de la Base de Datos | 140 |
| 3.3.2 | Descripción y Codificación de Clases..... | 150 |
| 3.3.2.1 | Class conexion | 150 |
| 3.3.2.2 | Class as-diagrams: Gráficos de barras con HTML y CSS..... | 151 |
| 3.3.2.2.1 | Instalando as-diagrams..... | 151 |
| 3.3.2.2.2 | Usando la clase CAsBarDiagram | 152 |
| 3.3.2.2.3 | CAsBarDiagram métodos y variables | 154 |
| 3.4 | Pruebas e Instalación..... | 158 |
| 3.4.1 | Pruebas | 158 |
| 3.4.2 | Instalación..... | 159 |
| 4 | CAPÍTULO IV. VERIFICACIÓN Y VALIDACIÓN DE RESULTADOS | 162 |
| 4.1 | Verificación..... | 162 |

| | | |
|------------------------------|--|------------|
| 4.2 | Validación | 163 |
| 4.3 | Conclusiones | 164 |
| 4.4 | Recomendaciones | 164 |
| ANEXOS | | 166 |
| A. MANUAL TÉCNICO | | 166 |
| 1 INTRODUCCIÓN | | 169 |
| 1.1 | PROPÓSITO | 169 |
| 1.1.1 | PROPÓSITO DEL DOCUMENTO | 169 |
| 1.1.2 | AUDIENCIA A LA QUE VA DIRIGIDO | 169 |
| 1.2 | ALCANCE..... | 170 |
| 1.2.1 | IDENTIFICACIÓN DEL PRODUCTO MEDIANTE UN NOMBRE..... | 170 |
| 1.2.2 | QUÉ HACE Y QUÉ NO HACE EL PRODUCTO | 170 |
| 1.2.3 | APLICACIONES DEL SOFTWARE: BENEFICIOS, OBJETIVOS Y METAS..... | 171 |
| 1.3 | DEFINICIÓN, ACRÓNIMOS Y ABREVIATURAS..... | 171 |
| 1.4 | REFERENCIAS..... | 171 |
| 1.5 | VISIÓN GENERAL | 171 |
| 1.5.1 | DESCRIPCIÓN DEL CONTENIDO DEL RESTO DEL DOCUMENTO | 171 |
| 1.5.2 | ORGANIZACIÓN DEL DOCUMENTO | 172 |
| 2 DESCRIPCIÓN GENERAL | | 172 |
| 2.1 | PERSPECTIVA DEL PRODUCTO | 172 |
| 2.1.1 | INDICAR SI ES UN PRODUCTO INDEPENDIENTE O PARTE DE UN SISTEMA MAYOR | 172 |
| 2.1.2 | INTERFACES DE SISTEMAS | 172 |
| 2.1.3 | LIMITACIONES DE MEMORIA | 173 |
| 2.1.4 | OPERACIONES..... | 173 |
| 2.1.4.1 | MODO DE OPERACIÓN DE LOS DISTINTOS GRUPOS DE USUARIOS..... | 173 |
| 2.1.4.2 | FUNCIONES RESPALDOS DEL PROCESAMIENTO DE DATOS | 174 |
| 2.1.4.3 | OPERACIONES DE BACKUP Y RECUPERACIÓN | 174 |
| 2.1.5 | REQUERIMIENTOS PARA ADAPTARSE A LA UBICACIÓN | 175 |

| | | |
|------------|--|------------|
| 2.1.5.1 | DATOS O SECUENCIAS DE INICIALIZACIÓN ESPECÍFICOS DE CUALQUIER LUGAR, MODO DE OPERACIÓN | 175 |
| 2.1.5.2 | CARACTERÍSTICAS QUE DEBEN SER MODIFICADAS PARA UNA INSTALACIÓN EN PARTICULAR | 176 |
| 2.2 | FUNCIONES DEL PRODUCTO | 177 |
| 2.3 | CARACTERÍSTICAS DEL USUARIO | 177 |
| 2.4 | RESTRICCIONES | 177 |
| 2.5 | SUPOSICIONES Y DEPENDENCIAS | 178 |
| 2.6 | REQUISITOS PARA FUTURAS VERSIONES DEL SISTEMA..... | 178 |
| 3 | REQUISITOS ESPECÍFICOS | 178 |
| 3.1 | REQUISITOS DE INTERFAZ EXTERNA | 178 |
| 3.1.1 | INTERFACES DE USUARIO | 178 |
| 3.1.2 | INTERFACES DE SOFTWARE | 179 |
| 3.2 | REQUISITOS FUNCIONALES | 179 |
| 3.2.1 | FLUJOS DE INFORMACIÓN | 179 |
| 3.2.1.1 | DIAGRAMA DE FLUJO DE DATOS NIVEL 0 | 179 |
| 3.2.1.2 | DIAGRAMA DE FLUJO DE DATOS NIVEL 1 | 180 |
| 3.3 | REQUISITOS DE RENDIMIENTO..... | 180 |
| B. | MANUAL DE USUARIO | 180 |
| | BIBLIOGRAFÍA | 226 |

1 CAPÍTULO I. PROYECTO DE INVESTIGACIÓN

1.1 Antecedentes

El Nivel de desarrollo actual de la PUCESA hace necesario la implementación de nuevas herramientas que faciliten las actividades del área administrativa.

La carencia de adecuados sistemas informáticos de control académico, genera relativa lentitud en los procesos y procedimientos, todo lo cual imposibilita una dinamización de los mismos a nivel administrativo.

Es indispensable que toda organización educativa priorice dentro de su estructura la implementación de sistemas informáticos de control que permitan una verdadera interacción que a través del manejo estadístico de la información permita desarrollar ciertos esquemas en el ámbito educativo de su competencia.

La Dirección de estudiantes de la PUCESA no cuenta con un sistema práctico que facilite el control en el área de actividades de Extensión Universitaria lo que ocasiona relativa demora en dar respuesta a las necesidades de los estudiantes.

Por otra parte se hace necesario contar con varios tipos de información como reportes gráficos estadísticos que permitan una mejor comprensión, interpretación y análisis de los datos de los estudiantes.

1.2 Significado del Problema

Carencia de los procedimientos y herramientas que optimicen los procesos a cargo de la Dirección de Estudiantes de la PUCESA en lo que respecta a las Actividades de Extensión Universitaria y obtención de información de los estudiantes.

1.3 Definición del Problema

¿Con qué tipo de sistema de control sobre Actividades de Extensión Universitaria se cuenta actualmente en la Dirección de Estudiantes de la PUCESA?

¿Qué consecuencias trae el retardo en la respuesta a las necesidades de los estudiantes de la PUCESA en relación a sus actividades de Extensión Universitaria?

¿Cómo afecta la falta de información en forma de reportes estadísticos sobre los datos de los estudiantes de la PUCESA?

1.4 Planteamiento del Tema

Diseño e implementación del sistema de control de Extensión Universitaria e información estadística para la Dirección de Estudiantes de la PUCESA.

1.5 Delimitación del Tema

El presente estudio está orientado al diseño e implementación de un sistema informático que facilite a la Dirección de Estudiantes de la Pontificia Universidad Católica del Ecuador – Sede Ambato el control sobre la Ejecución de Actividades de Extensión Universitaria, así como también la obtención de información a manera de gráficos estadísticos que faciliten su análisis; el mismo que será desarrollado en un tiempo de siete meses en el período 2006-2007.

Se diseñará la interfaz Web de la aplicación, así como también la base de datos que contendrá la información de Extensión Universitaria de los Alumnos.

Se diseñarán reportes de acuerdo a las necesidades de la Dirección de Estudiantes de la PUCESA en lo que respecta a las actividades de Extensión Universitaria.

Se investigará sobre las herramientas compatibles con el hardware y software con que cuenta el actual servidor Web de la PUCESA; es decir, librerías

gráficas de uso con el lenguaje de programación PHP, que puedan ser incorporadas en un servidor Sun Solaris. Estas herramientas permitirán la representación de la información de los datos generales de los estudiantes a manera de gráficos estadísticos y serán necesarias para el desarrollo del sistema en este aspecto.

1.6 Hipótesis

Con la implementación de un Sistema Informático de Control en la Dirección de Estudiantes de la PUCESA, se facilitará a la Administración el control y aprobación de Ejecución de las Actividades de Extensión Universitaria y la obtención de información estadística de los Estudiantes.

1.7 Objetivos

1.7.1 Objetivo General

Desarrollar un Sistema Informático que permita facilitar a nivel administrativo el control y aprobación de Ejecución de Actividades de Extensión Universitaria y obtención de información a nivel estadístico de los estudiantes de la PUCESA.

1.7.2 Objetivos Específicos

- Realizar un Sistema Informático a nivel administrativo para Facilitar el control y aprobación de Ejecución de Actividades de Extensión Universitaria de los estudiantes de la PUCESA.
- Implementar un Sistema Informático que permita reducir el tiempo de respuesta a las solicitudes y necesidades de los estudiantes en el área de Ejecución de Actividades de Extensión Universitaria de los Estudiantes de la PUCESA.
- Generar un Sistema Informático para facilitar un análisis de información a nivel estadístico de los estudiantes por parte de la Dirección de Estudiantes de la PUCESA.

1.8 Metodología de Trabajo

La investigación estará enmarcada dentro de los siguientes ámbitos:

- La investigación bibliográfica, que permite apoyarse en fuentes de carácter documental para el desarrollo del sistema.
- La investigación de campo, que facilita la recopilación de información de las necesidades de los usuarios a nivel administrativo de la Dirección de

Estudiantes de la PUCESA en forma directa para poder plasmarlos en el mismo.

La investigación se basará en los paradigmas que se describen a continuación:

El Empirista, que permite obtener todos los elementos necesarios para la solución del problema, en función del conocimiento adquirido y la práctica.

El Pragmático, que permite comprobar que mediante la implementación de un sistema de control se agilizarán los procesos ya mencionados por parte de la Dirección de Estudiantes de la PUCESA.

1.9 Justificación

El presente tema de investigación tiene relevancia a nivel local ya que permitirá mejorar la atención a los requerimientos de los estudiantes en lo que respecta a sus actividades de Extensión Universitaria por parte de la Dirección de Estudiantes de la PUCESA.

Además tiene relevancia a nivel nacional, pues el mismo podrá servir de guía para el manejo de información de otras sedes de la Pontificia Universidad Católica del Ecuador.

Se cuenta con los conocimientos, herramientas, guías y medios económicos necesarios para la realización e implementación de la solución al problema planteado.

2 CAPÍTULO II. MARCO TEÓRICO

2.1 Dirección de Estudiantes de la PUCESA

2.1.1 Actividades de Extensión Universitaria

Son todas aquellas actividades tendientes a:

- La proyección social de la Universidad.
- La promoción y difusión de la cultura.
- Los servicios a terceros.
- Las relaciones de la Universidad con otras instituciones de educación superior nacionales y extranjeras.

(Tomado del Reglamento Académico de la PUCE, Art. 3)

2.1.2 Políticas de Extensión Universitaria

- La Extensión Universitaria será asumida como una responsabilidad tanto académica como social del quehacer universitario y el referente más sensible de su inserción en la sociedad.

- La Extensión Universitaria se constituye en el vínculo natural, que a través de diversas manifestaciones culturales, extracurriculares y de acción social, potencian los valores éticos, espirituales y cristianos como contribución al desarrollo de la región.
- Las Actividades de Extensión Universitaria favorecerán la interacción de la Universidad con los demás componentes de la sociedad para mutuo beneficio, y se tornan en una expresión clara de la formación integral que como principio tiene la PUCESA.
- El quehacer universitario es el espacio propicio para impulsar planes de desarrollo integral en la búsqueda de un crecimiento social, económico y cultural donde la Extensión Universitaria se torna en un medio idóneo que favorece el aporte directo del estudiante.
- Se favorecerá el contacto específico de los estudiantes con instituciones, sectores y/o personas de realidades críticas en función de generar un espíritu de solidaridad y proyección social.
- Como signo de la identidad cristiana de la Universidad y en una actitud de solidaridad, la Extensión Universitaria busca suscitar un compromiso y fraternidad con grupos humanos vulnerables de realidades específicas.
- Las actividades de Extensión Universitaria se planifican, ejecutan y evalúan en conjunto, universidad y comunidad.

- Ninguna actividad desarrollada como Extensión Universitaria por parte de los estudiantes será remunerada.

Las actividades de Extensión Universitaria se realizarán fuera de las horas clase, salvo casos excepcionales previa autorización.

Toda actividad contemplada como Extensión Universitaria será autofinanciada por los estudiantes y realizada en instituciones públicas, ONG's y entidades sin fines de lucro cuyos representantes no tengan vínculos familiares con el extensionista.

La Universidad realizará el seguimiento y verificación del cumplimiento de las actividades programadas en las horas de extensión.

La Universidad velará por la integridad física y emocional de los estudiantes mientras realicen sus horas de extensión.

La Universidad se reserva el derecho de dar por terminada una actividad o convenio cuando creyere conveniente.

Para acreditarse horas de extensión, por las actividades realizadas, el estudiante debe contar con la respectiva autorización que se emana desde el Departamento de Pastoral Universitaria y la Dirección de Estudiantes.

Toda actividad tendrá un mínimo de 10 horas por ocasión, debiendo registrarse hasta totalizar 120 horas en el Departamento de Pastoral Universitaria y 80 horas en la Dirección de Estudiantes.

Toda situación referente a las Extensiones Universitarias que no se contemple en el presente documento, será tratada y resuelta por el Departamento de Pastoral Universitaria y la Dirección de Estudiantes.

2.1.3 Procedimiento para solicitar horas de Extensión Universitaria en el Departamento de Pastoral Universitaria y la Dirección de Estudiantes

- El estudiante regular de la PUCESA requerirá autorización para realizar una actividad de Extensión Universitaria por lo que deberá presentar una solicitud en papel bond simple al Departamento de Pastoral Universitaria o a la Dirección de Estudiantes; donde debe consignarse: el nombre del alumno, paralelo y Escuela a la que pertenece, el número de matrícula, la actividad a realizar, el nombre de la Institución o del responsable, y la fecha de inicio.
- El Departamento de Pastoral Universitaria o la Dirección de Estudiantes solicitará al responsable institucional, al administrador del convenio o coordinador/tutor de la actividad que entregue un informe escrito sobre el desempeño, al igual que un certificado de conducta que indique que el estudiante ha cumplido a cabalidad la actividad a él asignada. Este

procedimiento se repetirá cuantas veces sea necesario hasta completar el número de horas establecidos, donde al reverso de la solicitud se avalizará la actividad respectiva por parte del Departamento de Pastoral Universitaria o la Dirección de Estudiantes, adjuntando los informes correspondientes.

- Una vez realizadas la totalidad de horas de Extensión Universitaria, el estudiante solicitará al Departamento de Pastoral Universitaria o a la Dirección de Estudiantes la aprobación de las mismas; la solicitud se realizará en hoja membretada.
- El Departamento de Pastoral Universitaria y la Dirección de Estudiantes emitirá la aprobación en la parte de atrás de la solicitud del alumno, para posteriormente entregarlo a la secretaría de la respectiva unidad académica, la misma que archivará el documento en la carpeta personal del alumno.
- El Departamento de Pastoral Universitaria y la Dirección de Estudiantes mantendrá un respaldo digital respecto del número de horas consignadas a cada alumno.

2.2 Ingeniería de Software

2.2.1 Definición

El software es un conjunto de instrucciones (programas de computación) que cuando se ejecutan proveen una función y desempeño deseados, estructuras de datos que permiten a los programas manipular la información adecuadamente, y documentos que describen el funcionamiento y uso de los programas.

2.2.2 Características del software

Para comprender lo que es el software (y finalmente una comprensión de Ingeniería de Software), es importante examinar las características del software que lo hace diferente de otras cosas que los seres humanos construyen. Cuando el hardware se construye, el proceso creativo humano (análisis, diseño, construcción y prueba) se traduce finalmente de forma física. Si nosotros construimos una nueva computadora, nuestros bocetos iniciales, dibujos de diseño y prototipo evolucionan en un producto físico (los chips, placas base, fuentes de alimentación, etc.).

El software es algo lógico antes que un elemento físico del sistema. Por consiguiente, el software tiene características que son considerablemente diferentes que aquellas del hardware:

1. El software se desarrolla o se diseña, no se fabrica en el sentido clásico.

Aunque existen algunas similitudes entre el desarrollo del software y la fabricación del hardware, son fundamentalmente diferentes.

En ambas actividades, la alta calidad se logra a través de un buen diseño, pero la fase de fabricación para el hardware puede introducir problemas de calidad que son inexistentes (o fácilmente corregibles) para el software.

Ambas actividades son dependientes de las personas. Ambas actividades requieren la construcción de un "producto" pero los alcances son diferentes.

Los costos del Software son concentrados en ingeniería. Esto significa que los proyectos de software no pueden manejarse como si fueran proyectos de fabricación.

2. El software no se desgasta

El hardware exhibe proporciones de fracaso relativamente altas tempranamente en su vida (estos fracasos son a menudo atribuibles al diseño o defectos de fabricación); se corrigen los defectos y la tasa de fracaso cae a un nivel estable (con suerte, bastante bajo) por algún período de tiempo. Sin embargo, a medida que pasa el tiempo, la proporción de fracaso se eleva nuevamente cuando los componentes del hardware son

afectados por la acumulación de polvo, la vibración, el abuso, extremas temperaturas, y muchas otras influencias del medio. Simplemente, el hardware empieza a desgastarse.

Por otro lado, el software no es susceptible a los efectos medioambientales que causan el desgaste del hardware. Los defectos no descubiertos causarían altas proporciones de fracaso tempranamente en la vida de un programa. Sin embargo, éstos se corrigen (con suerte, sin introducir otros errores).

Sin embargo, la implicación es clara, el software no se desgasta. Pero se deteriora.

Durante su vida, el software sufrirá el cambio (mantenimiento). Cuando los cambios son hechos, es probable que algunos nuevos defectos sean introducidos, el software se está deteriorando debido a los cambios.

Otro aspecto de uso ilustra la diferencia entre el hardware y software.

Cuando un componente del hardware se desgasta, es reemplazada por una parte de repuesto. No existen partes de repuesto para el software. Cada falla del software indica un error en el diseño o en el proceso a través del cual el diseño se tradujo al código de máquina ejecutable. Por consiguiente, el mantenimiento del software involucra considerablemente más complejidad que el mantenimiento de hardware.

3. Aunque la industria está acercándose al ensamblaje basado en componentes, la mayoría del software continúa siendo una creación personalizada.

Consideremos la manera en que el control de hardware para un producto computarizado es diseñado y construido. El ingeniero de diseño dibuja un simple esquema digital de circuitos, hace algún análisis fundamental para asegurar que el apropiado funcionamiento sea logrado, y entonces va al estante dónde se encuentran los catálogos de componentes digitales. Cada circuito integrado (llamado IC o chip) tiene un número de parte y una función e interfaz definida, y un conjunto estándar de pautas para integración. Después, cada componente es seleccionado, y puesto fuera del estante.

Cuando una disciplina de la ingeniería evoluciona, una colección de componentes estándar de diseño son creados. Los tornillos estándar y los circuitos integrados listos para ser usados son sólo dos de miles de componentes estándares usados por ingenieros mecánico y eléctricos cuando diseñan nuevos sistemas. Los componentes reusables se han creado para que el ingeniero pueda concentrarse en los elementos verdaderamente innovadores de un diseño, es decir, las partes del diseño que representa algo nuevo. En el mundo del hardware, la reutilización de componentes es una parte natural del proceso de ingeniería. En el mundo del software, esto es algo que sólo ha empezado para ser logrado en una amplia escala.

Un componente del software debe ser diseñado e implementado para que pueda ser reusado en muchos programas diferentes. En los años sesenta, fueron construidas las librerías de subrutina científicas que fueron reusadas en una amplia serie de aplicaciones científicas y de ingeniería. Estas librerías de subrutina reusaron los algoritmos bien-definidos de una manera eficaz pero tenían un dominio limitado de aplicación. Hoy, se ha extendido el punto de vista de reuso para abarcar no sólo los algoritmos sino también la estructura de datos. Los componentes reusables modernos encapsulan datos y el proceso aplicado a los datos, habilitando al ingeniero de software a crear las nuevas aplicaciones a partir de partes reusables. Por ejemplo, las interfaces gráficas de usuario de hoy se construyen usando componentes reusables que habilitan la creación de ventanas gráficas, menús desplegables, y una amplia variedad de mecanismos de interacción. La estructura de datos y el proceso detallado requeridos para la construcción de la interfaz son contenidos con una librería de componentes reusable para la construcción de la interfaz.

2.2.3 Aplicaciones de Software

Software de Sistemas.- El software de Sistemas es una colección de programas escrita para servir a otros programas. Algunos software de sistemas (compiladores, editores, y utilidades de administración de archivos) procesan complejas estructuras de información. Otras aplicaciones de sistemas (componentes del sistema operativo, telecomunicaciones,

procesadores) procesan indeterminados datos. En cualquier caso, el área de software de sistemas se caracteriza por una alta interacción con el hardware de computación; alto uso de usuarios múltiples; funcionamiento concurrente que requiere planificación, recursos compartidos, y sofisticada dirección de procesos; estructura compleja de datos; y múltiples interfaces externas.

Software de tiempo real.- El software que mide, analiza y controla sucesos del mundo real conforme ocurren se llama de tiempo real. Los elementos del software de tiempo real incluyen un componente de recolección que colecciona y estructura la información de un ambiente externo, un componente de análisis que transforma la información como es requerida por la aplicación, un componente de control/salida que responde al ambiente externo, y un componente de supervisión que coordina todos los otros componentes para dar respuesta en tiempo real.

Software de Gestión.- La informática comercial es la más grande área de aplicación de software. Los "sistemas" discretos (por ejemplo, nómina, cuentas de haberes/débitos, inventario) ha evolucionado en sistemas de Administración de información, software que accesa a uno o más bases de datos que contienen la información de negocios. Las aplicaciones en ésta área reestructuran los datos existentes de manera que facilitan las operaciones de negocio o toma de decisiones. Además de la aplicación al procesamiento de datos convencional, las aplicaciones del software de negocios también abarcan la informática interactiva (por ejemplo, procesamiento en puntos de venta).

Software científico y de Ingeniería.- El Software científico y de Ingeniería ha sido caracterizado por los algoritmos de manejo de números. Las aplicaciones van desde la astronomía a la vulcanología, de la biología molecular a la fabricación automatizada. Sin embargo, las aplicaciones dentro del área de ingeniería y ciencias se están moviendo fuera de los algoritmos numéricos convencionales. El diseño asistido por computadora, simulación de sistemas, y otras aplicaciones interactivas han empezado a asumir características de software de tiempo real e incluso de Software de Sistemas.

Software Empotrado.- Los productos inteligentes se han vuelto comunes casi en todo mercado industrial y de consumo. El software empotrado reside en una memoria de sólo lectura y se usa para controlar productos y sistemas para los mercados industriales y de consumo.

El software empotrado puede realizar funciones muy limitadas o proporciona una significativa función y capacidad de mando (por ejemplo, funciones digitales en un automóvil como el mando de combustible, y los sistemas de frenado).

Software de Computadoras Personales.- El mercado de Software para computador personal ha crecido durante las últimas décadas. Procesadores de palabras, hojas de cálculo, gráficos de computadora, multimedia, entretenimiento, administración de base de datos, aplicaciones financieras personales y de negocios, red externa, y acceso a base de datos son sólo algunas de centenares de aplicaciones.

Software Basado en la Web.- Las páginas Web recuperadas por un navegador son software que incorpora instrucciones ejecutables (Ej., CGI, HTML, Perl, o Java), y datos (ej., el hipertexto y una variedad de formatos visuales y audio). En esencia, la red se vuelve una computadora que proporciona recursos de software casi ilimitados a los que cualquiera puede acceder.

Software de inteligencia artificial.- El software de Inteligencia artificial hace uso de algoritmos no numéricos para resolver problemas complejos que no son adecuados el cálculo o el análisis directo. Los sistemas expertos, también llamados sistemas basados en conocimiento, reconocimiento de patrones (imagen y voz), redes neuronales artificiales, teoremas de prueba y juegos son aplicaciones representativas de ésta categoría.

2.2.4 Calidad de Software

2.2.4.1 Definición de Calidad

El Diccionario "American Heritage" define a la calidad como “una característica o atributo de algo.” Como un atributo de un artículo, la calidad se refiere a las características medibles, cosas que podemos comparar a las normas conocidas como la longitud, color, propiedades eléctricas y maleabilidad. Sin embargo, el software, como una entidad intelectual, es más desafiante para caracterizar que los objetos físicos.

No obstante, las medidas de las características de un programa existen. Estas propiedades incluyen complejidad, cohesión, el número de puntos de función, líneas de código, y muchos otros. Cuando se examina un artículo basado en sus características medibles, pueden encontrarse dos tipos de calidad: calidad de diseño y calidad de conformidad.

La calidad de diseño se refiere a las características que los diseñadores especifican para un artículo. La calidad de materiales, tolerancias, y especificaciones de funcionamiento contribuyen a la calidad de diseño. Cuanto más alta calidad de materiales se usen, tolerancias más firmes y mayores niveles de desempeño son especificados, la calidad del diseño de un producto aumenta, si el producto es manufacturado según las especificaciones.

La calidad de conformidad es el grado a que las especificaciones del diseño se siguen durante la fabricación. De nuevo, a mayor grado de conformidad, superior es el nivel de calidad de conformidad.

En el desarrollo del software, la calidad de diseño abarca requisitos, especificaciones, y el diseño del sistema. La calidad de conformidad es un problema enfocado principalmente en la aplicación. Si la aplicación sigue el diseño y el sistema resultante cumple los requisitos y metas en desempeño, la calidad de conformidad es alta. ¿Pero son calidad de diseño y calidad de conformidad los únicos problemas que los ingenieros de software deben considerar?

Robert Glass, defiende que la más “intuitiva” relación está en el orden de:

Satisfacción del usuario = producto dócil + buena calidad + entrega dentro del presupuesto y horario.

Finalmente, Glass considera que la calidad es importante, pero si el usuario no está satisfecho, nada más importa realmente. DeMarco refuerza ésta punto de vista cuando establece: “La calidad de un producto está dada en función de cuánto mejora el mundo.” Este punto de vista de calidad sostiene que si un producto de software provee un sustancial beneficio a los usuarios finales, ellos pueden estar dispuestos a tolerar dependencia ocasional o problemas de desempeño.

2.2.4.2 Definición de Calidad de Software

Incluso los diseñadores del software más expertos estarán de acuerdo que software de alta calidad es una meta importante.

Se han propuesto muchas definiciones de calidad del software en la literatura. Para nuestro propósito, la calidad del software está definida como:

“Conformidad con una explícita funcionalidad y requisitos de desempeño declarados, estándares de desarrollo explícitamente documentados, y

características implícitas que son esperadas de todo software profesionalmente desarrollado”.

Esta definición podría modificarse o extenderse. De hecho, una definición definitiva de calidad del software podría debatirse eternamente. El propósito de la definición sirve dar énfasis a éstos tres puntos importantes:

- Los requisitos del software son el fundamento mediante el cual la calidad es medida. Falta de conformidad con los requisitos es falta de calidad.
- Los Estándares específicos definen un criterio de desarrollo que guía la manera en que el software se diseña. Si el criterio no se sigue, ciertamente dará como resultado una falta de la calidad.
- Un conjunto de requisitos implícitos, a menudo no mencionados (por ejemplo, el deseo para la facilidad de uso y el buen mantenimiento). Si el software está conforme a sus requisitos explícitos pero no reúne los requisitos implícitos, la calidad del software es sospechosa.

2.3 Bases de Datos

2.3.1 Definición

- Es un conjunto de datos interrelacionados y almacenados sin redundancias perjudiciales o innecesarias, los cuales sirven a las aplicaciones sin estar relacionados de una manera directa entre ellos y se caracterizan por:
 1. Servir a una o más aplicaciones.
 2. Existir independencia entre los datos y los programas que los manejan.

- Es una colección de datos que están lógicamente relacionados entre sí, los datos están estructurados según un modelo de Base de Datos que refleja las relaciones y restricciones que tienen estos datos en el mundo real. La descripción y definición de los datos se encuentra almacenada en la propia Base de Datos.

- Es un conjunto de información que: Es estructurada, es accesible por múltiples usuarios de forma simultánea y tiene independencia entre los datos y los programas que los van a tratar.

A estas tres características podemos añadir:

- Coherencia.
- No redundancia innecesaria.
- Seguridad.

Una base de datos puede ser utilizada por varias aplicaciones y usuarios. Toda base de datos debe permitir insertar, modificar y borrar datos por lo que en las bases de datos se guarda información de dos tipos:

- Los datos de usuarios (datos usados por las aplicaciones).
- Los datos de sistema (datos que la base de datos utiliza para su gestión).

2.3.2 Organización de una Base de Datos

Para considerar a una base de datos organizada debe cumplir los siguientes objetivos:

- ***Versatilidad para recuperar la información.***- Esto quiere decir que, dependiendo de los usuarios o las aplicaciones, puedan hacer diferentes cosas o traten a los datos de distinta manera.

- **Desempeño.-** Tiene que atender con la rapidez adecuada a las peticiones que se les haga dependiendo del uso que se vaya a hacer de los datos, atendiendo a lo que se la requiera.
- **Redundancia mínima.-** Tiene que tener un índice de redundancia lo más bajo posible. Un objetivo importante en las Bases de Datos es eliminar las redundancias siempre y cuando esto no implique un aumento de complejidad y una reducción de rendimiento.
- **Capacidad de acceso.-** Tener una alta capacidad de acceso para ganar el mayor tiempo posible en la realización de consultas. Con una buena organización física se conseguirán tiempos de respuesta mucho menores.
- **Integridad.-** Tener un alto índice de integridad, esto significa que al tener muchos usuarios accediendo a una misma base de datos no puede haber fallos en la inserción de datos, errores por redundancia o lenta actualización o almacenamiento, es decir, que los valores almacenados cumplan ciertas restricciones definidas con anterioridad y que los fallos producidos en el sistema no destruyan los datos. Con este fin los sistemas de almacenamiento, inserción y realización han de asegurar que el sistema puede recuperarse en caso de fallo, sin pérdida de información.
- **Seguridad.-** Por supuesto tienen que tener un nivel altísimo de seguridad y privacidad ya que los datos que se pueden almacenar en una base de datos pueden ser altamente confidenciales o importantes. Esto explica la

necesidad de disponer de procedimientos que protejan los datos frente a fallos del sistema evitando su pérdida y facilitando su recuperación o frente a usos indebidos o mal intencionados por parte de personas no autorizadas. En este punto también entran los medios físicos de protección contra fuego, robo, etc.

- **Ajuste.-** Con el tiempo el volumen de los datos que se necesitan van aumentando, por lo que la organización física tiene que ser buena y es posible que haya que ajustarla para atender a las nuevas necesidades de almacenamiento. Para llevar a cabo un ajuste correcto es necesario una dependencia física de los datos que no suponga modificaciones en los programas de aplicación.
- **Interfaz con el pasado y con el futuro.-** Con el paso del tiempo las necesidades van cambiando por tanto se hace necesario actualizar el sistema de procesamiento. Cuando se determina instalar un software de Base de Datos es importante que pueda trabajar con datos anteriores (versiones anteriores), igualmente, una Base de Datos ha de estar preparada para los futuros cambios del software de gestión y de almacenamiento. Cuando se hacen cambios en la estructura lógica de los datos estos no deben afectar a los programas de aplicación que utilizan estos datos, esto es la independencia lógica de los datos.

2.3.3 Abstracción de la Información

Una Base de Datos es en esencia una colección de archivos relacionados entre sí, de la cual los usuarios pueden extraer información sin considerar las fronteras de los archivos.

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

Existen diferentes niveles de abstracción para simplificar la interacción de los usuarios con el sistema; Interno, conceptual y externo, específicamente el de almacenamiento físico, el del usuario y el del programador.

2.3.3.1 Nivel físico

Es la representación del nivel más bajo de abstracción, en éste se describe en detalle la forma en como se almacenan los datos en los dispositivos de almacenamiento, mediante señaladores o índices para el acceso aleatorio a los datos.

2.3.3.2 Nivel conceptual

El siguiente nivel más alto de abstracción, describe qué datos son almacenados realmente en la base de datos y las relaciones que existen entre los mismos, describe la base de datos completa en términos de su estructura de diseño. El nivel conceptual de abstracción lo usan los administradores de bases de datos, quienes deben decidir qué información se va a guardar en la base de datos.

Consta de las siguientes definiciones:

- **Definición de los datos:** Se describen el tipo de datos y la longitud de campo de todos los elementos direccionables en la base. Los elementos por definir incluyen artículos elementales (atributos), totales de datos y registros conceptuales (entidades).
- **Relaciones entre datos:** Se definen las relaciones entre datos para enlazar tipos de registros relacionados para el procesamiento de archivos múltiples.

En el nivel conceptual la base de datos aparece como una colección de registros lógicos, sin descriptores de almacenamiento. En realidad los archivos conceptuales no existen físicamente. La transformación de registros conceptuales a registros físicos para el almacenamiento se lleva a cabo por el sistema y es transparente al usuario.

2.3.3.3 Nivel de visión

Nivel más alto de abstracción, es lo que el usuario final puede visualizar del sistema terminado, describe sólo una parte de la base de datos al usuario. El sistema puede proporcionar muchas visiones para la misma base de datos.

2.3.4 Modelos de Base de Datos

Es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia.

El objetivo de cualquier sistema de información es representar mediante una abstracción del mundo real toda la información necesaria para el cumplimiento de los fines.

Los modelos de datos se dividen en tres grupos:

- Modelos lógicos basados en objetos.
- Modelos lógicos basados en registros.
- Modelos físicos de datos.

2.3.4.1 Modelos lógicos basados en objetos

Se usan para describir datos en los niveles conceptual y de visión, es decir, con este modelo representamos los datos de la forma como nosotros los captamos en el mundo real, tienen una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. Existen diferentes modelos de este tipo, pero el más utilizado por su sencillez y eficiencia es el modelo Entidad-Relación.

2.3.4.1.1 Modelo Entidad-Relación

Este modelo representa a la realidad a través de entidades, que son objetos que existen que se distinguen de otros por sus características.

Las entidades pueden ser de dos tipos:

- **Tangibles:** Son todos aquellos objetos físicos que podemos ver, tocar o sentir.
- **Intangibles:** Todos aquellos eventos u objetos conceptuales que no podemos ver, aun sabiendo que existen.

Las características de las entidades en base de datos se llaman atributos. A su vez una entidad se puede asociar o relacionar con más entidades a través de relaciones.

2.3.4.2 Modelos lógicos basados en registros

Se utilizan para describir datos en los niveles conceptual y físico.

Estos modelos utilizan registros e instancias para representar la realidad, así como las relaciones que existen entre estos registros (enlaces) o apuntadores. A diferencia de los modelos de datos basados en objetos, se usan para especificar la estructura lógica global de la base de datos y para proporcionar una descripción a nivel más alto de la implementación.

Los tres modelos de datos más ampliamente aceptados son: Modelo Jerárquico, Modelo de Red y Modelo Relacional.

2.3.4.2.1 Modelo jerárquico

Utiliza árboles para representar los datos a un nivel lógico, cada árbol esta compuesto por una jerarquía de elementos llamados nodos, al nodo más alto llamado raíz y cada nodo representa a un tipo de registro que se llama segmento con sus correspondientes campos.

Inconvenientes:

- No se permite más de una relación entre dos segmentos.
- El árbol se suele recorrer en un cierto orden.

- Para acceder a cualquier segmento es necesario pasar por el punto raíz.

2.3.4.2.2 Modelo de red

Utiliza estructuras de datos en las cuales las entidades se representan con nodos y las relaciones con líneas entre nodos, representa los datos mediante colecciones de registros y sus relaciones se representan por medio de enlaces. Los registros se organizan en un conjunto de gráficas arbitrarias.

2.3.4.2.3 Modelo relacional

Utiliza tablas bidimensionales para la representación lógica de los datos y las relaciones entre ellas, en las cuales los renglones (tuplas) equivalen a los cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla.

Ventajas:

- Ofrece una visión sencilla del conjunto de datos de la Base de Datos.
- Puede entenderlo y usarlo cualquier usuario.
- Los usuarios no tienen que saber donde se encuentran físicamente los datos.

- Se puede ampliar el esquema conceptual sin modificar los programas de aplicación.

2.3.4.3 Modelos físicos de datos

Se usan para describir a los datos en el nivel más bajo, aunque existen muy pocos modelos de este tipo, básicamente capturan aspectos de la implementación de los sistemas de base de datos.

2.3.5 Sistema de Gestión de Base de Datos (SGBD)

El sistema de Gestión de Bases de Datos es la porción más importante del software Base de Datos. Un SGBD es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica como: Definición de los datos en los distintos niveles de abstracción, manipulación de los datos es decir, inserción, modificación, borrado y consulta, mantenimiento de la integridad de la base de datos y control de la seguridad.

Las funciones principales de un SGBD son:

- 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.
- Interacción con el manejador de archivos. Así el Manejador de base de datos es el 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 fallas 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. Consiste en contar con mecanismos que permitan el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

2.4 Base Datos MySql

2.4.1 Introducción

MySQL es un gestor de base de datos interpretador de SQL (Lenguaje de Consulta Estructurado) sencillo de usar e increíblemente rápido. Es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales.

Las características principales de MySQL son:

- **Es un gestor de base de datos.-** Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda.
- **Es una base de datos relacional.-** Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.
- **Es Open Source.-** El código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL para aplicaciones no comerciales.
- **Es una base de datos muy rápida, segura y fácil de usar.-** Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando y optimizando en velocidad. Por eso es una de las bases de datos más usadas en el Internet.
- **Existe una gran cantidad de software que la usa.**

2.4.2 Comandos en MySQL

- Un comando normalmente consiste de una sentencia SQL seguida por un punto y coma.
- Cuando emitimos un comando, MySQL lo manda al servidor para que lo ejecute, nos muestra los resultados y regresa el prompt indicando que está listo para recibir más consultas.
- MySQL muestra los resultados de la consulta como una tabla (filas y columnas). La primera fila contiene etiquetas para las columnas. Las filas siguientes muestran los resultados de la consulta. Normalmente las etiquetas de las columnas son los nombres de los campos de las tablas que estamos usando en alguna consulta. Si lo que estamos recuperando es el valor de una expresión, las etiquetas en las columnas son la expresión en sí.
- MySQL muestra cuántas filas fueron regresadas y cuánto tiempo tardó en ejecutarse la consulta, estos valores pueden ser un tanto imprecisos porque pueden verse afectados por factores tales como la carga del servidor y la velocidad de comunicación en una red.
- Las palabras clave de SQL pueden ser escritas usando mayúsculas o minúsculas.

- Un comando no necesita ser escrito en una sola línea, así que los comandos que requieran de varias líneas no son un problema. MySQL determinará en dónde finaliza la sentencia cuando encuentre el punto y coma, no cuando encuentre el fin de línea.

2.4.3 Conectar al Servidor MySQL

Para conectarse al servidor, usualmente necesitamos de un nombre de usuario (login) y de una contraseña (password), y si el servidor al que nos deseamos conectar está en una máquina diferente de la nuestra, también necesitamos indicar el nombre o la dirección IP de dicho servidor. Una vez que conocemos estos tres valores, podemos conectarnos de la siguiente manera:

```
mysql -h NombreDelServidor -u NombreDeUsuario -p
```

Cuando ejecutamos este comando, se nos pedirá que proporcionemos también la contraseña para el nombre de usuario que estamos usando.

Si la conexión al servidor MySQL se pudo establecer de manera satisfactoria, recibiremos el mensaje de bienvenida y estaremos en el prompt de MySQL.

Después de que nos hemos conectado de manera satisfactoria, podemos desconectarnos en cualquier momento al escribir “quit” o “exit”.

2.4.4 Creación de Base de Datos

Una base de datos tiene un nombre con el cual accederemos a ella, para que el servidor nos muestre las bases de datos existentes, se lo solicitamos usando el comando “SHOW DATABASES”:

```
mysql> SHOW DATABASES;
```

Para crear una base de datos se usa el comando “CREATE DATABASE”:

```
mysql> CREATE DATABASE NombreDeLaBase;
```

Al crear una base de datos no se selecciona ésta de manera automática; debemos hacerlo de manera explícita, por ello usamos el comando “USE”.

```
mysql> USE NombreDeLaBase;
```

Para ver las tablas que conforman una base, lo hacemos usando la sentencia “SHOW TABLES”, una vez que ya tenemos seleccionada la base con la que queremos trabajar.

```
mysql> SHOW TABLES;
```

Al crear una tabla debemos resolver qué campos (columnas) tendrá y qué tipo de datos almacenarán cada uno de ellos, es decir, su estructura.

La tabla debe ser definida con un nombre que la identifique y con el cual accederemos a ella.

Cuando se crea una tabla debemos indicar su nombre y definir sus campos con su tipo de dato, para crear una tabla se usa la sentencia “CREATE TABLE”.

```
mysql> CREATE TABLE NombreDeLaTabla (nombre varchar(30),  
clave varchar(10));
```

En esta tabla definimos 2 campos:

- nombre: que contendrá una cadena de hasta 30 caracteres de longitud, que almacenará un nombre de usuario y
- clave: otra cadena de caracteres de 10 de longitud, que guardará la clave de cada usuario.

Cada usuario ocupará un registro de esta tabla, con su respectivo nombre y clave.

Para ver la estructura de una tabla usamos el comando “DESCRIBE” junto al nombre de la tabla:

```
mysql> DESCRIBE NombreDeLaTabla;
```

Se mostrará cada campo, su tipo, lo que ocupa en bytes y otros datos como la aceptación de valores nulos, etc.

Para eliminar una tabla se usa el comando “DROP TABLE”.

```
mysql> DROP TABLE NombreDeLaTabla;
```

2.4.5 Carga y Recuperación de Registros de una Tabla

Un registro es una fila de la tabla que contiene los datos propiamente dichos.

Cada registro tiene un dato por cada columna. Al ingresar los datos de cada registro debe tenerse en cuenta la cantidad y el orden de los campos.

Para agregar registros en una tabla se usa la sentencia “INSERT INTO”:

```
mysql> INSERT INTO NombreDeLaTabla (NombresCampos,..)  
VALUES ('Valores',...);
```

Especificamos los nombres de los campos entre paréntesis y separados por comas y luego los valores para cada campo, también entre paréntesis y separados por comas.

Es importante ingresar los valores en el mismo orden en que se nombran los campos.

Los datos ingresados que corresponden a campos de cadenas de caracteres se colocan entre comillas simples. Las comillas simples son OBLIGATORIAS.

Para recuperar registros de una tabla usamos “SELECT”:

```
mysql> SELECT Campo FROM NombreDeLaTabla;
```

El comando “SELECT” recupera los registros de una tabla. Luego del comando “SELECT” indicamos el o los nombres de los campos que queremos recuperar separados por comas.

Con el asterisco (*) indicamos que seleccione todos los campos de la tabla que nombramos.

La cláusula, “WHERE” que es opcional, permite especificar condiciones para la consulta “SELECT”. Es decir, podemos recuperar algunos registros que cumplan con ciertas condiciones indicadas con la cláusula “WHERE”.

2.4.6 Operadores Relacionales

Los operadores relacionales son usados en combinación con la cláusula “WHERE”, vinculan un campo con un valor para que MySQL compare cada registro (el campo especificado) con el valor dado.

Los operadores relacionales son los siguientes:

| | |
|----|---------------|
| = | igual |
| <> | distinto |
| > | mayor |
| < | menor |
| >= | mayor o igual |
| <= | menor o igual |

Existen otros operadores que simplifican algunas consultas:

- “BETWEEN” significa “entre”. Si agregamos el operador “NOT” antes de “BETWEEN” el resultado se invierte.
- Con “IN” averiguamos si el valor de un campo dado está incluido en una lista de valores especificada. Con “IN” averiguamos si el valor del campo está incluido en la lista, con “NOT” antecediendo la condición, invertimos el resultado.

2.4.7 Borrar Registros de una Tabla

Para eliminar los registros de una tabla usamos el comando “DELETE”:

```
mysql> DELETE FROM NombreDeLaTabla;
```

Para borrar todos los registros de una tabla se usa “DELETE” sin condición “WHERE”.

Si queremos eliminar uno o varios registros debemos indicar cuál o cuáles, para ello utilizamos el comando “DELETE” junto con la cláusula “WHERE” con la cual establecemos la condición que deben cumplir los registros a borrar.

También podemos eliminar todos los registros de una tabla mediante el comando “TRUNCATE TABLE”.

```
mysql> TRUNCATE TABLE NombreDeLaTabla;
```

La sentencia “TRUNCATE TABLE” vacía la tabla (elimina todos los registros) y deja la tabla con la misma estructura.

La diferencia con “DROP TABLE” es que esta sentencia borra la tabla, “TRUNCATE TABLE” la vacía.

La diferencia con “DELETE” es la velocidad, es más rápido el comando “TRUNCATE TABLE” que el “DELETE” (se nota cuando la cantidad de registros es muy grande) ya que éste borra los registros uno a uno.

2.4.8 Modificar Registros de una Tabla

Para modificar uno o varios datos de uno o varios registros utilizamos el comando “UPDATE” (actualizar).

```
mysql> UPDATE NombreDeLaTabla SET  
NombreDelCampo='Valor';
```

Utilizamos “UPDATE” junto al nombre de la tabla y “SET” junto con el campo a modificar y su nuevo valor.

Podemos modificar algunos registros, para ello debemos establecer condiciones de selección con “WHERE”. Las condiciones no son obligatorias, pero si omitimos la cláusula “WHERE”, la actualización afectará a todos los registros.

2.4.9 Clave Primaria de una Tabla

Una clave primaria es un campo (o varios) que identifica 1 solo registro (fila) en una tabla.

Para un valor del campo clave existe solamente 1 registro. Los valores no se repiten ni pueden ser nulos.

Establecemos que un campo sea clave primaria al momento de creación de la tabla:

```
mysql> CREATE TABLE NombreDeLaTabla (  
nombre varchar(20),  
clave varchar(10),  
PRIMARY KEY(nombre)  
);
```

Para definir un campo como clave primaria agregamos “PRIMARY KEY” luego de la definición de todos los campos y entre paréntesis colocamos el nombre del campo que queremos como clave.

Los campos definidos como clave primaria no pueden repetirse.

2.4.10 Campo entero con autoincremento

Un campo de tipo entero puede tener otro atributo extra ‘auto_increment’.

Los valores de un campo ‘auto_increment’, se inician en 1 y se incrementan en 1 automáticamente.

Se utiliza generalmente en campos correspondientes a códigos de identificación para generar valores únicos para cada nuevo registro que se inserta.

Sólo puede haber un campo “AUTO_INCREMENT” y debe ser clave primaria (o estar indexado).

Para establecer que un campo autoincremente sus valores automáticamente, éste debe ser entero (integer) y debe ser clave primaria.

Para definir un campo autoincrementable colocamos “AUTO_INCREMENT” luego de la definición del campo al crear la tabla.

Cuando un campo tiene el atributo “AUTO_INCREMENT” no es necesario ingresar valor para él, porque se inserta automáticamente tomando el último valor como referencia, o 1 si es el primero.

Para ingresar registros omitimos el campo definido como “AUTO_INCREMENT”.

Si continuamos ingresando registros, el código (dato que no ingresamos) se cargará automáticamente siguiendo la secuencia de autoincremento.

Pero debemos tener cuidado con la inserción de un dato en campos “auto_increment”. Debemos tener en cuenta que:

- Si el valor está repetido aparecerá un mensaje de error y el registro no se ingresará.

- Si el valor dado saltea la secuencia, lo toma igualmente y en las siguientes inserciones, continuará la secuencia tomando el valor más alto.
- Si el valor ingresado es 0, no lo toma y guarda el registro continuando la secuencia.
- Si el valor ingresado es negativo (y el campo no está definido para aceptar sólo valores positivos), lo ingresa.

2.4.11 Tipos de datos

Al crear una tabla debemos elegir la estructura adecuada, esto es, definir los campos y sus tipos más precisos, según el caso.

Los Tipos de Datos Básicos son:

- **varchar:** Se usa para almacenar cadenas de caracteres. Una cadena es una secuencia de caracteres. Se coloca entre comillas (simples). El tipo "varchar" define una cadena de longitud variable en la cual determinamos el máximo de caracteres. Puede guardar hasta 255 caracteres. Para almacenar cadenas de hasta 30 caracteres, definimos un campo de tipo varchar(30).

- **integer:** Se usa para guardar valores numéricos enteros, de -2000000000 a 2000000000 aproximadamente.
- **float:** Se usa para almacenar valores numéricos decimales. Se utiliza como separador el punto (.).

Hay más tipos, incluso, subtipos.

Los valores que podemos guardar son:

- **Texto:** Para almacenar texto usamos cadenas de caracteres. Las cadenas se colocan entre comillas simples. Podemos almacenar dígitos con los que no se realizan operaciones matemáticas. Tenemos los siguientes tipos: varchar, char y text.
- **Números:** Existe variedad de tipos numéricos para representar enteros, negativos, decimales. Para almacenar valores enteros usamos el tipo integer. Para almacenar valores con decimales utilizamos: float o decimal.
- **Fechas y Horas:** Para guardar fechas y horas dispone de varios tipos: date (fecha), datetime (fecha y hora), time (hora), year (año) y timestamp.
- **Otros Tipos:** enum y set representan una enumeración y un conjunto respectivamente.

- Otro valor que podemos almacenar es el valor “null”. El valor ‘null’ significa “valor desconocido” o “dato inexistente. No es lo mismo que 0 o una cadena vacía.

2.4.11.1 Tipos de Datos Texto

Para almacenar Texto usamos cadenas de caracteres. Las cadenas se colocan entre comillas simples. Podemos almacenar dígitos con los que no se realizan operaciones matemáticas.

Tenemos los siguientes tipos:

- **varchar(x):** Define una cadena de caracteres de longitud variable en la cual determinamos el máximo de caracteres con el argumento “x” que va entre paréntesis. Su rango va de 1 a 255 caracteres. Un varchar(10) ocupa 11 bytes, pues en uno de ellos almacena la longitud de la cadena. Ocupa un byte más que la cantidad definida.
- **char(x):** Define una cadena de longitud fija, su rango es de 1 a 255 caracteres. Si la cadena ingresada es menor a la longitud definida, almacena espacios en blanco a la derecha, tales espacios se eliminan al recuperarse el dato. Un char(10) ocupa 10 bytes, pues al ser fija su longitud, no necesita ese byte adicional donde guardar la longitud. Por ello, si la longitud es invariable, es conveniente utilizar el tipo char; caso contrario, el tipo varchar.

Ocupa tantos bytes como se definen con el argumento “x”. Si ingresa un argumento mayor al permitido (255) aparece un mensaje indicando que no se permite y sugiriendo que use “blob” o “text”. Si omite el argumento, coloca 1 por defecto.

- **blob o text:** Bloques de datos de 60000 caracteres de longitud aproximadamente.

2.4.11.2 Tipos de Datos Numéricos

Existe variedad de tipos numéricos para representar enteros, negativos, decimales.

Para almacenar valores enteros usamos:

- **integer(x) o int(x):** Su rango es de -2000000000 a 2000000000 aprox. El tipo “int unsigned” va de 0 a 4000000000. El tipo “integer” tiene subtipos:
 - **mediumint(x):** Va de -8000000 a 8000000 aprox. Sin signo va de 0 a 16000000 aprox.
 - **smallint(x):** Va de -30000 a 30000 aprox., sin signo, de 0 a 60000 aprox.

- **tinyint(x):** Define un valor entero pequeño, cuyo rango es de -128 a 127. El tipo sin signo va de 0 a 255.
- **bool o boolean:** Sinónimos de tinyint(1). Un valor cero se considera falso, los valores distintos de cero, verdadero.
- **bigint(x):** Es un entero largo. Va de -9000000000000000000 a 9000000000000000000 aprox. Sin signo es de 0 a 10000000000000000000 .

Para almacenar valores con decimales utilizamos:

- **float (t,d):** Número de coma flotante. Su rango es de $-3.4e+38$ a $-1.1e-38$ (9 cifras).
- **decimal o numeric (t,d):** El primer argumento indica el total de dígitos y el segundo, la cantidad de decimales. El rango depende de los argumentos, también los bytes que ocupa. Si queremos almacenar valores entre 0.00 y 99.99 debemos definir el campo como tipo “decimal (4,2)”. Si no se indica el valor del segundo argumento, por defecto es 0. Para los tipos “float” y “decimal” se utiliza el punto como separador de decimales.

Todos los tipos enteros pueden tener el atributo “unsigned”, esto permite sólo valores positivos y duplica el rango. Los tipos de coma flotante

también aceptan el atributo “unsigned”, pero el valor del límite superior del rango no se modifica.

2.4.11.3 Tipos de Datos Fechas y Horas

Para guardar fechas y horas dispone de varios tipos:

- **date:** Representa una fecha con formato “YYYY-MM-DD”. El rango va de “1000-01-01” a “9999-12-31”.
- **datetime:** Almacena fecha y hora, su formato es “YYYY-MM-DD HH:MM:SS”. El rango es de “1000-01-01 00:00:00” a “9999-12-31 23:59:59”.
- **time:** Almacena una hora. Su formato es “HH:MM:SS”. El rango va de “-838:59:59” a “838:59:59”.
- **year(2) y year(4):** Almacena un año. Su formato es “YYYY” o “YY”. Permite valores desde 1901 a 2155 (en formato de 4 dígitos) y desde 1970 a 2069 (en formato de 2 dígitos).

Si ingresamos los valores como cadenas, un valor entre “00” y “69” es convertido a valores “year” en el rango de 2000 a 2069; si el valor está entre “70” y “99”, se convierten a valores “year” en el rango 1970 a 1999.

Si ingresamos un valor numérico 0, se convierte en “0000”; entre 1 y 69, se convierte a valores “year” entre 2001 a 2069; entre 70 y 99, es convertido a valores “year” de 1970 a 1999.

Para almacenar valores de tipo fecha se permiten como separadores “/”, “-” y “.”.

Si ingresamos ‘06-12-31’ (año de 2 dígitos), lo toma como ‘2006-12-31’.

Si ingresamos ‘2006-2-1’ (mes y día de 1 dígito), lo toma como ‘2006-02-01’.

Si ingresamos ‘20061231’ (cadena sin separador), lo toma como ‘2006-12-31’.

Si ingresamos 20061231 (numérico), lo toma como ‘2006-12-31’.

Si ingresamos ‘20061231153021’ (cadena sin separadores), lo toma como ‘2006-12-31 15:30:21’.

Si ingresamos ‘200612311530’ (cadena sin separadores con un dato faltante) no lo reconoce como fechahora y almacena ceros.

Si ingresamos ‘2006123’ (cadena sin separadores con un dato faltante) no lo reconoce como fecha y almacena ceros.

Si ingresamos '2006-12-31 11:30:21' (valor date time) en un campo 'date', toma sólo la parte de la fecha, la hora se corta, se guarda '2006-12-31'.

2.4.12 Valores por defecto

Para campos de cualquier tipo no declarados "not null" el valor por defecto es "null".

Para campos declarados "not null", el valor por defecto depende del tipo de dato. Para cadenas de caracteres el valor por defecto es una cadena vacía. Para valores numéricos el valor por defecto es 0; en caso de ser "auto_increment" es el valor mayor existente +1 comenzando en 1.

Para todos los tipos, excepto "blob", "text" y "auto_increment" se pueden explicitar valores por defecto con la cláusula "default".

Un valor por defecto se inserta cuando no está presente al ingresar un registro y en algunos casos en que el dato ingresado es inválido.

Los campos para los cuales no se ingresaron valores tomarán los valores por defecto según el tipo de dato del campo.

2.4.13 Valores inválidos

Un valor por defecto se inserta cuando no está presente al ingresar un registro y en algunos casos en que el dato ingresado es inválido.

Un valor es inválido por tener un tipo de dato incorrecto para el campo o por estar fuera de rango.

Veamos los distintos tipos de datos inválidos.

Para campos de tipo carácter:

- **valor numérico:** Si en un campo definido de tipo carácter ingresamos un valor numérico, lo convierte automáticamente a cadena.
- **mayor longitud:** Si intentamos guardar una cadena de caracteres mayor a la longitud definida, la cadena se corta guardando sólo la cantidad de caracteres que quepa.

Para campos numéricos:

- **cadena:** Si en un campo numérico ingresamos una cadena, lo pasa por alto y coloca 0. Por ejemplo, si en un campo de tipo “integer” guardamos ‘abc’, almacenará 0.

- **valores fuera de rango:** Si en un campo numérico intentamos guardar un valor fuera de rango, se almacena el valor límite del rango más cercano (menor o mayor).
- **valores incorrectos:** Si cargamos en un campo definido de tipo decimal un valor con más decimales que los permitidos en la definición, el valor es redondeado al más cercano.

Para campos definidos `auto_increment` el tratamiento es el siguiente:

- Pasa por alto los valores fuera del rango, 0 en caso de no ser “unsigned” y todos los menores a 1 en caso de ser “unsigned”.
- Si ingresamos un valor fuera de rango continúa la secuencia.
- Si ingresamos un valor existente, aparece un mensaje de error indicando que el valor ya existe.

Para campos de fecha y hora:

- **valores incorrectos:** Si intentamos almacenar un valor que MySQL no reconoce como fecha (sea fuera de rango o un valor inválido), convierte el valor en ceros (según el tipo y formato).

Para campos de cualquier tipo:

- **valor “null”**: Si un campo está definido “not null” e intentamos ingresar “null”, aparece un mensaje de error y la sentencia no se ejecuta.

2.4.14 Operadores Lógicos

Se Puede establecer más de una condición con la cláusula “WHERE”, haciendo uso de los operadores lógicos.

- **and**, significa “y”
- **or**, significa “y/o”
- **xor**, significa “o”
- **not**, significa “no”, invierte el resultado
- **()**, paréntesis

Los operadores lógicos se usan para combinar condiciones.

Los registros recuperados en una sentencia que une 2 condiciones con el operador “AND”, cumplen con las 2 condiciones.

Los registros recuperados con una sentencia que une 2 condiciones con el operador “OR”, cumplen 1 de las condiciones o ambas.

Los registros recuperados con una sentencia que une 2 condiciones con el operador “XOR”, cumplen 1 de las condiciones, no ambas.

El operador “NOT” invierte el resultado de la condición a la cual antecede.

Los registros recuperados en una sentencia en la cual aparece el operador “NOT”, no cumplen con la condición a la cual afecta el “NO”.

Los paréntesis se usan para encerrar condiciones, para que se evalúen como una sola expresión.

Cuando explicitamos varias condiciones con diferentes operadores lógicos (combinamos “AND”, “OR”) permite establecer el orden de prioridad de la evaluación; además permite diferenciar las expresiones más claramente.

Si bien los paréntesis no son obligatorios en todos los casos, se recomienda utilizarlos para evitar confusiones.

El orden de prioridad de los operadores lógicos es el siguiente: “NOT” se aplica antes que “AND” y “AND” antes que “OR”, si no se especifica un orden de evaluación mediante el uso de paréntesis.

El orden en el que se evalúan los operadores con igual nivel de precedencia es indefinido, por ello se recomienda usar los paréntesis.

2.5 Programación Web

Para realizar una página con la programación Web, se deben tener claros, tres conceptos fundamentales los cuales son, el URL (Uniform Resource Locators), es un sistema con el cual se localiza un recurso dentro de la red, este recurso puede ser una página web, un servicio o cualquier otra cosa. En resumen el URL no es más que un nombre, que identifica una computadora, dentro de esa computadora un archivo que indica el camino al recurso que se solicita.

El siguiente concepto dentro de la programación Web, es el protocolo encargado de llevar la información que contiene una página Web por toda la red de internet, como es el HTTP (Hypertext Transfer Protocol).

Y por último el lenguaje necesario cuya funcionalidad es la de representar cualquier clase de información que se encuentre almacenada en una página Web, este lenguaje es el HTML (Hypertext Markup Language).

En la programación Web, el HTML es el lenguaje que permite codificar o preparar documentos de hipertexto, que viene a ser el lenguaje común para la construcción de una página Web.

Con el comienzo del internet y la programación web, se desfasaron los diseños gráficos tradicionales, con lo que se empezaron a diseñar interfaces concretas para este medio, buscando ficheros pequeños para facilitar la carga de los mismos. La programación web se orientaba a un diseño muy cargado e

interactuando con el usuario, mientras que al empezar a competir con millones de webs se ha optado más por el diseño sencillo y de fácil comprensión.

En programación web se creó la necesidad de conocer a fondo diferentes lenguajes de programación como HTML, JavaScript y DHTML.

2.5.1 Lenguajes de Programación Web

El navegador es una aplicación capaz de interpretar las órdenes recibidas en forma de código HTML fundamentalmente y convertirlas en las páginas que son el resultado de dicha orden.

En la *Figura 2.1.* se muestra el funcionamiento de las páginas dinámicas usando aplicaciones del lado del cliente.

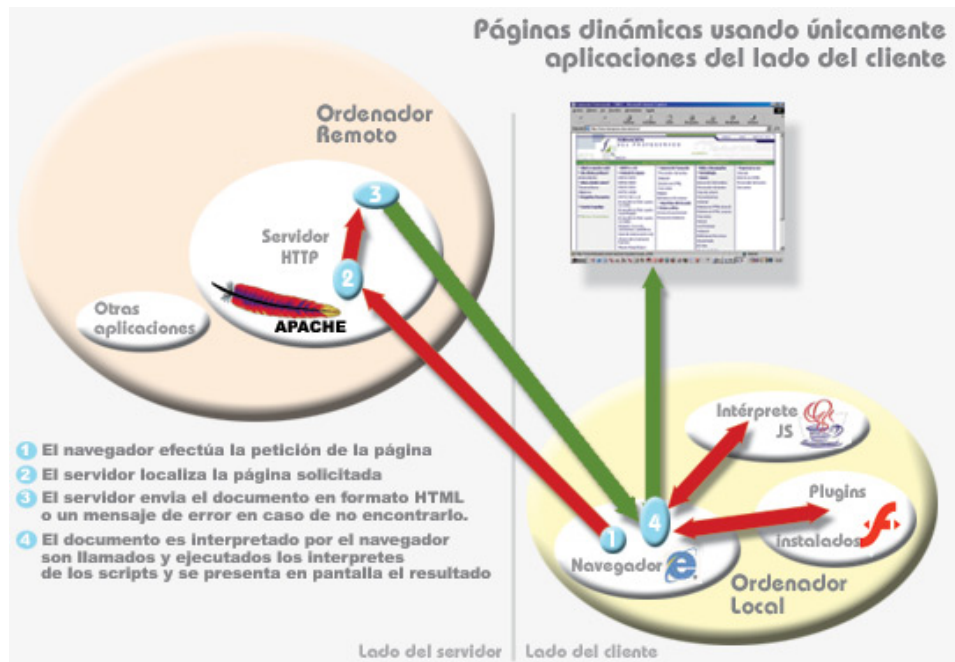


Figura 2.1. Aplicaciones del Lado del Cliente

Al hacer uso de un enlace hipertexto en una página, se establece una petición de un archivo HTML residente en el servidor (un ordenador que se encuentra continuamente conectado a la red) el cual es enviado e interpretado por nuestro navegador (el cliente).

Así pues, podemos hablar de lenguajes de lado servidor que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Por otro lado, los lenguajes de lado cliente (entre los cuales no sólo se encuentra el HTML sino también el Java y el JavaScript los cuales son simplemente incluidos en el código HTML) son aquellos que pueden ser directamente “digeridos” por el navegador y no necesitan un pretratamiento.

En la *Figura 2.2.* se muestra el funcionamiento de las páginas dinámicas usando aplicaciones del lado del servidor y del cliente.

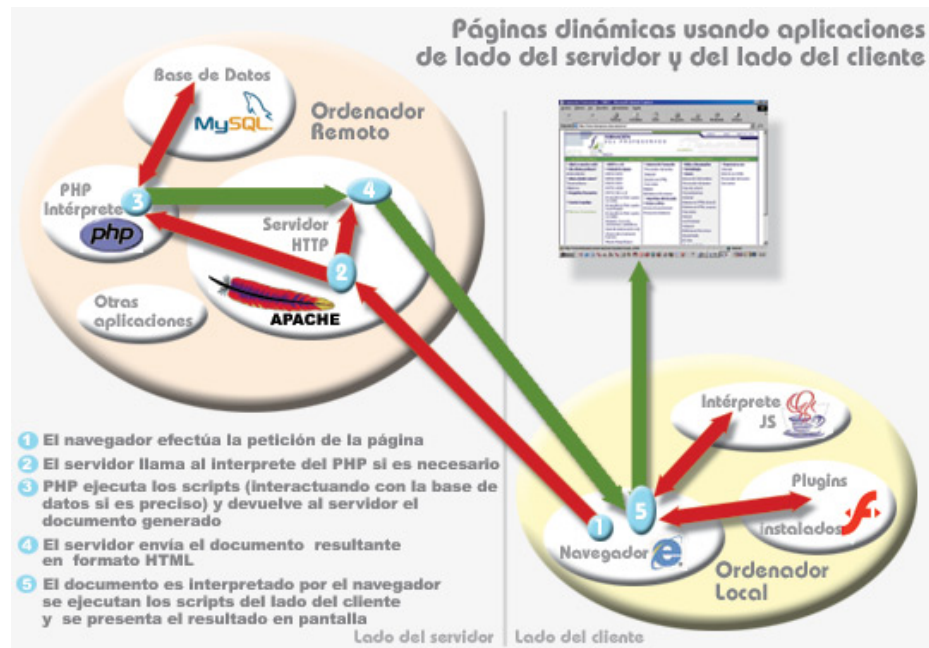


Figura 2.2. Aplicaciones del Lado del Servidor y Cliente

Cada uno de estos tipos tiene por supuesto ventajas y desventajas. Así, por ejemplo, un lenguaje de lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. Inversamente, un lenguaje de lado servidor es independiente del cliente por lo que es mucho menos rígido respecto al cambio de un navegador a otro o respecto a las versiones del mismo.

2.5.1.1 Lenguajes del Lado del Cliente

- **HTML(Hypertext Markup Language)**

Lenguaje de Marcación de Hipertexto, es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). Fue creado tomando dos herramientas preexistentes: El concepto de Hipertexto (Conocido también como link o ancla) el cual permite conectar dos elementos entre sí y el SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizará en la forma como éste lo entienda.

El entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrecen los sistemas operativos Windows (Bloc de notas), UNIX (el editor vi o ed) o el que ofrece MS Office (Word). El conjunto de etiquetas que se creen, se deben guardar con la extensión .htm o .html.

Estos documentos pueden ser mostrados por los visores o “browsers” de paginas Web en Internet, como Netscape Navigator, Mosaic, Opera y Microsoft Internet Explorer.

También existe el HTML Dinámico (DHTML), que es una mejora de Microsoft de la versión 4.0 de HTML que le permite crear efectos especiales como, por ejemplo, texto que vuela desde la página palabra por palabra o efectos de transición al estilo de anuncio publicitario giratorio entre página y página.

El lenguaje llamado HTML indica al navegador donde colocar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página.

El lenguaje consta de etiquetas que tienen esta forma o <P>. Cada etiqueta significa una cosa, por ejemplo significa que se escriba en negrita (bold) o <P> significa un párrafo, <A> es un enlace, etc. Casi todas las etiquetas tienen su correspondiente etiqueta de cierre, que indica que a partir de ese punto no debe de afectar la etiqueta. Por ejemplo se utiliza para indicar que se deje de escribir en negrita. Así que el HTML no es más que una serie de etiquetas que se utilizan para definir la forma o estilo que queremos aplicar a nuestro documento. Esto está en negrita.

- **JAVASCRIPT**

Javascript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del

cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Entre las acciones típicas que se pueden realizar en Javascript tenemos dos: Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, javascript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Además, Javascript pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente.

- **APPLETS DE JAVA**

Es otra manera de incluir código a ejecutar en los clientes que visualizan una página web. Se tratan de programas hechos en Java, que se transfieren con las páginas web y que el navegador ejecuta en el espacio de la página.

Los applets de Java están programados en Java y precompilados, es por ello que la manera de trabajar de éstos varía un poco con respecto a los lenguajes de script como Javascript. Los applets son más difíciles de programar que los scripts en Javascript y requieren conocimientos básicos o medios del lenguaje Java.

La principal ventaja de utilizar applets consiste en que son mucho menos dependientes del navegador que los scripts en Javascript, incluso independientes del sistema operativo del ordenador donde se ejecutan. Además, Java es más potente que Javascript, por lo que el número de aplicaciones de los applets podrá ser mayor.

Como desventajas en relación con Javascript cabe señalar que los applets son más lentos de procesar y que tienen espacio muy delimitado en la página donde se ejecutan, es decir, no se mezclan con todos los componentes de la página ni tienen acceso a ellos. Es por ello que con los applets de Java no podremos hacer directamente cosas como abrir ventanas secundarias, controlar Frames, formularios, capas, etc.

Java es compatible con todos los sistemas porque basa su funcionamiento en los Byte Codes, que no es más que una precompilación del código fuente de Java.

Estos Byte Codes no son el programa en Java propiamente dicho, sino un archivo que contiene un código intermedio que puede manejar la Máquina Virtual de Java. Cada sistema operativo dispone de una Máquina Virtual de Java que puede interpretar los Byte Codes y transformarlos a sentencias ejecutables en el sistema en cuestión.

- **VBSCRIPT (Visual Basic Script)**

Es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer. Es por ello que su utilización está relegada al uso de Javascript.

Está basado en Visual Basic, lenguaje para crear aplicaciones Windows. Tanto su sintaxis como la manera de trabajar están inspiradas en él. Sin embargo, no todo lo que se puede hacer en Visual Basic lo podremos hacer en Visual Basic Script, pues este último es una versión reducida del primero.

El modo de funcionamiento de VBScript para construir efectos especiales en páginas web es muy similar al utilizado en Javascript y los recursos a los que se puede acceder también son los mismos.

- **CSS(Cascading Style Sheets)**

Hojas de Estilo en Cascada, es una tecnología que nos permite crear páginas web de una manera más exacta. Gracias a las CSS se pueden manipular los resultados finales de una página, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores, etc. Incluso podemos definir nuestros propios estilos en un archivo externo a nuestras páginas; así, si en algún momento queremos cambiar alguno de ellos, automáticamente se nos actualizarán todas las páginas vinculadas de nuestro sitio.

2.5.1.2 Lenguajes del Lado del Servidor

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante que contiene únicamente código HTML, que es compatible con todos los navegadores.

- **CGI(Common Gateway Interface)**

La Interfaz Común de Pasarela, es una importante tecnología de la World Wide Web que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa.

Las aplicaciones CGI fueron una de las primeras maneras prácticas de crear contenido dinámico para las páginas web. En una aplicación CGI, el servidor web pasa las solicitudes del cliente a un programa externo. La salida de dicho programa es enviada al cliente en lugar del archivo estático tradicional.

CGI ha hecho posible la implementación de funciones nuevas y variadas en las páginas web, de tal manera que esta interfaz rápidamente se volvió un estándar, siendo implementada en todo tipo de servidores web.

Los CGI se escriben habitualmente en el lenguaje Perl, sin embargo, otros lenguajes como C, C++ o Visual Basic pueden ser también empleados para construirlos.

El funcionamiento básico de un programa CGI es parecido al del conjunto de las páginas dinámicas del servidor, con algunas particularidades.

1. Se realiza una petición http, a la que pueden acompañar datos llegados o bien por un formulario o bien a través de la URL.
2. El servidor ejecuta los programas CGI a los que se accede y trabaja con los recursos necesarios para llevar a cabo las acciones, como por ejemplo bases de datos.
3. El programa CGI va escribiendo en la salida estándar el resultado de la ejecución del CGI, que incluye etiquetas HTML, ya que lo que se escribe es una página web.

Algunas desventajas de la programación en CGI son las siguientes:

1. Los resultados se escriben directamente con el CGI, así que el código del programa se mezcla con el del HTML haciendo difícil su comprensión y mantenimiento.
2. Cada programa CGI que se pone en marcha lo hace en un espacio de memoria propio. Así, si tres usuarios ponen en marcha un CGI a la vez se multiplicará por tres la cantidad de recursos que ocupe ese CGI. Esto significa una grave ineficiencia.

- **PERL(Practical Extracting and Reporting Lenguaje)**

El Lenguaje Práctico de Extracción y Reporte, es un lenguaje de programación muy utilizado para construir aplicaciones CGI para el web. Se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.

Es un lenguaje libre de uso, eso quiere decir que es gratuito. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows.

Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como Javascript o ASP(Active Server Pages). Esto quiere decir que el código de los scripts en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito. Además es extensible a partir de otros lenguajes, ya que desde Perl podremos hacer llamadas a subprogramas escritos en otros lenguajes. También desde otros lenguajes podremos ejecutar código Perl.

Las ventajas principales son el que se trata de un lenguaje muy maduro, que lleva mucho tiempo funcionando, y que tiene cientos de bibliotecas operativas y listas para ser usadas.

Perl debe ser usado precompilado vía ModPerl o el impacto en procesador es muy alto.

- **ASP(Active Server Pages)**

Las Páginas de Servidor Activo, son una tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft).

Las páginas pueden ser escritas en VBScript que es a su vez un derivado de Visual Basic. Las páginas ASP pueden hacer uso de objetos COM (Component Object Model) que son objetos en algún otro lenguaje (ej.: ejecutables en C++ o Java); de manera que si ya se tiene algo programado las páginas ASP a través del IIS(Internet Information Services) pueden hacer uso de los métodos en estos objetos.

Para conectarse a una base de datos, normalmente se utiliza ADO(ActiveX Data Objects) que es un adaptador universal a bases de datos que se especializa posteriormente para hablar con una base de datos concreta. El esquema de trabajo es crear objetos que ejecutan la lógica de la aplicación (Business Objects) y luego hacer la capa de interfaz con ASP(Active Server Pages).

- **PHP(Hipertext Preprocesor)**

PHP es un pre-procesador de páginas HTML que está motivado y dirigido expresamente a construir páginas. Los argumentos CGI son importados directamente al espacio de nombres global de cualquier programa PHP, PHP puede correr como un CGI o como un plug-in del servidor web.

Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

En la *Figura 2.3.* se muestra el esquema de funcionamiento de las páginas PHP.

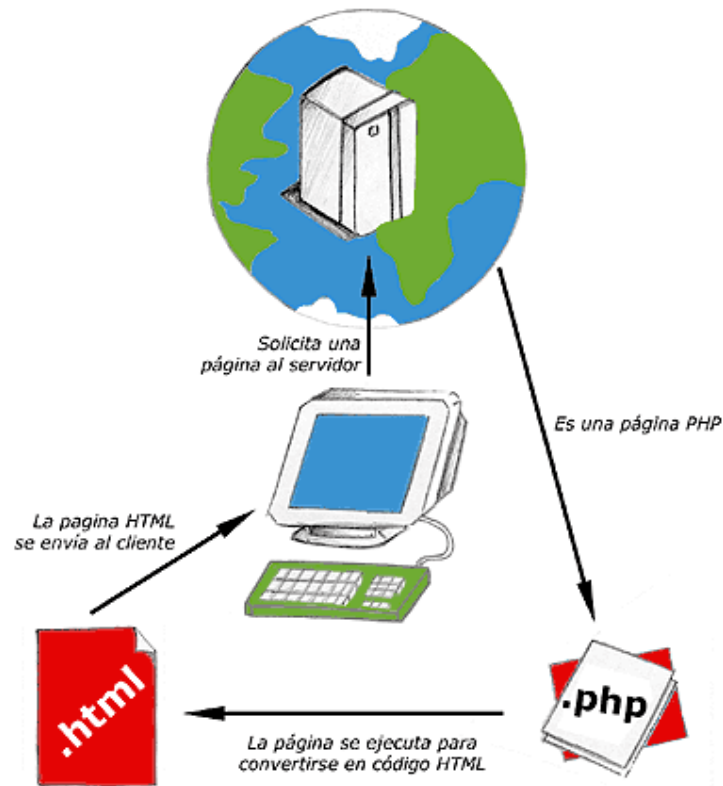


Figura 2.3. Esquema del funcionamiento de las páginas PHP

- **JSP(Java Server Pages)**

Las Páginas de Servidor Java, son una tecnología orientada a crear páginas web con programación en Java.

Con JSP podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual.

JSP originado por SUN, provee de un lenguaje de scripting en el lado del servidor que se comunica con clases Java, objetos RMI, CORBA, etc. La metodología de trabajo es la misma que con Visual Basic con la diferencia de que es una plataforma mucho más abierta.

El código JSP se puede poner dentro de las páginas HTML, o se puede precompilar en Servlets (programas en Java también pueden ser compilados en Servlets). Un servlet es una subclase de servlet que tiene métodos para atender requerimientos. El servidor pasa las variables hacia y desde el ambiente de ejecución del Servlet.

Para programar en JSP se requiere conocer Java, ser metódico y ordenado. El lenguaje no relaja las condiciones de tipado que le son propias.

2.5.1.3 Lenguajes del Lado Cliente-Servidor

- **DHTML(Dynamic HTML)**

DHTML no es precisamente un lenguaje de programación, se trata más bien de una nueva capacidad de la que disponen los navegadores modernos, mediante la cual se puede tener un mayor control sobre la página.

Cualquier página que responde a las actividades del usuario y realiza efectos y funcionalidades se puede englobar dentro del DHTML, pero en este caso nos referimos más a efectos en el navegador por los cuales se pueden mostrar y ocultar elementos de la página, se puede modificar su posición, dimensiones, color, etc.

DHTML nos da más control sobre la página, gracias a que los navegadores modernos incluyen una nueva estructura para visualizar en páginas web denominada capa, las cuales se pueden ocultar, mostrar, desplazar, etc.

Para realizar las acciones sobre la página, como modificar la apariencia de una capa, es necesario un lenguaje de programación del lado del cliente como Javascript o VBScript.

En la actualidad, DHTML engloba también la programación en el servidor.

Dentro del concepto de DHTML se engloban también las Hojas de Estilo en Cascada o CSS (Cascade Style Sheets).

- **XML(Extensible Markup Language)**

El Lenguaje de Marcación Extensible, es una tecnología en realidad muy sencilla que tiene a su alrededor a otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores.

XML, con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, permite compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes.

XML permite al programador dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de estos o el recorrido de las estructuras están a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello.

2.6 Lenguaje de Programación PHP

PHP es un lenguaje de scripting que permite la generación dinámica de contenidos en un servidor web. El significado de sus siglas es HyperText Preprocessor. Entre sus principales características cabe destacar su potencia, su alto rendimiento, su facilidad de aprendizaje y su escasez de consumo de recursos.

El funcionamiento de las páginas en PHP alojadas en un servidor es el siguiente: (Ver Figura 2.4.)

- El navegador del cliente solicita el documento PHP.
- Llega la solicitud del servidor y el servidor localiza el documento, lanza el intérprete de PHP y ejecuta todo su código.
- Una vez ejecutado el código se genera el resultado en HTML y lo devuelve al servidor para que lo transfiera al cliente.
- El servidor transfiere el resultado en HTML y es mostrado en el navegador del cliente.



Figura 2.4. Funcionamiento Lenguaje PHP

El lenguaje PHP es un lenguaje de programación de estilo clásico, es decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones, etc.; no es un lenguaje de marcas como podría ser HTML, XML o WML(Wireless Markup Language). Está más cercano a JavaScript o a C, pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el

servidor como podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador. Este resultado es normalmente una página HTML pero igualmente podría ser una página WML.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

2.6.1 Aplicaciones y Ventajas

Existen tres campos en los que scripts escritos en PHP son usados:

- **Scripts en la parte del servidor.** Este es el campo más tradicional y el principal campo de trabajo. Se necesitan tres cosas para que esto funcione. El parseador PHP (CGI ó módulo), un servidor web y un navegador. Se necesita correr el servidor web con PHP instalado. El resultado del programa PHP se puede obtener a través del navegador, conectando con el servidor web.
- **Scripts en línea de comandos.** Se puede crear un script PHP y correrlo sin ningún servidor web ó navegador. Solamente se necesita el parseador PHP para usarlo de esta manera. Este tipo de uso es ideal para scripts ejecutados regularmente desde cron (en Unix ó Linux) ó el Planificador

de tareas (en Windows). Estos scripts también pueden ser usados para tareas simples de procesado de texto.

- **Escribir aplicaciones gráficas clientes.** PHP no es probablemente el mejor lenguaje para escribir aplicaciones gráficas, pero PHP, tiene algunas características avanzadas en programas clientes, utilizando PHP-GTK para escribir dichos programas. Es también posible escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal.

PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluido HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más.

PHP soporta la mayoría de servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, O'Reilly Website Pro server, Caudium, Xitami, OmniHTTPd y muchos otros. PHP tiene módulos disponibles para la mayoría de los servidores, para aquellos otros que soporten el estándar CGI, PHP puede usarse como procesador CGI.

- **Soporte para una gran cantidad de bases de datos.** Quizás su característica más potente y destacable, escribir un interfaz vía web para una base de datos es una tarea simple con PHP.

PHP también tiene soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También se pueden crear raw sockets. PHP soporta WDDX para intercambio de datos entre lenguajes de programación en web. Y hablando de interconexión, PHP puede utilizar objetos Java de forma transparente como objetos PHP Y la extensión de CORBA puede ser utilizada para acceder a objetos remotos.

2.6.2 Introducción al Lenguaje PHP

El código PHP puede incluirse dentro del código html de la página. Para delimitar la sección de código PHP podemos hacerlo de varias formas:

- Usando las etiquetas `<?php` y `?>`
- Usando las etiquetas `<? y ?>`
- Mediante `<script lenguaje="php">` y `</script>`

La separación de instrucciones se hace de la misma manera que en C o Perl, terminando cada declaración con un punto y coma.

PHP soporta el estilo de comentarios de 'C', 'C++' y de la interfaz de comandos de Unix, `"//Comentario"`, `"/*Comentario*/"` y `"#Comentario"`. Los estilos de comentarios de una línea actualmente sólo comentan hasta el final de la línea o el bloque actual de código PHP, lo primero que ocurra.

2.6.3 Variables

Una variable podría definirse como una posición de memoria creada para introducir o asignar cualquier valor o dato. Durante la ejecución del script el valor de la variable puede "variar"(valga la redundancia) tanto de tipo como de valor. En PHP (al igual que en JavaScript) no hace falta declarar la variable ya que simplemente anteponiendo el carácter \$ al nombre de la variable estamos indicando que es una variable.

Otro hecho que cabe destacar a la hora de programar en PHP y la declaración de variables es que PHP es un lenguaje "CASE SENSITIVE" es decir que diferencia entre mayúsculas y minúsculas y debido a esta razón no sería lo mismo \$Variable=valor; que \$VaRiABle=valor; ya que PHP lo interpretaría como dos variables completamente diferentes.

2.6.4 Tipos de Datos

Los tipos de cada variable en PHP no están tan claros como en C. El intérprete asigna el tipo de una variable según el uso que se esté haciendo de ella. Para asignar un tipo fijo a una variable se utiliza la función settype().

Los tipos son:

1. Enteros
2. Flotantes
3. String
4. Arrays
5. Objetos
6. Juggling

2.6.4.1 Enteros

Los enteros se pueden especificar usando una de las siguientes sintaxis:

`$a = 1234;` # número decimal

`$a = -123;` # un número negativo

`$a = 0123;` # número octal (equivalente al 83 decimal)

`$a = 0x12;` # número hexadecimal (equivalente al 18 decimal)

2.6.4.2 Números en punto flotante

Los números en punto flotante ("double") se pueden especificar utilizando cualquiera de las siguientes sintaxis:

`$a = 1.234;` `$a = 1.2e3;`

2.6.4.3 String

Las cadenas pueden estar delimitadas por comillas dobles (") o comillas simples ('). Si la cadena está delimitada por comillas dobles, cualquier variable incluida dentro de ella será sustituida por su valor. El carácter de barra invertida ("\") se puede usar para especificar caracteres especiales

Las cadenas se pueden concatenar usando el operador '.' (punto).

2.6.4.4 Arrays

Los Arrays en PHP se pueden utilizar tanto como Arrays indexados o como Arrays asociativos. Los Arrays de una sola dirección, pueden ser tanto escalares como asociativos. En realidad no existe ninguna diferencia entre ellos. Las funciones que se utilizan para crear Arrays de este tipo son list() o array() . En el caso de que no se especifique el índice en un array, el elemento que se asigna se añade al final.

Los arrays pueden ser ordenados utilizando las siguientes funciones: asort(), arsort(), ksort(), rsort(), sort(), uasort(), usort() y uksort() dependiendo del tipo de ordenación deseada.

Se puede contar el número de elementos de un array usando la función count().

Se puede recorrer un array usando las funciones `next()` y `prev()`. Otra forma habitual de recorrer un array es usando la función `each()`.

En PHP, los arrays multidimensionales combinan las propiedades de un array unidimensional. Los índices de un array multidimensional pueden ser tanto numéricos como asociativos.

2.6.4.5 Objetos

Para inicializar un objeto se utiliza el método `new`, y para acceder a cada uno de sus métodos se utiliza el operador `->`.

2.6.4.6 Juggling

PHP no requiere (o soporta) la declaración explícita del tipo en la declaración de variables, define su tipo según el contenido y el contexto en el que se utilice, es decir, si se asigna una cadena a una variable, el tipo de esa variable será `string`. Si a esa misma variable se le asigna un número, el tipo cambiará a `entero`. Para asegurar de que una variable es del tipo adecuado se utiliza la función `settype()`. Para obtener el tipo de una variable se utiliza la función `gettype()`. También es posible utilizar el mecanismo del `casting` tal y como se utiliza en C.

2.6.5 Operadores

2.6.5.1 Aritméticos

Los operadores de PHP son muy parecidos a los de C y JavaScript, si se conocen estos lenguajes, resultaran familiares y fáciles de reconocer.

Estos son los operadores que se pueden aplicar a las variables y constantes numéricas. (Ver Tabla 2.1.)

| Operador | Nombre | Ejemplo | Descripción |
|----------|----------------|---------|---|
| + | Suma | 5 + 6 | Suma dos números |
| - | Resta | 7 - 9 | Resta dos números |
| * | Multiplicación | 6 * 3 | Multiplica dos números |
| / | División | 4 / 8 | Divide dos números |
| % | Módulo | 7 % 2 | Devuelve el resto de dividir ambos números, en este ejemplo el resultado es 1 |
| ++ | Suma 1 | \$a++ | Suma 1 al contenido de una variable. |
| -- | Resta 1 | \$a-- | Resta 1 al contenido de una variable. |

Tabla 2.1. Operadores Aritméticos en PHP

2.6.5.2 Comparación

Los operadores de comparación son usados para comparar valores y así poder tomar decisiones. (Ver Tabla 2.2.)

| Operador | Nombre | Ejemplo | Devuelve cierto cuando: |
|----------|---------------|------------|------------------------------|
| == | Igual | \$a == \$b | \$a es igual \$b |
| != | Distinto | \$a != \$b | \$a es distinto \$b |
| < | Menor que | \$a < \$b | \$a es menor que \$b |
| > | Mayor que | \$a > \$b | \$a es mayor que \$b |
| <= | Menor o igual | \$a <= \$b | \$a es menor o igual que \$b |
| >= | Mayor o igual | \$a >= \$b | \$a es mayor o igual que \$b |

Tabla 2.2. Operadores de Comparación en PHP

2.6.5.3 Lógicos

Los operadores lógicos son usados para evaluar varias comparaciones, combinando los posibles valores de estas. (Ver Tabla 2.3.)

| Operador | Nombre | Ejemplo | Devuelve cierto cuando: |
|----------|--------|-------------|---|
| && | Y | \$a && \$b | Devuelve verdadero cuando ambas condiciones son verdaderas. |
| And | Y | \$a and \$b | Devuelve verdadero cuando ambas condiciones son verdaderas. |
| | O | \$a \$b | Devuelve verdadero cuando al menos una de las dos es verdadera. |
| Or | O | \$a or \$b | Devuelve verdadero cuando al menos una de las dos es verdadera. |
| ! | No | !\$a | Niega el valor de la expresión. |

Tabla 2.3. Operadores Lógicos en PHP

2.6.6 Estructuras de Control

2.6.6.1 Estructura IF

IF es una estructura de control utilizada para tomar decisiones según se cumpla una condición (o varias) o no. Su estructura básica es la siguiente:

```

if(condición/es){
    Acción a realizar;
}
else
{
    Acción a realizar en caso de que no se cumpla;
}

```

2.6.6.2 Estructura SWITCH

Toma distintas decisiones en función de distintos estados de una variable.

Su sintaxis es la siguiente:

```
switch(expresión){  
  
  case valor1:  
    Sentencia a ejecutar cuando la expresión tiene como valor  
    valor1;  
    break;  
  
  case valor2:  
    Sentencia a ejecutar cuando la expresión tiene como valor  
    valor2;  
    break;  
  
  case valor3:  
    Sentencia a ejecutar cuando la expresión tiene como valor  
    valor3;  
    break;  
  
  default:  
    Sentencia que se ejecuta por defecto cuando no se cumple ninguna de  
    las condiciones anteriores;  
    break;  
  
}
```

2.6.6.3 Bucle FOR

Se usa para repetir una misma operación un número determinado de veces.

Su sintaxis es la siguiente:

```
for(inicialización;condición;actualización){  
    Sentencia a ejecutar mientras se cumpla la condición;  
}
```

El bucle for esta compuesto de 3 partes:

1. Inicialización: Se ejecuta tan solo al iniciar por primera vez el bucle. En esta parte se suele colocar la variable que contara el número de veces que se repite el bucle.
2. Condición: Es la condición que se evaluara cada vez que se inicie el bucle. Esta condición es la que determina la duración del bucle.
3. Actualización: Sirve para indicar los cambios que queremos ejecutar en las variables cada vez que se ejecuta el bucle.

Un ejemplo de su uso seria el siguiente:

```
for($i=1;i<=10;i++){  
    echo "El número actual es".$i;  
}
```

De esta forma escribiría todos los números contenidos entre 1 y 10.

2.6.6.4 Bucle WHILE

Este bucle se usa cuando queremos repetir la ejecución de unas sentencias un número indefinido de veces. Su sintaxis es la siguiente:

```
while(condición){  
    Sentencia a ejecutar;  
}
```

2.6.6.5 Bucle DO...WHILE

Este bucle se usa cuando no sabemos el número de veces que va a ejecutarse un bucle pero lo que si tenemos claro es que por lo menos una vez si que se ejecutara la acción. Su sintaxis es la siguiente:

```
do{  
    Sentencia del bucle;  
}while(condición)
```

2.6.6.6 Break

Se usa para detener el bucle y dejar de interpretar el código que sigue después del break.

2.6.6.7 Continue

Sirve para volver al principio del bucle desde cualquier parte del bucle.

2.6.7 Funciones en PHP

Una de las herramientas más importantes en cualquier lenguaje de programación son las funciones. Una función consiste en un conjunto de rutinas y acciones que a lo largo del script van a ser ejecutadas multitud de veces agrupados en una FUNCION y desde cualquier punto del script puede ser llamada y ejecutada. A su vez, esta función puede recibir parámetros externos de los cuales dependa el resultado de una función.

Las funciones deben ser colocadas siempre antes de realizar la llamada a la función. La sintaxis de una función es la siguiente:

```
function nombre(parámetros){  
    Instrucciones de la función;  
return valor;  
}
```

Opcionalmente podemos pasarle parámetros a las funciones que se trataran como variable locales y así mismo podemos devolver un resultado con la instrucción “return valor”.

Para llamar a la función sería de la siguiente forma:

nombre(parámetros);

Un hecho relevante que cabe destacar es que las variables que declaremos dentro de la función solo existirán o tendrán dicho valor dentro de la función.

Existen casos en los cuales no sabemos el número de parámetros que le pasaremos a la función y en estos casos debemos usar las funciones creadas al efecto como son:

func_num_args() Numero de parámetros que se le han pasado a la función

func_get_args() Devuelve un elemento de los que forman la lista de argumentos.

2.6.8 PHP y MySQL

2.6.8.1 Conexión con MySQL

MySQL es uno de los gestores de bases de datos más utilizados en entornos en los cuales se emplea PHP ya que PHP dispone de numerosas funciones que se compaginan perfectamente con MySQL.

La forma genérica de obtener información de tablas en Mysql es la siguiente:

- Conexión con el gestor.
- Preparación de la consulta SQL.
- Ejecución de la consulta.
- Procesamiento del resultado obtenido en el cursor.
- Liberación de recursos (esta es opcional, aunque es recomendable).
- Cierre de la conexión con el gestor.

Para realizar estas y otras muchas más cosas disponemos de las siguientes funciones:

Al ejecutar la instrucción “mysql_connect” creamos un vínculo entre la base de datos y la página PHP, este vínculo será usado posteriormente en las consultas que hagamos a la base de datos.

mysql_connect: Abre una conexión al Servidor MySQL.

```
int mysql_connect(string[hostname [:port][:/path/to/socket]],  
string[username], string [password])
```

Esta función establece una conexión a un servidor MySQL. Todos los argumentos son opcionales y los valores por defecto son localhost, el usuario dueño del proceso, sin password.

Si se realiza dos veces una llamada a esta función con los mismos parámetros, devuelve el mismo identificador que para la llamada anterior.

Todas las conexiones terminan al terminar la ejecución del script, a no ser que se elimine la conexión llamando a la función `mysql_close()`.

Para realizar una consulta a la Base de Datos se usa la instrucción “`mysql_query`”.

mysql_query: Envía una consulta SQL a MySQL.

```
int mysql_query(string query, int [identificador del link] )
```

Envía una consulta a una base de datos activa en el servidor asociado al link que se especifica. En el caso de que no se especifique ninguna conexión abierta, la función intenta reestablecer una conexión con la función `mysql_connect()`.

La cadena de la consulta, debe terminar con ;.

La función devuelve TRUE o FALSE si la consulta tiene éxito. Si el valor devuelto es TRUE, significa que la consulta puede ser realizada, no que existan datos que respondan a esa consulta.

Para consultar cuantas líneas responden a esa consulta, se puede utilizar la función `mysql_affected_rows()` y nos dirá cuantas líneas se han visto afectadas por la última operación INSERT, DELETE, REPLACE o UPDATE. En el caso de tratarse de una operación SELECT la función `mysql_query()` devuelve un identificador de resultado que se puede utilizar con la función `mysql_result()`.

mysql_result: Obtiene los datos resultado de una consulta.

```
int mysql_result(int result, int row, mixed [field] )
```

Devuelve el contenido de una celda de la tabla dinámica que devuelve MySQL como resultado de una operación previa.

El último argumento es opcional y puede contener el desplazamiento del campo que se desea obtener, el nombre del campo o algo como `tabla.campo` o el alias del campo que se desee consultar si se estableció previamente.

Cuando se trabaja con amplios conjuntos de resultados, es mejor utilizar alguna de las funciones que extraen un registro de esa consulta.

Hay que tener cuidado de no hacer llamadas a funciones que obtienen resultados de `result` junto con esta función, puesto que el puntero que va leyendo de los resultados, puede perderse.

Una vez que se ha terminado de usar el vínculo con la base de datos, lo liberaremos con la instrucción “mysql_close” para que la conexión no quede ocupada.

mysql_close: Cierra la conexión MySQL.

```
int mysql_close(int [identificador de link]
```

Esta función cierra la conexión permanente que mantiene el link. El parámetro es opcional.

Si no se especifica, se cerrará la última conexión abierta. Si una conexión no se cierra, el script la cierra automáticamente al terminar su ejecución.

Esta función no opera correctamente con las conexiones abiertas con la función mysql_pconnect().

2.6.8.2 Consultas a la base de datos

Para realizar consultas a MySQL desde PHP se usan las siguientes instrucciones:

“mysql_query”, realiza la consulta a la base de datos en el lenguaje de consultas SQL.

“mysql_fetch_array”, extrae los datos de una consulta a un array.

“mysql_free_result”, libera la memoria usada por una consulta.

2.6.8.3 Otras Funciones de PHP y MySQL

La lista de funciones disponibles es la siguiente:

- **mysql_affected_rows**: Obtiene el número de registros modificados en la última operación MySQL.

```
int mysql_affected_rows(int [identificador de link])
```

Devuelve el número de líneas afectadas por la última operación, INSERT, UPDATE o DELETE, asociadas con el identificador que se especifica.

Si no se especifica el link, se asume que se refiere al último.

Excepción: Si la última operación ha borrado todos los registros de una tabla, esta función devuelve 0.

Esta función no tiene efecto con ninguna operación SELECT, puesto que no modifican ninguna línea.

- **mysql_change_user:** Modifica el usuario responsable de las operaciones.

```
int mysql_change_user(string user, string password, string[database],  
int [identificador de link])
```

Los parámetros database y link son opcionales.

Si la nueva combinación de usuario y password no es correcta, seguirá activa la actual.

- **mysql_create_db:** Crea una base de datos en el gestor de Bases de Datos.

```
int mysql_create_db(string databaseName, int [identificador del link])
```

Esta función crea una nueva base de datos asociada al link especificado.

Para mantener la coherencia con otras versiones, se utiliza `mysql_createdb()`

- **mysql_data_seek:** Realiza un movimiento en el puntero de los datos resultado.

```
int mysql_data_seek(int result_identifier, int row_number)
```

Esta función permite que el puntero a los resultados apunte a la línea especificada.

El número de líneas comienza en 0.

- **mysql_drop_db:** Realiza una operación Drop sobre una base de datos.

```
int mysql_drop_db(string database_name, int [identificador del link])
```

Devuelve True o False según haya o no tenido éxito.

Nota: Cuidado, según la versión de MySQL, borra toda la base de datos aún conteniendo tablas con datos.

- **mysql_errno:** Devuelve un código de error relacionado con la última operación MySQL realizada.

```
int mysql_errno(int [identificador del link])
```

Esta función devuelve el número del código de error que devuelve el servidor MySQL.

- **mysql_error:** Devuelve el mensaje de error asociado a un código concreto relacionado con la última operación MySQL realizada.

```
string mysql_error(int [identificador del link])
```

Devuelve la cadena asociada a un error que posee el identificador de conexión que se especifica.

- **mysql_fetch_array:** Introduce el resultado en un array asociativo.

```
array mysql_fetch_array(int result, int [result type])
```

Esta función es una versión extendida de `mysql_fetch_row()`. Lo que hace es almacenar el resultado en un array indexado, almacenando así los datos en un índice asociativo utilizando los nombres de los campos como claves.

Si existen más de una columna con el mismo nombre, tendrá precedencia la última columna. Para acceder a las otras columnas con el mismo nombre, se debe indexar la columna o hacer un alias para ella.

El segundo argumento, es una constante que puede tomar los siguientes valores:

`MYSQL_ASSOC`

`MYSQL_NUM`

`MYSQL_BOTH`

- **mysql_fetch_field:** Devuelve un objeto con la información de un resultado.

```
object mysql_fetch_field(int result, int [field_offset])
```

Devuelve un objeto con la información que contiene result Puede utilizarse para obtener información de los campos de ciertas consultas.

Si el campo del desplazamiento no está especificado, se selecciona el siguiente dato que no haya sido ya asignado mediante esta función.

Las propiedades del objeto son:

- name: Nombre del campo.
- table: Nombre de la tabla a la que pertenece el campo.
- max_length: Longitud máxima.
- not_null: Será verdadero si el campo posee esta propiedad.
- primary_key: Verdadero si el campo es la clave principal.
- unique_key: Verdadero si el campo es clave única.
- multiple_key: Verdadero si el campo no es clave única.
- numeric: Verdadero si se trata de un campo numérico.
- blob: Verdadero si la columna es a BLOB.
- type: Tipo de la columna.
- unsigned: Verdadero si se trata de un dato sin signo.
- zerofill: Verdadero si la columna tiene la propiedad zerofill.

- **mysql_fetch_lengths:** Devuelve la longitud de los resultados.

```
array mysql_fetch_lengths(int result)
```

Devuelve un array que contiene las longitudes de todos los campos que forman el último registro asignado con la función `mysql_fetch_row()`, `mysql_fetch_array()` o `mysql_fetch_object()`.

- **mysql_fetch_object:** Devuelve el resultado como un objeto.

```
object mysql_fetch_object(int result, int [result_type])
```

Devuelve un objeto con las propiedades del último registro extraído de un resultado.

Esta función es igual que `mysql_fetch_array()` con la diferencia de que se obtiene un objeto en vez de un array.

El segundo argumento, es una constante que puede tomar los siguientes valores.

`MYSQL_ASSOC`

`MYSQL_NUM`

`MYSQL_BOTH`

- **mysql_fetch_row:** Devuelve el resultado como un array.

```
array mysql_fetch_row(int result)
```

Devuelve un array con un registro del resultado o false en el caso de que no existan registros.

Si se llama dos veces a esta función, él asignará el primer registro del resultado la primera vez y el segundo registro del resultado la segunda vez.

- **mysql_field_name:** Devuelve el nombre de un campo especificado en un resultado.

```
string mysql_field_name(int result, int field_index)
```

Devuelve el nombre del campo que se especifica, mediante el orden que ocupe dentro del resultado.

- **mysql_field_seek:** Asigna el puntero de los resultados a una posición determinada.

```
int mysql_field_seek(int result, int field_offset)
```

Busca el desplazamiento que se especifica en el segundo argumento, dentro del resultado.

- **mysql_field_table:** Obtiene el nombre de la tabla de un campo especificado.

```
string mysql_field_table(int result, int field_offset)
```

- **mysql_field_type:** Obtiene el tipo de un campo especificado.

Obtiene el nombre de la tabla para el campo que se indica en el desplazamiento.

```
string mysql_field_type(int result, int field_offset)
```

Se devuelve el tipo del campo que se indica mediante el desplazamiento.

Las cadenas que devuelve pueden ser:

- int
- real
- string
- blob

Otros tipos detallados en la documentación de MySQL.

- **mysql_field_flags:** Obtiene los "Flags" asociados del campo especificado.

```
string mysql_field_flags(int result, int field_offset)
```

Devuelve una cadena con las propiedades del campo separadas por un espacio. Para partir esa cadena se puede utilizar la función explode()

Las palabras que puede devolver son:

- not_null
- primary_key
- unique_key
- multiple_key
- blob
- unsigned
- zerofill
- binary
- enum
- auto_increment
- timestamp

- **mysql_field_len:** Obtiene la longitud de un campo especificado.

```
int mysql_field_len(int result, int field_offset)
```

Devuelve la longitud del campo especificado.

- **mysql_free_result:** Libera la memoria de los resultados.

```
int mysql_free_result(int result)
```

Libera la memoria que utiliza el parámetro.

- **mysql_insert_id:** Obtiene el identificador de la última operación INSERT.

```
int mysql_insert_id(int [link_identifier] )
```

Devuelve el ID generado para un campo AUTO_INCREMENTED.

Devolverá el id autogenerado devuelto por la última operación INSERT del enlace que se especifica como argumento.

Si el identificador no se especifica, se utiliza la última conexión abierta.

- **mysql_list_fields:** Lista los campos resultado.

```
int mysql_list_fields(string database_name, string table_name, int
[link_identifier]
```

Devuelve información del nombre de la tabla que se proporciona. El puntero que devuelve, puede ser utilizado con las funciones `mysql_field_flags()`, `mysql_field_len()`, `mysql_field_name()` y `mysql_field_type()`.

El resultado es un entero positivo, sólo devuelve -1 en caso de que exista algún error.

- **mysql_list_dbs:** Lista las bases de datos disponibles en el servidor MySQL.

```
int mysql_list_dbs(int [link_identifier] )
```

Devuelve un puntero que contiene la lista de bases de datos disponibles en ese servidor MySQL.

Con la información que devuelve, puede llamarse a la función `mysql_tablename()`.

- **mysql_list_tables:** Lista las tablas de una base de datos MySQL.

```
int mysql_list_tables(string database, int [link_identifier] )
```

Extrae un puntero a la lista de tablas que están en una base de datos.

Para extraer el nombre de las tablas se puede posteriormente utilizar `mysql_db_query()` o `mysql_tablename()`.

- **mysql_num_fields:** Obtiene el número de campos del resultado.

```
int mysql_num_fields(int result);
```

Devuelve el número de campos que contiene el parámetro resultado.

- **mysql_num_rows:** Obtiene el número de líneas del resultado.

```
int mysql_num_rows(int result)
```

Devuelve el número de líneas que contiene el parámetro resultado.

- **mysql_pconnect:** Abre una conexión permanente con una base de datos.

```
int mysql_pconnect(string [hostname [:port] [:/path/to/socket] ] ,  
string [username] , string [password] )
```

Devuelve una conexión persistente a un servidor MySQL.

Todos los argumentos son opcionales y sus valores son los mismos que para la función `mysql_connect()`.

Esta función primero, intenta encontrar una conexión ya existente, si la encuentra, devuelve un identificador para ella, y en el caso de que no la encuentre, abre una nueva conexión permanente.

Después de esto, mantiene la conexión hasta que finaliza la ejecución del script.

Hay que tener cuidado, porque `mysql_close()` no cierra conexiones abiertas con esta función.

- **mysql_select_db:** Selecciona una base de datos MySQL.

```
int mysql_select_db(string database_name, int [identificad. del link] )
```

Esta función asigna la base de datos activa en el servidor asociado al identificador del link.

- **mysql_tablename:** Obtiene el nombre de la tabla de un campo.

```
string mysql_tablename(int result, int i)
```

Esta función devuelve el nombre de una tabla de un puntero de resultados que ha devuelto previamente la función `mysql_num_rows`.

2.6.9 Cookies

Una cookie es una pequeña cantidad de datos almacenada por el navegador del usuario cuando solicita una página a un servidor. El que envía que se genere la cookie es el servidor.

Una cookie consta de un nombre, un valor, una fecha de expiración y un servidor. Una cookie está limitada a 4KB.

Luego que una cookie es creada sólo el sitio que la creó puede leerla. Luego de creada una cookie, cada vez que el navegador del usuario visita el sitio, se envía dicha cookie. Otra cosa importante que hay que tener en cuenta es que el usuario del browser puede configurar el mismo para no permitir la creación de cookies, lo que significa que el uso de cookies debe hacerse con moderación y cuando la situación lo requiera.

Para la creación de una cookie desde PHP debemos llamar a la función `setcookie`.

```
int setcookie(string name, string value, int expire, string path, string domain, int secure);
```

Todos los argumentos excepto el nombre son opcionales.

- **Nombre.** Nombre de la cookie. Si creamos una cookie solamente con el nombre, en el cliente se eliminará la cookie que exista con ese nombre. También podemos reemplazar cualquier argumento con una cadena vacía ("").
- **Value.** Valor que almacenará la cookie en el cliente.
- **Expire.** El argumento expire es un argumento entero que indica la hora en que se eliminara la cookie en el formato de hora que devuelven las funciones UNIX time() y mktime(). Normalmente se usa time() + N. segundos de duración, para especificar la duración de una cookie.
- **Path.** Subdirectorio en donde tiene valor la cookie.
- **Dominio.** Dominio en donde tiene valor la cookie. Si ponemos como dominio www.domain.com la cookie no se transmite para domain.com, mientras que si ponemos domain.com la cookie se transmite tanto para domain.com como para www.domain.com
- **Secure.** El argumento secure indica que la cookie sólo se transmitirá a través de una conexión segura HTTPS.

2.6.10 Variables de Sesión

Sirven para hacer que variables estén disponibles en múltiples páginas sin tener que pasarlas como parámetro. A diferencia de las cookies, las variables de sesión se almacenan en el servidor y tienen un tiempo limitado de existencia.

Para identificar al usuario que generó las variables de sesión, el servidor genera una clave única que es enviada al navegador y almacenada en una cookie. Luego, cada vez que el navegador solicita otra página al mismo sitio, envía esta cookie (clave única) con la cual el servidor identifica de qué navegador proviene la petición y puede rescatar de un archivo de texto las variables de sesión que se han creado. Cuando han pasado 20 minutos sin peticiones por parte de un cliente (navegador) las variables de sesión son eliminadas automáticamente (se puede configurar el entorno de PHP para variar este tiempo).

Una variable de sesión es más segura que una cookie ya que se almacena en el servidor. Otra ventaja es que no tiene que estar enviándose continuamente como sucede con las cookies. Otra ventaja de emplear una variable de sesión en lugar de una cookie es que cuando el navegador del cliente está configurado para desactivar las cookies las variables de sesión, tienen forma de funcionar (enviando la clave como parámetro en cada hipervínculo).

Como desventaja podemos decir que ocupa espacio en el servidor.

2.6.11 Clases

Las Clases son máximo exponente de la Programación Orientada a Objetos (POO). PHP no es un lenguaje orientado a objeto, pero implementa las características que permiten definir las clases.

Las clases sirven para hacer el código más legible, y lo que es más importante, reutilizable. Escribir una Clase es sin duda más largo que escribir el código directamente, pero a la larga es más rentable por su portabilidad a otras aplicaciones y su mantenimiento.

Las Clases no son más que una serie de variables y funciones que describen y actúan sobre algo.

El operador “new” crea una instancia de una clase.

La herencia en PHP se realiza con extends.

Los constructores de cada clase se llaman automáticamente por el operador “new”. El constructor de cada clase debe tomar el mismo nombre de la clase y cuando se trata de una clase derivada, hay que tener cuidado porque sólo se llama automáticamente al constructor de dicha clase pero no al constructor de la clase padre.

3 CAPÍTULO III. DESARROLLO DEL PROYECTO

3.1 Análisis del Sistema

El análisis del sistema es el conjunto de todos los elementos relacionados que van a constituir el Sistema de Extensión y Estadísticas (SEE), que representados en un plan lógico identifican las necesidades de la Dirección de Estudiantes de la Pontificia Universidad Católica del Ecuador Sede Ambato, la viabilidad para su implementación en función del hardware y software; partiendo de un análisis estructurado y técnico y, estableciendo las especificaciones y restricciones del mismo.

3.1.1 Identificación de las necesidades

La carencia de un sistema de registro y control de las actividades de Extensión Universitaria y obtención de Información Estadística de los Estudiantes en la PUCESA, da origen a las necesidades que se describen a continuación:

Un sistema automatizado de Registro y Control de Actividades de Extensión realizadas por los Estudiantes.

- Un sistema automatizado de registro de Información de Admisión de los estudiantes para su representación Gráfica Estadística que facilite su análisis.
- Una interfaz web para su accesibilidad desde internet.
- Un sistema estructurado que proporcione seguridad en el almacenamiento y recuperación de información acerca de las actividades de Extensión e Información Estadística de los Estudiantes.
- Un sistema que brinde compatibilidad de instalación y funcionamiento en el Servidor SUN que posee la Universidad y de igual manera en posteriores servidores que se utilicen.
- Uso de una base de datos relacional que permita almacenar los diferentes registros en forma segura sin ocupar un excesivo espacio en disco y que se adapte a las características del Servidor SUN e250 que posee la Universidad donde va a estar instalada.

3.1.2 Estudio de Viabilidad

El estudio de viabilidad enfoca básicamente las siguientes áreas:

- **Viabilidad Económica.** El sistema está enfocado al uso de software gratuito que además ya se encuentra implementado y en uso por otras

aplicaciones alojadas en el servidor de la Universidad; por lo que los costos del desarrollo e implementación del sistema son mínimos y accesibles en lo que se refiere al hardware y software a utilizarse en comparación al alto beneficio resultado de su implementación.

- **Viabilidad Técnica.** Existe la compatibilidad y disponibilidad de espacio en disco, necesario para el almacenamiento de datos, alojamiento del sitio Web y demás componentes requeridos por el sistema para ser instalado en el servidor SUN e250 de la Universidad; además, se cuenta con el software y hardware necesarios para la codificación del mismo, así como los conocimientos e información suficientes para su desarrollo.
- **Viabilidad Legal.** Se ha determinado que no existe ninguna de infracción, violación o responsabilidad legal en que se podría incurrir a causa del sistema.
- **Viabilidad de Alternativas.** La compatibilidad del sistema proporcionará la alternativa en caso de que el servidor de la Universidad no se encuentre a disposición por cualquier circunstancia, de que sea implementado de manera local en otro computador que cumpla con las necesidades de básicas del mismo, como son: Servidor de base de datos MySQL y compatibilidad con PHP.

Basados en el cumplimiento de los diferentes aspectos considerados en el estudio de Viabilidad se ha determinado que es factible el desarrollo e implementación del mismo.

3.1.3 Análisis Técnico

Abarca el análisis de los principios técnicos del sistema e información de necesaria para su óptimo rendimiento, fiabilidad, mantenimiento y productividad.

- **Requerimientos Servidor SUN e250.**
 1. PHP versión 4.2.2.
 2. MySQL versión 4.1.14.
 3. Apache.
 4. Módulos de conexión de PHP con MySQL.

- **Riesgos del Desarrollo.** Se puede cumplir con los requerimientos y necesidades que son motivo del desarrollo e implementación del proyecto.

- **Disponibilidad de Recursos.** Se posee el hardware y software necesarios para el desarrollo e implementación del sistema.

- **Tecnología.** La tecnología y conocimientos con los que se cuentan actualmente son los necesarios, puesto que las mismas permiten el desarrollo e implementación eficiente de la aplicación.

3.1.4 Análisis de Requisitos

El análisis de las necesidades de la Universidad que originaron el proyecto conlleva a las siguientes conclusiones:

- Es posible el desarrollo e implementación del sistema de control de Actividades de Extensión e información Estadística para la Dirección de Estudiantes de la PUCESA.
- El *front end* del sistema será desarrollado en lenguaje PHP, el mismo que es un lenguaje multiplataforma que está orientado al desarrollo de aplicaciones Web, además de soporte de programación orientada a objetos, procedimientos, y facilidad de conexión con la base de datos que en nuestro caso será MySQL.
- El *back end* del sistema será desarrollado en MySQL, al igual que PHP también estará hospedada en el servidor de la Universidad. MySQL es un potente sistema de gestión de base de datos que con la utilización del motor de almacenamiento InnoDB, brindará la transaccionalidad de datos que el sistema requiere.

3.1.4.1 Especificaciones del Sistema

Con el propósito de entender de mejor manera la estructura del Sistema de Extensión e Información Estadística, se detallan las siguientes especificaciones al respecto del mismo:

| <i>Especificaciones del Sistema de Extensión y Estadísticas</i> | |
|---|---|
| Lenguajes de programación | PHP, HTML, JavaScript |
| Base de Datos | MySQL |
| Entorno en el que va a operar | Sistemas Operativos y Browsers |
| Funciones | <ul style="list-style-type: none"> • Validación y control de usuarios. • Mantenimiento de la base de datos; el sistema debe permitir la: <ul style="list-style-type: none"> • creación, • modificación y • eliminación de los diferentes datos del sistema. • Administración de Estudiantes: <ul style="list-style-type: none"> • Actividades de Extensión, • Información correspondiente a la Admisión en los que se basarán las estadísticas. • Generación de Gráficos Estadísticos de Barras por Carrera y Fecha. • Activar y desactivar categorías en las que se basan las representaciones estadísticas. • Mantenimiento de las notas máximas de los bloques de preguntas para el Examen de Admisión. • Reportes sobre las actividades de Extensión Realizadas y registros de Admisión Ingresados. |

| | |
|----------------------|---|
| Restricciones | <ul style="list-style-type: none">• El sistema será utilizado exclusivamente para registro y control de Actividades de Extensión Universitaria y generación de gráficos estadísticos de barras basados en la información obtenida en la admisión de los estudiantes.• No será un sistema académico, lo cual implica que no deberá almacenar ninguna información innecesaria que no guarde relación con las actividades de Extensión Universitaria e información obtenida en la Admisión de los Estudiantes.• El sistema no realizara directamente funciones de respaldos de procesamiento de datos ni operaciones de backup y recuperación automatizados. Todos estos procesos quedan a cargo de la administración del sitio Web donde estará hospedado el sistema, previa capacitación para su utilización.• El sistema necesita aproximadamente de 5 Mb de espacio en disco para el alojamiento de las páginas Web y, 10 Mb para el almacenamiento de la base de datos.• El sistema hace uso para el registro de información de estudiantes, del Maestro de Estudiantes del Sistema de Evaluación Docente razón por la cual es requisito su instalación en conjunto con la Base de Datos "evaluacion" para hacer uso de éste proceso. |
|----------------------|---|

3.1.5 Análisis Estructurado

3.1.5.1 Diagrama de Flujo de Datos Nivel 0

Representa un modelo fundamental del sistema. Se simbolizan entidades externas e internas de forma general, salidas, entradas y procesos básicos.

Diagrama de Flujo de Datos Nivel 0

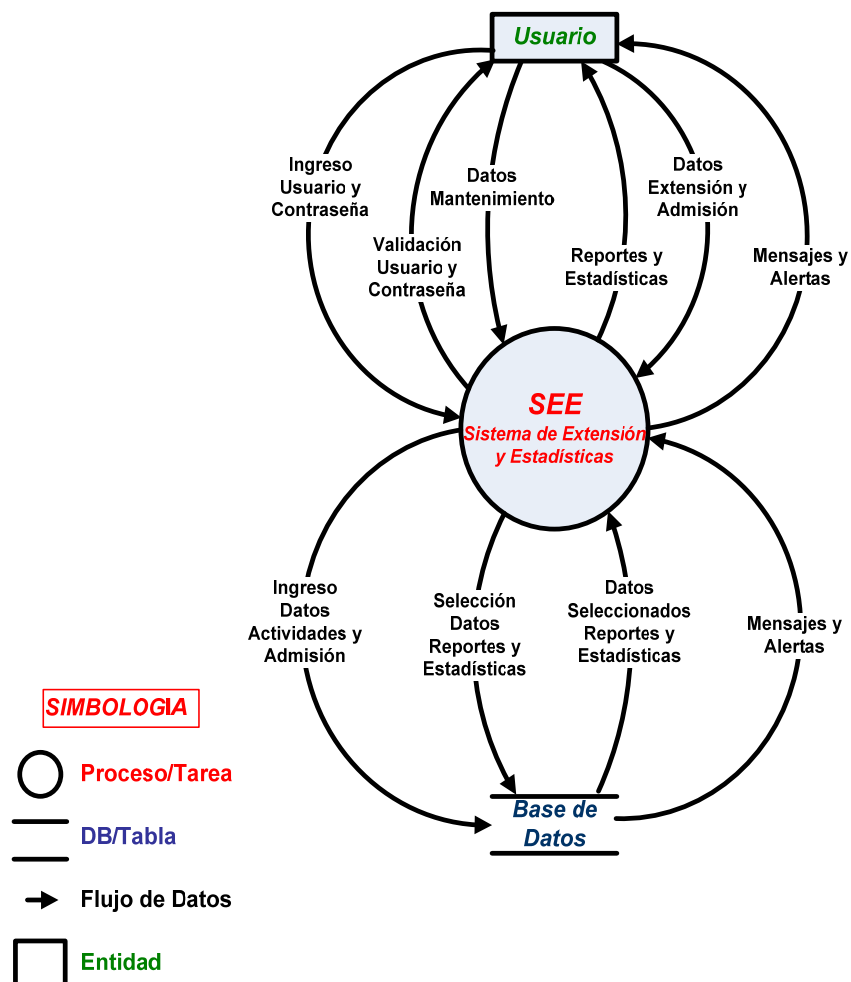


Figura 3.1. Diagrama de Flujo de Datos Nivel 0

3.1.5.2 Diagrama de Flujo de Datos Nivel 1

Representa un modelo del sistema con mayor detalle, las rutinas y procesos del sistema, así como la interacción con el usuario.

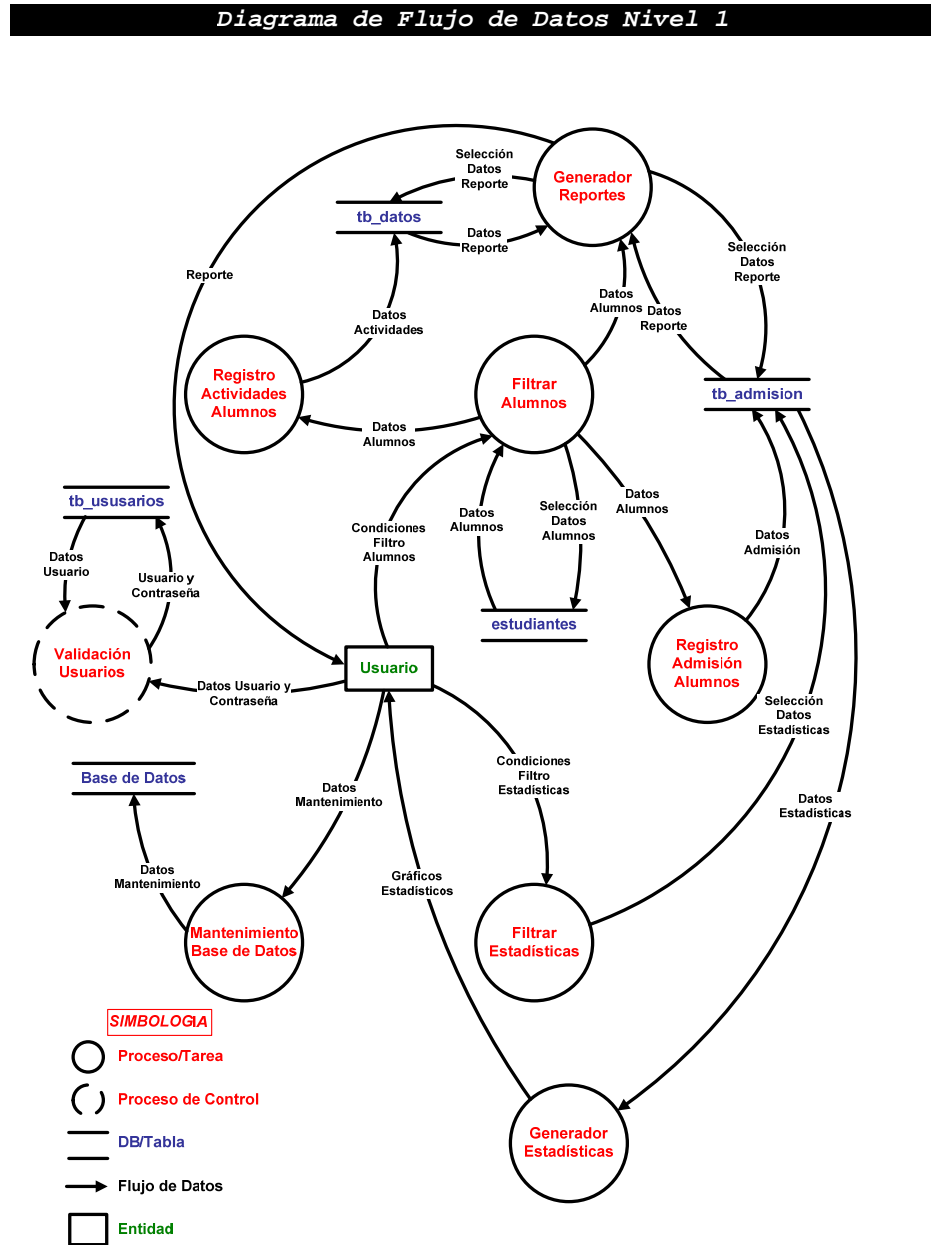


Figura 3.2. Diagrama de Flujo de Datos Nivel 1

3.2 Diseño del Sistema

Para el diseño del sistema, se toman en cuenta los requisitos explícitos contenidos en el análisis; es decir, necesidades, viabilidad, análisis técnico, especificaciones y análisis estructurado, y se los traduce en una representación técnica de software; una representación lo más cercana al lenguaje de máquina del problema real.

Se desarrolla un diseño detallado que represente el sistema de manera que se pueda leer y entender, se pueda traducir en código, probar y mantener.

Para la elaboración del diseño se deben tomar en cuenta los siguientes criterios técnicos:

- El sistema exige una organización jerárquica que haga un uso inteligente del control entre los componentes del software.
- El diseño debe ser modular y tener una división lógica del Software en elementos que realicen funciones y subfunciones específicas.
- El diseño debe contener abstracciones de datos y procedimientos.
- El diseño debe llevar a módulos que presenten características de funcionamiento independiente.
- El diseño debe conducir a interfaces que reduzcan la complejidad de las conexiones entre los módulos y el Front end.

- El diseño debe representar un método que pueda ser repetido en base a la información obtenida del análisis de requisitos.

El diseño del sistema representa en forma detallada la base de datos, el sitio Web e interfaces, y los métodos y procedimientos usados para cumplir con los requerimientos del Sistema de Extensión y Estadísticas.

3.2.1 Diseño de la Base de Datos

El diseño de la base de datos del sistema SEE describe de forma detallada los objetos que la conforman como son entidades con sus atributos, relaciones, índices, tipo de datos y longitudes de cada columna.

Información del Modelo Físico. Detalla información básica y estadística del modelo físico de la base de datos. Aquí se localiza el nombre de la base de datos, número de entidades, atributos, índices y relaciones.

Diagrama Entidad - Relación. Define en forma gráfica a las entidades, relaciones y atributos que conforman la base de datos del sistema SEE.

Tablas. Detalla a todas las entidades con sus respectivas columnas, claves primarias y secundarias, índices y relaciones.

Columnas. Describe cada atributo de las entidades de la base de datos, tipo de dato, si es clave primaria o secundaria, si es un dato nulo o único.

Relaciones. Describe las relaciones existentes en la base de datos, el tipo de relación, su tabla origen y destino.

Índices. Detalla a los índices y los campos que las conforman, que identifican las claves foráneas en una determinada entidad.

Información del Modelo Físico

| Nombre Base de Datos | Extension |
|----------------------|-----------|
| Tablas | 14 |
| Columnas | 114 |
| Índices | 30 |
| Relaciones | 6 |

Diagrama Entidad - Relación

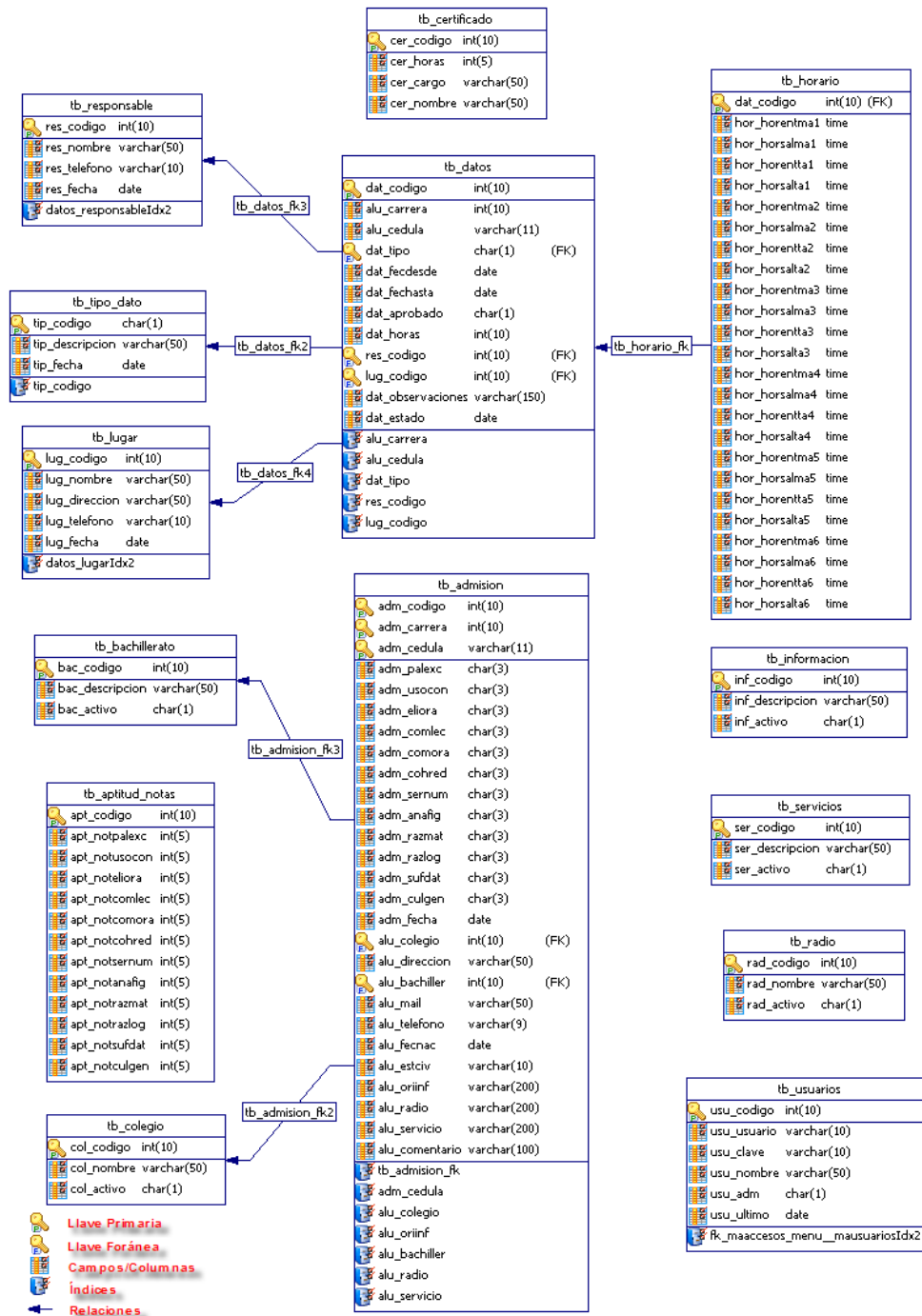


Figura 3.3. Diagrama Entidad - Relación

Tablas - Columnas - Relaciones - Índices

Tabla: tb_admision

| Nombre Columna | Tipo Dato | Llave | No | Auto | Valor Por |
|----------------------|-------------------------|----------|------|------------|------------|
| | | Primaria | Nulo | Incremento | Defecto |
| adm_codigo | int(10) | SI | SI | NO | 0 |
| adm_carrera | int(10) | SI | SI | NO | 0 |
| adm_cedula | varchar(11) | SI | SI | NO | 0 |
| adm_palex | Char(3) | NO | SI | NO | 0 |
| adm_usocon | Char(3) | NO | SI | NO | 0 |
| adm_eliora | Char(3) | NO | SI | NO | 0 |
| adm_comlec | Char(3) | NO | SI | NO | 0 |
| adm_comora | Char(3) | NO | SI | NO | 0 |
| adm_cohred | Char(3) | NO | SI | NO | 0 |
| adm_serenum | Char(3) | NO | SI | NO | 0 |
| adm_anafig | Char(3) | NO | SI | NO | 0 |
| adm_razmat | Char(3) | NO | SI | NO | 0 |
| adm_razlog | Char(3) | NO | SI | NO | 0 |
| adm_sufdat | char(3) | NO | SI | NO | |
| adm_culgen | char(3) | NO | SI | NO | |
| adm_fecha | Date | NO | SI | NO | |
| alu_colegio | int(10) | NO | SI | NO | 0 |
| alu_direccion | varchar(50) | NO | SI | NO | |
| alu_bachiller | int(10) | NO | SI | NO | 0 |
| alu_mail | varchar(50) | NO | NO | NO | |
| alu_telefono | varchar(9) | NO | NO | NO | |
| alu_fecnac | Date | NO | SI | NO | 0000-00-00 |
| alu_estciv | varchar(10) | NO | NO | NO | |
| alu_oriinf | varchar(200) | NO | NO | NO | 0 |
| alu_radio | varchar(200) | NO | NO | NO | 0 |
| alu_servicio | varchar(200) | NO | NO | NO | 0 |
| alu_comentario | varchar(100) | NO | NO | NO | |
| Nombre Indice | Columnas | | | | |
| PRIMARY | adm_codigo,adm_carrera, | | | | |
| | adm_cedula | | | | |
| tb_admision_fk | adm_carrera | | | | |
| adm_cedula | adm_cedula | | | | |
| alu_colegio | alu_colegio | | | | |
| alu_oriinf | alu_oriinf | | | | |
| alu_bachiller | alu_bachiller | | | | |
| alu_radio | alu_radio | | | | |
| alu_servicio | alu_servicio | | | | |

tb_admision_fk2

| Tabla Origen | Tabla Destino | Borrado | Actualización | Link |
|--------------|---------------|----------|---------------|------------------------|
| tb_colegio | Tb_admision | RESTRICT | RESTRICT | col_codigo=alu_colegio |

tb_admision_fk3

| Tabla Origen | Tabla Destino | Borrado | Actualización | Link |
|-----------------|---------------|----------|---------------|--------------------------|
| tb_bachillerato | tb_admision | RESTRICT | RESTRICT | bac_codigo=alu_bachiller |

Tabla: tb_aptitud_notas

| Nombre Columna | Tipo Dato | Llave | No | Auto | Valor Por |
|----------------------|-----------------|----------|------|------------|-----------|
| | | Primaria | Nulo | Incremento | Defecto |
| apt_codigo | int(10) | SI | SI | NO | 0 |
| apt_notpalexc | int(5) | NO | SI | NO | 0 |
| apt_notusocon | int(5) | NO | SI | NO | 0 |
| apt_noteliora | int(5) | NO | SI | NO | 0 |
| apt_notcomlec | int(5) | NO | SI | NO | 0 |
| apt_notcomora | int(5) | NO | SI | NO | 0 |
| apt_notcohred | int(5) | NO | SI | NO | 0 |
| apt_notsernum | int(5) | NO | SI | NO | 0 |
| apt_notanafig | int(5) | NO | SI | NO | 0 |
| apt_notrazmat | int(5) | NO | SI | NO | 0 |
| apt_notrazlog | int(5) | NO | SI | NO | 0 |
| apt_otsufdat | int(5) | NO | SI | NO | 0 |
| apt_notculgen | int(5) | NO | SI | NO | 0 |
| Nombre Indice | Columnas | | | | |
| PRIMARY | apt_codigo | | | | |

Tabla: tb_bachillerato

| Nombre Columna | Tipo Dato | Llave | No Nulo | Auto | Valor Por Defecto |
|----------------------|-----------------|----------|---------|------------|-------------------|
| | | Primaria | | Incremento | |
| bac_codigo | int(10) | SI | SI | NO | 0 |
| bac_descripcion | varchar(50) | NO | SI | NO | |
| bac_activo | char(1) | NO | SI | NO | |
| Nombre Indice | Columnas | | | | |
| PRIMARY | bac_codigo | | | | |

Tabla: tb_certificado

| Nombre Columna | Tipo Dato | Llave | No | Auto | Valor Por Defecto |
|----------------------|-----------------|----------|------|------------|-------------------|
| | | Primaria | Nulo | Incremento | |
| cer_codigo | int(10) | SI | SI | NO | 0 |
| cer_horas | int(5) | NO | SI | NO | 0 |
| cer_cargo | varchar(50) | NO | SI | NO | |
| cer_nombre | varchar(50) | NO | SI | NO | |
| Nombre Indice | Columnas | | | | |
| PRIMARY | cer_codigo | | | | |

Tabla: tb_colegio

| Nombre Columna | Tipo Dato | Llave | No | Auto | Valor Por Defecto |
|----------------------|-----------------|----------|------|------------|-------------------|
| | | Primaria | Nulo | Incremento | |
| col_codigo | int(10) | SI | SI | NO | 0 |
| col_nombre | varchar(50) | NO | SI | NO | |
| col_activo | char(1) | NO | SI | NO | |
| Nombre Indice | Columnas | | | | |
| PRIMARY | col_codigo | | | | |

Tabla: tb_datos

| Nombre Columna | Tipo Dato | Llave | No Nulo | Auto | Valor Por |
|----------------------|-----------------|----------|---------|------------|------------|
| | | Primaria | | Incremento | Defecto |
| dat_codigo | int(10) | SI | SI | NO | 0 |
| alu_carrera | int(10) | NO | SI | NO | 0 |
| alu_cedula | varchar(11) | NO | SI | NO | 0 |
| dat_tipo | char(1) | NO | SI | NO | |
| dat_fecdesde | Date | NO | SI | NO | 0000-00-00 |
| dat_fechasta | Date | NO | SI | NO | 0000-00-00 |
| dat_aprobado | char(1) | NO | SI | NO | |
| dat_horas | int(10) | NO | SI | NO | 0 |
| res_codigo | int(10) | NO | SI | NO | 0 |
| lug_codigo | int(10) | NO | SI | NO | 0 |
| dat_observaciones | varchar(150) | NO | NO | NO | |
| dat_estado | Date | NO | NO | NO | |
| Nombre Indice | Columnas | | | | |
| PRIMARY | dat_codigo | | | | |
| | alu_carrera | | | | |
| | alu_cedula | | | | |
| | dat_tipo | | | | |
| | res_codigo | | | | |
| | lug_codigo | | | | |

tb_datos_fk2

| Tabla Origen | Tabla Destino | Borrado | Actualización | Link |
|--------------|---------------|----------|---------------|---------------------|
| tb_tipo_dato | Tb_datos | RESTRICT | RESTRICT | tip_codigo=dat_tipo |

tb_datos_fk3

| Tabla Origen | Tabla Destino | Borrado | Actualización | Link |
|----------------|---------------|----------|---------------|-----------------------|
| tb_responsable | tb_datos | RESTRICT | RESTRICT | res_codigo=res_codigo |

tb_datos_fk4

| Tabla Origen | Tabla Destino | Borrado | Actualización | Link |
|--------------|---------------|----------|---------------|-----------------------|
| tb_lugar | Tb_datos | RESTRICT | RESTRICT | lug_codigo=lug_codigo |

Tabla: tb_horario

| Nombre Columna | Tipo Dato | Llave | No Nulo | Auto | Valor Por Defecto |
|----------------------|-----------------|----------|---------|------------|-------------------|
| | | Primaria | | Incremento | |
| dat_codigo | int(10) | SI | SI | NO | 0 |
| hor_horentma1 | time | NO | NO | NO | |
| hor_horsalma1 | time | NO | NO | NO | |
| hor_horentta1 | time | NO | NO | NO | |
| hor_horsalta1 | time | NO | NO | NO | |
| hor_horentma2 | time | NO | NO | NO | |
| hor_horsalma2 | time | NO | NO | NO | |
| hor_horentta2 | time | NO | NO | NO | |
| hor_horsalta2 | time | NO | NO | NO | |
| hor_horentma3 | time | NO | NO | NO | |
| hor_horsalma3 | time | NO | NO | NO | |
| hor_horentta3 | time | NO | NO | NO | |
| hor_horsalta3 | time | NO | NO | NO | |
| hor_horentma4 | time | NO | NO | NO | |
| hor_horsalma4 | time | NO | NO | NO | |
| hor_horentta4 | time | NO | NO | NO | |
| hor_horsalta4 | time | NO | NO | NO | |
| hor_horentma5 | time | NO | NO | NO | |
| hor_horsalma5 | time | NO | NO | NO | |
| hor_horentta5 | time | NO | NO | NO | |
| hor_horsalta5 | time | NO | NO | NO | |
| hor_horentma6 | time | NO | NO | NO | |
| hor_horsalma6 | time | NO | NO | NO | |
| hor_horentta6 | time | NO | NO | NO | |
| hor_horsalta6 | time | NO | NO | NO | |
| Nombre Indice | Columnas | | | | |
| PRIMARY | dat_codigo | | | | |

tb_horario_fk

| Tabla Origen | Tabla Destino | Borrado | Actualización | Link |
|--------------|---------------|----------|---------------|-----------------------|
| tb_datos | Tb_horario | RESTRICT | RESTRICT | dat_codigo=dat_codigo |

Tabla: tb_informacion

| Nombre Columna | Tipo Dato | Llave | No Nulo | Auto | Valor Por Defecto |
|----------------------|-----------------|----------|---------|------------|-------------------|
| | | Primaria | Nulo | Incremento | |
| inf_codigo | int(10) | SI | SI | NO | 0 |
| inf_descripcion | varchar(50) | NO | SI | NO | |
| inf_activo | char(1) | NO | SI | NO | |
| Nombre Indice | Columnas | | | | |
| PRIMARY | inf_codigo | | | | |

Tabla: tb_lugar

| Nombre | Tipo Dato | Llave | No | Auto | Valor Por |
|----------------------|-----------------|----------|------|------------|-----------|
| Columna | | Primaria | Nulo | Incremento | Defecto |
| lug_codigo | int(10) | SI | SI | NO | 0 |
| lug_nombre | varchar(50) | NO | SI | NO | |
| lug_direccion | varchar(50) | NO | SI | NO | |
| lug_telefono | varchar(10) | NO | NO | NO | |
| lug_fecha | Date | NO | NO | NO | |
| Nombre Indice | Columnas | | | | |
| PRIMARY | lug_codigo | | | | |
| datos_lugarIdx2 | lug_codigo | | | | |

Tabla: tb_radio

| Nombre Columna | Tipo Dato | Llave | No | Auto | Valor Por |
|----------------------|-----------------|----------|------|------------|-----------|
| | | Primaria | Nulo | Incremento | Defecto |
| rad_codigo | int(10) | SI | SI | NO | 0 |
| rad_nombre | varchar(50) | NO | SI | NO | |
| rad_activo | char(1) | NO | SI | NO | |
| Nombre Indice | Columnas | | | | |
| PRIMARY | rad_codigo | | | | |

Tabla: tb_responsable

| Nombre Columna | Tipo Dato | Llave | No Nulo | Auto | Valor Por |
|-----------------------|-----------------|----------|---------|------------|-----------|
| | | Primaria | | Incremento | Defecto |
| res_codigo | int(10) | SI | SI | NO | 0 |
| res_nombre | Varchar(50) | NO | SI | NO | |
| res_telefono | varchar(10) | NO | NO | NO | |
| res_fecha | Date | NO | NO | NO | |
| Nombre Indice | Columnas | | | | |
| PRIMARY | res_codigo | | | | |
| datos_responsableIdx2 | res_codigo | | | | |

Tabla: tb_servicios

| Nombre | Tipo Dato | Llave | No Nulo | Auto | Valor Por |
|----------------------|-----------------|----------|---------|------------|-----------|
| Columna | | Primaria | | Incremento | Defecto |
| ser_codigo | int(10) | SI | SI | NO | 0 |
| ser_descripcion | varchar(50) | NO | SI | NO | |
| ser_activo | char(1) | NO | SI | NO | |
| Nombre Indice | Columnas | | | | |
| PRIMARY | ser_codigo | | | | |

Tabla: tb_tipo_dato

| Nombre Columna | Tipo Dato | Llave Primaria | No Nulo | Auto Incremento | Valor Por Defecto |
|----------------------|-----------------|----------------|---------|-----------------|-------------------|
| tip_codigo | char(1) | SI | SI | NO | |
| tip_descripcion | varchar(50) | NO | SI | NO | |
| tip_fecha | Date | NO | NO | NO | |
| Nombre Indice | Columnas | | | | |
| PRIMARY | Tip_codigo | | | | |
| tip_codigo | Tip_codigo | | | | |

Tabla: tb_usuarios

| Nombre Columna | Tipo Dato | Llave Primaria | No Nulo | Auto Incremento | Valor Por Defecto |
|----------------------|-----------------|----------------|---------|-----------------|-------------------|
| usu_codigo | Int(10) | SI | SI | NO | 0 |
| usu_usuario | varchar(10) | NO | SI | NO | |
| usu_clave | varchar(10) | NO | SI | NO | |
| usu_nombre | varchar(50) | NO | SI | NO | |
| usu_adm | char(1) | NO | SI | NO | |
| usu_ultimo | Date | NO | NO | NO | |
| Nombre Indice | Columnas | | | | |
| PRIMARY | usu_codigo | | | | |
| fk_maaccesos_menu_ | usu_usuario | | | | |
| mausuariosIdx2 | o | | | | |

3.2.2 Diseño del Sitio Web y la Interfaz

El diseño del sitio Web del SEE define la interfaz y estructura del sistema de forma jerárquica.

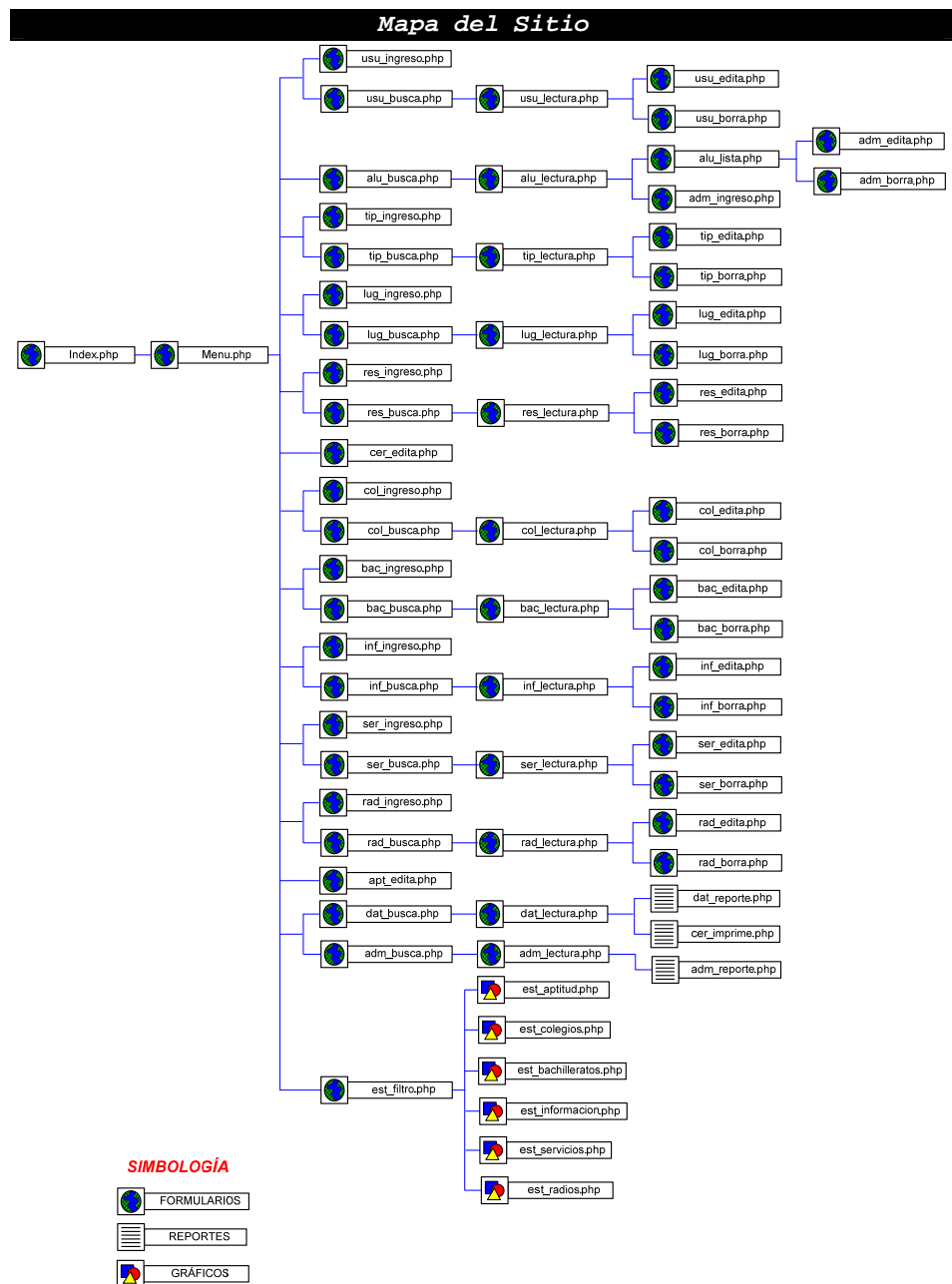


Figura 3.4. Mapa del Sitio

A continuación se detallan las carpetas y archivos del sitio con su extensión y una descripción.

Archivos Sitio Web

| Nombre del Archivo /Carpeta/ Página | Descripción |
|--|---|
| imagenes | Carpeta de imágenes sitio |
| img_est | Carpeta de imágenes Gráficos Estadísticos |
| img_menu | Carpeta de imágenes menú |
| js | Carpeta Control Calendario |
| ScriptX.cab | Archivo Control para Impresión |
| index.php | Página de inicio del sistema |
| mm_menu.js | Página javascript con funciones y procedimientos del menú |
| mm_css_menu.js | Página de estilos del menú |
| menu_adm.htm | Página Menú Administrador |
| menu_no_adm.htm | Página Menú No Administrador |
| estilos.ccs | Página de estilos del sitio |
| conexion.php | Página de conexión con la base de datos |
| validation.php | Página Script de Validación de Formularios |
| as-diagrams.php | Página Script para Gráficos Estadísticos |
| adm_borra.php | Página Eliminar Resultados Examen de Admisión/Encuestas |
| adm_busca.php | Página Filtro Búsqueda Registros de Admisión |
| adm_edita.php | Página Modificar Registros Admisión |
| adm_ingreso.php | Página Ingreso Registros Admisión |
| adm_lectura.php | Página Reporte por Pantalla Registros Admisión |
| adm_reporte.php | Página Reporte por Impresora Registros Admisión |
| alu_busca | Página Filtro Búsqueda Alumnos |
| alu_lectura.php | Página Listado de Alumnos |
| alu_lista.php | Página Detalles Registros de Alumnos |
| apt_edita.php | Página Modificar Notas Máximas Examen de Aptitud |
| bac_borra.php | Página Borrar Bachillerato |
| bac_busca.php | Página Filtro Búsqueda Bachillerato |
| bac_edita.php | Página Modificar Bachillerato |
| bac_ingreso.php | Página Ingreso Bachillerato |
| bac_lectura.php | Página Listar Bachillerato |
| cer_edita.php | Página Modificar Datos Certificado Actividades |
| cer_imprime.php | Página Impresión Certificado de Actividades |
| col_borra.php | Página Borrar Colegio |
| col_busca.php | Página Filtro Búsqueda Colegio |
| col_edita.php | Página Modificar Colegio |
| col_ingreso.php | Página Ingresar Colegio |
| col_lectura.php | Página Listar Colegios |
| dat_borra.php | Página Borrar Actividades |
| dat_busca.php | Página Filtro Búsqueda de Actividades |
| dat_edita.php | Página Modificar Actividades |
| dat_ingreso.php | Página Ingresar Actividades |

| | |
|------------------------------------|---|
| <code>dat_lectura.php</code> | Página Reporte por Pantalla Registros Actividades |
| <code>dat_lista.php</code> | Página Listar Actividades |
| <code>dat_reporte.php</code> | Página Reporte por Impresora Registros Actividades |
| <code>est_apitud.php</code> | Página Gráfico Estadístico Aciertos Examen de Aptitud |
| <code>est_bachilleratos.php</code> | Página Gráfico Estadístico por Bachilleratos |
| <code>est_colegios.php</code> | Página Gráfico Estadístico por Colegios |
| <code>est_filtro.php</code> | Página Filtro Estadísticas |
| <code>est_informacion.php</code> | Página Gráfico Estadístico por Origen Obtención Información |
| <code>est_radios.php</code> | Página Gráfico Estadístico por Radioemisoras |
| <code>est_servicios.php</code> | Página Gráfico Estadístico por Servicios Ofrecidos |
| <code>index.php</code> | Página Ingreso Usuario y Password |
| <code>inf_borra.php</code> | Página Borrar Origen de Información |
| <code>inf_busca.php</code> | Página Filtro Búsqueda Origen de Información |
| <code>inf_edita.php</code> | Página Modificar Origen de Información |
| <code>inf_ingreso.php</code> | Página Ingresar Origen de Información |
| <code>inf_lectura.php</code> | Página Listar Origen de Información |
| <code>lug_borra.php</code> | Página Borrar Lugar |
| <code>lug_busca.php</code> | Página Filtro Búsqueda Lugar |
| <code>lug_edita.php</code> | Página Modificar Lugar |
| <code>lug_ingreso.php</code> | Página Ingresar Lugar |
| <code>lug_lectura.php</code> | Página Listar Lugares |
| <code>menu.php</code> | Página Inicial para Menú |
| <code>rad_borra.php</code> | Página Borrar Radioemisora |
| <code>rad_busca.php</code> | Página Filtro Búsqueda Radioemisora |
| <code>rad_edita.php</code> | Página Modificar Radioemisora |
| <code>rad_ingreso.php</code> | Página Ingresar Radioemisora |
| <code>rad_lectura.php</code> | Página Listar Radioemisoras |
| <code>res_borra.php</code> | Página Borrar Responsable |
| <code>res_busca.php</code> | Página Filtro Búsqueda Responsable |
| <code>res_edita.php</code> | Página Modificar Responsable |
| <code>res_ingreso.php</code> | Página Ingreso Responsable |
| <code>res_lectura.php</code> | Página Listar Responsables |
| <code>salir.php</code> | Página Salir |
| <code>ser_borra.php</code> | Página Borrar Servicios Ofrecidos |
| <code>ser_busca.php</code> | Página Filtro Búsqueda Servicios Ofrecidos |
| <code>ser_edita.php</code> | Página Modificar Servicios Ofrecidos |
| <code>ser_ingreso.php</code> | Página Ingresar Servicios Ofrecidos |
| <code>ser_lectura.php</code> | Página Listar Servicios Ofrecidos |
| <code>tip_borra.php</code> | Página Borrar Tipo de Dato Actividad |
| <code>tip_busca.php</code> | Página Filtro Búsqueda Tipo de Dato Actividad |
| <code>tip_edita.php</code> | Página Modificar Tipo de Dato Actividad |
| <code>tip_ingreso.php</code> | Página Ingresar Tipo de Dato Actividad |
| <code>tip_lectura.php</code> | Página Listar Tipo de Dato Actividad |
| <code>usu_borra.php</code> | Página Borrar Usuario del Sistema |
| <code>usu_busca.php</code> | Página Filtro Búsqueda Usuarios del Sistema |
| <code>usu_edita.php</code> | Página Modificar Usuario del Sistema |
| <code>usu_ingreso.php</code> | Página Ingresar Usuario del Sistema |
| <code>usu_lectura.php</code> | Página Listar Usuarios del Sistema |

Menú de Usuarios

| Menú | Usuario Administ. | Usuario No Administ. | Página |
|---|----------------------|-------------------------|-----------------|
| 1. Usuarios | Si | No | - |
| 1.1. Nuevo Usuario | Si | No | usu_ingreso.php |
| 1.2. Consulta/Modificación Usuarios | Si | No | usu_busca.php |
| 2. Mantenimiento | Si | Si | - |
| 2.1. Alumno | Si | Si | alu_busca.php |
| 2.2. Actividades | Si | Si | - |
| 2.2.1. Tipo de Dato | Si | Si | - |
| 2.2.1.1. Nuevo Tipo de Dato | Si | Si | tip_ingreso.php |
| 2.2.1.2. Consulta/ Modificación Tipos de Datos | Si | Si | tip_busca.php |
| 2.2.2. Lugar | Si | Si | - |
| 2.2.2.1. Nuevo Lugar | Si | Si | lug_ingreso.php |
| 2.2.2.2. Consulta/ Modificación Lugares | Si | Si | lug_busca.php |
| 2.2.3. Responsable | Si | Si | - |
| 2.2.3.1. Nuevo Responsable | Si | Si | res_ingreso.php |
| 2.2.3.2. Consulta/ Modificación Responsables | Si | Si | res_busca.php |
| 2.2.4. Datos Certificado | Si | Si | cer_edita.php |
| 2.3. Estadísticas | Si | Si | - |
| 2.3.1. Colegios | Si | Si | - |
| 2.3.1.1. Nuevo Colegio | Si | Si | col_ingreso.php |
| 2.3.1.2. Consulta/ Modificación Colegios | Si | Si | col_busca.php |
| 2.3.2. Bachillerato | Si | Si | - |
| 2.3.2.1. Nuevo Bachillerato | Si | Si | bac_ingreso.php |
| 2.3.2.2. Consulta/ Modificación Bachilleratos | Si | Si | bac_busca.php |
| 2.3.3. Origen Información | Si | Si | - |
| 2.3.3.1. Nuevo Origen Información | Si | Si | inf_ingreso.php |

| | | | |
|---|----|----|-----------------|
| 2.3.3.2. Consulta/ Modificación Origen Información | Si | Si | inf_busca.php |
| 2.3.4. Servicios | Si | Si | - |
| 2.3.4.1. Nuevo Servicio | Si | Si | ser_ingreso.php |
| 2.3.4.2. Consulta/ Modificación Servicios | Si | Si | ser_busca.php |
| 2.3.5. Radioemisora | Si | Si | - |
| 2.3.5.1. Nueva Radioemisora | Si | Si | rad_ingreso.php |
| 2.3.5.2. Consulta/ Modificación Radioemisora | Si | Si | rad_busca.php |
| 2.3.6. Notas Bloques Preguntas Examen Admisión | Si | Si | apt_edita.php |
| 3. Reportes | Si | Si | - |
| 3.1. Actividades | Si | Si | dat_busca.php |
| 3.2. Datos Personales/Admisión/ Encuestas | Si | Si | adm_busca.php |
| 4. Estadísticas | Si | Si | - |
| 4.1. Aptitud Académica | Si | Si | est_filtro.php |
| 4.2. Ingreso por Colegios | Si | Si | est_filtro.php |
| 4.3. Ingreso por Bachilleratos | Si | Si | est_filtro.php |
| 4.4. Origen de Información | Si | Si | est_filtro.php |
| 4.5. Ingreso por Servicios | Si | Si | est_filtro.php |
| 4.6. Ingreso por Servicios | Si | Si | est_filtro.php |
| 5. Salir | Si | Si | - |

3.3 Desarrollo del sistema

El desarrollo del sistema SEE abarca la creación, codificación e interacción de los diferentes módulos entre sí, partiendo del diseño y análisis del sistema, desarrollando la codificación del *front end* e implementando las diferentes entidades con sus atributos para el *back end*, con el propósito de alcanzar un óptimo funcionamiento para que el proceso de registro de información y manipulación de la misma se desarrolle de la mejor manera.

3.3.1 Creación de la Base de Datos

A continuación se describe mediante una estructura SQL la creación de la base de datos, las entidades con sus atributos de la estructura del sistema SEE en el motor de base de datos MySQL:

Creación de la Base de Datos

```
-- Creación de la Base de Datos "extensión"

CREATE DATABASE extension;

-- Uso de la Base de Datos "extensión"

USE extension;
```

```
-- Colocar parámetro de inicialización
```

```
SET FOREIGN_KEY_CHECKS=0;
```

Tabla: tb_admision

```
-- Estructura para la tabla `tb_admision`
```

```
CREATE TABLE `tb_admision` (
  `adm_codigo` int(10) NOT NULL DEFAULT '0',
  `adm_carrera` int(10) NOT NULL DEFAULT '0',
  `adm_cedula` varchar(11) NOT NULL DEFAULT '0',
  `adm_palex` char(3) NOT NULL DEFAULT '0',
  `adm_usocon` char(3) NOT NULL DEFAULT '0',
  `adm_eliora` char(3) NOT NULL DEFAULT '0',
  `adm_comlec` char(3) NOT NULL DEFAULT '0',
  `adm_comora` char(3) NOT NULL DEFAULT '0',
  `adm_cohred` char(3) NOT NULL DEFAULT '0',
  `adm_sernum` char(3) NOT NULL DEFAULT '0',
  `adm_anafig` char(3) NOT NULL DEFAULT '0',
  `adm_razmat` char(3) NOT NULL DEFAULT '0',
  `adm_razlog` char(3) NOT NULL DEFAULT '0',
  `adm_sufdat` char(3) NOT NULL,
  `adm_culgen` char(3) NOT NULL,
  `adm_fecha` date,
  `alu_colegio` int(10) NOT NULL DEFAULT '0',
  `alu_direccion` varchar(50) NOT NULL,
  `alu_bachiller` int(10) NOT NULL DEFAULT '0',
  `alu_mail` varchar(50),
  `alu_telefono` varchar(9),
  `alu_fecnac` date NOT NULL DEFAULT '0000-00-00',
  `alu_estciv` varchar(10) NOT NULL,
  `alu_oriinf` varchar(200) NOT NULL DEFAULT '0',
  `alu_radio` varchar(200) NOT NULL DEFAULT '0',
  `alu_servicio` varchar(200) NOT NULL DEFAULT '0',
```

```

`alu_comentario` varchar(100),
PRIMARY KEY(`adm_codigo`, `adm_carrera`, `adm_cedula`),
INDEX `tb_admision_fk`(`adm_carrera`),
INDEX `adm_cedula`(`adm_cedula`),
INDEX `alu_colegio`(`alu_colegio`),
INDEX `alu_oriinf`(`alu_oriinf`),
INDEX `alu_bachiller`(`alu_bachiller`),
INDEX `alu_radio`(`alu_radio`),
INDEX `alu_servicio`(`alu_servicio`),
CONSTRAINT `tb_admision_fk2` FOREIGN KEY (`alu_colegio`)
REFERENCES `tb_colegio`(`col_codigo`)
ON DELETE RESTRICT
ON UPDATE RESTRICT,
CONSTRAINT `tb_admision_fk3` FOREIGN KEY (`alu_bachiller`)
REFERENCES `tb_bachillerato`(`bac_codigo`)
ON DELETE RESTRICT
ON UPDATE RESTRICT
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 11264 kB; (`adm_carrera`) REFER
`evaluacion/carrera_estudiante`(`COD_CARRERA`); (`adm_cedula`)
REFER `evaluacion/estudiantes`(`CEDULA`); (`alu_colegio`)
REFER `extension/tb_colegio`(`col_codigo`); (`alu_bachiller`)
REFER `extension/tb_bachillerato`(`bac_codigo`);';

```

Tabla: tb_aptitud_notas

```

-- Estructura para la tabla `tb_aptitud_notas`

CREATE TABLE `tb_aptitud_notas` (
  `apt_codigo` int(10) NOT NULL DEFAULT '0',
  `apt_notpalexc` int(5) NOT NULL DEFAULT '0',
  `apt_notusocon` int(5) NOT NULL DEFAULT '0',
  `apt_noteliora` int(5) NOT NULL DEFAULT '0',

```

```

`apt_notcomlec` int(5) NOT NULL DEFAULT '0',
`apt_notcomora` int(5) NOT NULL DEFAULT '0',
`apt_notcohred` int(5) NOT NULL DEFAULT '0',
`apt_notsernum` int(5) NOT NULL DEFAULT '0',
`apt_notanafig` int(5) NOT NULL DEFAULT '0',
`apt_notrazmat` int(5) NOT NULL DEFAULT '0',
`apt_notrazlog` int(5) NOT NULL DEFAULT '0',
`apt_notzufdat` int(5) NOT NULL DEFAULT '0',
`apt_notculgen` int(5) NOT NULL DEFAULT '0',
PRIMARY KEY(`apt_codigo`)
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 11264 kB';

```

Tabla: tb_bachillerato

```

-- Estructura para la tabla `tb_bachillerato`

CREATE TABLE `tb_bachillerato` (
  `bac_codigo` int(10) NOT NULL DEFAULT '0',
  `bac_descripcion` varchar(50) NOT NULL,
  `bac_activo` char(1) NOT NULL,
  PRIMARY KEY(`bac_codigo`)
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 11264 kB';

```

Tabla: tb_certificado

```

-- Estructura para la tabla `tb_certificado`

CREATE TABLE `tb_certificado` (
  `cer_codigo` int(10) NOT NULL DEFAULT '0',
  `cer_horas` int(5) NOT NULL DEFAULT '0',
  `cer_cargo` varchar(50) NOT NULL,

```

```

    `cer_nombre` varchar(50) NOT NULL,
    PRIMARY KEY(`cer_codigo`)
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 11264 kB';

```

Tabla: tb_colegio

```
-- Estructura para la tabla `tb_colegio`
```

```

CREATE TABLE `tb_colegio` (
  `col_codigo` int(10) NOT NULL DEFAULT '0',
  `col_nombre` varchar(50) NOT NULL,
  `col_activo` char(1) NOT NULL,
  PRIMARY KEY(`col_codigo`)
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 11264 kB';

```

Tabla: tb_datos

```
-- Estructura para la tabla `tb_datos`
```

```

CREATE TABLE `tb_datos` (
  `dat_codigo` int(10) NOT NULL DEFAULT '0',
  `alu_carrera` int(10) NOT NULL DEFAULT '0',
  `alu_cedula` varchar(11) NOT NULL DEFAULT '0',
  `dat_tipo` char(1) NOT NULL,
  `dat_fecdesde` date NOT NULL DEFAULT '0000-00-00',
  `dat_fechasta` date NOT NULL DEFAULT '0000-00-00',
  `dat_aprobado` char(1) NOT NULL,
  `dat_horas` int(10) NOT NULL DEFAULT '0',
  `res_codigo` int(10) NOT NULL DEFAULT '0',
  `lug_codigo` int(10) NOT NULL DEFAULT '0',
  `dat_observaciones` varchar(150),

```

```

`dat_estado` date,
PRIMARY KEY(`dat_codigo`),
INDEX `alu_carrera`(`alu_carrera`),
INDEX `alu_cedula`(`alu_cedula`),
INDEX `dat_tipo`(`dat_tipo`),
INDEX `res_codigo`(`res_codigo`),
INDEX `lug_codigo`(`lug_codigo`),
CONSTRAINT `tb_datos_fk2` FOREIGN KEY (`dat_tipo`)
  REFERENCES `tb_tipo_dato`(`tip_codigo`)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT,
CONSTRAINT `tb_datos_fk3` FOREIGN KEY (`res_codigo`)
  REFERENCES `tb_responsable`(`res_codigo`)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT,
CONSTRAINT `tb_datos_fk4` FOREIGN KEY (`lug_codigo`)
  REFERENCES `tb_lugar`(`lug_codigo`)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 11264 kB; (`alu_carrera`) REFER
`evaluacion/carrera_estudiante`(`COD_CARRERA`); (`alu_cedula`)
REFER `evaluacion/estudiantes`(`CEDULA`); (`dat_tipo`) REFER
`extension/tb_tipo_dato`(`tip_codigo`); (`res_codigo`) REFER
`extension/tb_responsable`(`res_codigo`); (`lug_codigo`) REFER
`extension/tb_lugar`(`lug_codigo`));'

```

Tabla: tb_horario

```
-- Estructura para la tabla `tb_horario`
```

```
CREATE TABLE `tb_horario` (
  `dat_codigo` int(10) NOT NULL DEFAULT '0',
  `hor_horentma1` time,
  `hor_horsalma1` time,
  `hor_horentta1` time,
  `hor_horsalta1` time,
  `hor_horentma2` time,
  `hor_horsalma2` time,
  `hor_horentta2` time,
  `hor_horsalta2` time,
  `hor_horentma3` time,
  `hor_horsalma3` time,
  `hor_horentta3` time,
  `hor_horsalta3` time,
  `hor_horentma4` time,
  `hor_horsalma4` time,
  `hor_horentta4` time,
  `hor_horsalta4` time,
  `hor_horentma5` time,
  `hor_horsalma5` time,
  `hor_horentta5` time,
  `hor_horsalta5` time,
  `hor_horentma6` time,
  `hor_horsalma6` time,
  `hor_horentta6` time,
  `hor_horsalta6` time,
  PRIMARY KEY(`dat_codigo`),
  CONSTRAINT `tb_horario_fk` FOREIGN KEY (`dat_codigo`)
    REFERENCES `tb_datos`(`dat_codigo`)
    ON DELETE RESTRICT
```

```

        ON UPDATE RESTRICT
    )
ENGINE=INNODB
COMMENT = 'InnoDB free: 11264 kB; (`dat_codigo`) REFER
`extension/tb_datos`(`dat_codigo`);'

```

Tabla: tb_informacion

```
-- Estructura para la tabla `tb_informacion`
```

```

CREATE TABLE `tb_informacion` (
    `inf_codigo` int(10) NOT NULL DEFAULT '0',
    `inf_descripcion` varchar(50) NOT NULL,
    `inf_activo` char(1) NOT NULL,
    PRIMARY KEY(`inf_codigo`)
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 11264 kB';

```

Tabla: tb_lugar

```
-- Estructura para la tabla `tb_lugar`
```

```

CREATE TABLE `tb_lugar` (
    `lug_codigo` int(10) NOT NULL DEFAULT '0',
    `lug_nombre` varchar(50) NOT NULL,
    `lug_direccion` varchar(50) NOT NULL,
    `lug_telefono` varchar(10),
    `lug_fecha` date,
    PRIMARY KEY(`lug_codigo`),
    INDEX `datos_lugarIdx2`(`lug_codigo`)
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 11264 kB';

```

Tabla: tb_radio

```
-- Estructura para la tabla `tb_radio`

CREATE TABLE `tb_radio` (
  `rad_codigo` int(10) NOT NULL DEFAULT '0',
  `rad_nombre` varchar(50) NOT NULL,
  `rad_activo` char(1) NOT NULL,
  PRIMARY KEY(`rad_codigo`)
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 11264 kB';
```

Tabla: tb_responsable

```
-- Estructura para la tabla `tb_responsable`

CREATE TABLE `tb_responsable` (
  `res_codigo` int(10) NOT NULL DEFAULT '0',
  `res_nombre` varchar(50) NOT NULL,
  `res_telefono` varchar(10),
  `res_fecha` date,
  PRIMARY KEY(`res_codigo`),
  INDEX `datos_responsableIdx2`(`res_codigo`)
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 11264 kB';
```

Tabla: tb_servicios

```
-- Estructura para la tabla `tb_servicios`

CREATE TABLE `tb_servicios` (
  `ser_codigo` int(10) NOT NULL DEFAULT '0',
  `ser_descripcion` varchar(50) NOT NULL,
  `ser_activo` char(1) NOT NULL,
```

```

        PRIMARY KEY(`ser_codigo`)
    )
ENGINE=INNODB
COMMENT = 'InnoDB free: 11264 kB';

```

Tabla: tb_tipo_dato

```

-- Estructura para la tabla `tb_tipo_dato`

CREATE TABLE `tb_tipo_dato` (
  `tip_codigo` char(1) NOT NULL,
  `tip_descripcion` varchar(50) NOT NULL,
  `tip_fecha` date,
  PRIMARY KEY(`tip_codigo`),
  UNIQUE INDEX `tip_codigo`(`tip_codigo`)
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 11264 kB';

```

Tabla: tb_usuarios

```

-- Estructura para la tabla `tb_usuarios`

CREATE TABLE `tb_usuarios` (
  `usu_codigo` int(10) NOT NULL DEFAULT '0',
  `usu_usuario` varchar(10) NOT NULL,
  `usu_clave` varchar(10) NOT NULL,
  `usu_nombre` varchar(50) NOT NULL,
  `usu_adm` char(1) NOT NULL,
  `usu_ultimo` date,
  PRIMARY KEY(`usu_codigo`),
  INDEX `fk_maaccesos_menu__mausuariosIdx2`(`usu_usuario`)
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 11264 kB';

```

3.3.2 Descripción y Codificación de Clases

3.3.2.1 Class conexion

Esta clase realiza la conexión con la base de datos del sistema.

```

class conexion {
    var $link;
    var $base_datos;
    var $mensaje;
    var $color;

    function hacer_conexion ($bd = "extension") {
        @$this -> link = mysql_connect ("localhost", "root");
        if (!(($this -> link)) {
            $this -> mensaje = "ERROR: Se presento un error en la cadena de
conexión!<br>Intente más tarde...";
            $this -> color = "#FF0000";
        }
        else {
            $this -> base_datos = mysql_select_db ($bd, $this -> link);
            if (!(($this -> base_datos)) {
                $this -> mensaje = "ERROR: Se produjo un error al seleccionar
la base de datos!<br>Intente más tarde";
                $this -> color = "#FF0000";
            }
        }
    }
}

```

3.3.2.2 Class as-diagrams: Gráficos de barras con HTML y CSS

Es un script de PHP que no usa librerías gráficas ni código java o javascript, por lo tanto puede trabajar en cualquier plataforma o navegador.

3.3.2.2.1 Instalando as-diagrams

Descargar del Internet el archivo distribuible as-diagrams.zip y descomprimirlo en una carpeta temporal. Si se va a usar la clase en más de una página, colocar as-diagrams.php en uno de los directorios listados en el PHP.ini en la variable "include_path". Caso contrario, colocarlo en el directorio del script principal.

Mover o copiar el contenido de la carpeta img a la carpeta de imágenes común o crear una nueva (el path relativo a ésta carpeta se modifica en la variable \$imgpath)

Abrir el archivo del script as-diagrams.php en un editor de texto y modificar:

```
var $imgpath = 'img/'; // colocar todas las imágenes de graficación  
en ésta carpeta o modificar el nombre de donde se encuentran.
```

```
$bt_total = ... - Localizar el título de 'Totales' de ser necesario.
```

\$imgpath = ... - Setear el path relativo a la carpeta de imágenes.

Se pueden ajustar las hojas de estilo en as-diagrams.php para seguir el diseño del sitio donde quiere ser aplicado.

3.3.2.2 Usando la clase CAsBarDiagram

Se pueden generar gráficos de barras a partir de un array precalculado de datos o a través de una sentencia de SQL enviada a la variable objeto CAsBarDiagram.

Incluir el modulo en el script donde será usado: `require_once('as-diagrams.php');`

Definir el objeto var de la clase CAsBarDiagram, y setear las variables que sean necesarias:

```
$graph = new CAsBarDiagram;
```

```
$graph->bwidth = 15; // Ancho de barra en pixeles.
```

```
$graph->showdigits = 0; // Ocultar la parte de "representación numérica" debajo de las barras (activo por defecto).
```

```
$graph->bt_total = 'Totales'; // Título para la columna de Totales al lado derecho.
```

\$graph->precision = 0; // Número de Dígitos después del punto decimal.

\$graph->showtotals = 0; // Ocultar columna de Totales, (activo por defecto).

Hacer los arrays de una dimensión para las “leyendas” – uno para el eje X (\$legend_x), y uno para el eje Y (\$legend_y).

\$legend_x = array('Enero','Febrero','Marzo','Abril','Mayo','Junio');

\$legend_y = array('esferos','lápices','clips','papel');

Se pueden escoger entre 2 métodos:

- Colocando los datos en \$data[x][y], y pasar \$data al método para graficar de la clase CAsBarDiagram:

\$data = array(array(200,300,120), array(450,500,100),

array(650,200,300), array(380,200,90),array(750,340,200));

\$graph->DiagramBar(\$legend_x, \$legend_y, \$data, 1, 'Ventas');

- Llamar al método GatherData, pasando la sentencia SQL, los datos serán graficados a partir del los resultados obtenidos por la consulta (No olvidar hacer antes la conexión a la base MySQL):

```
$graph->InitData($legend_x, $legend_y); // resetear los datos
internos del array de ser necesario.
```

```
$graph->GatherData($SQLqry, $legend_x, $legend_y);
```

```
// ... GatherData puede ser llamado más de una vez.
```

```
$graph->DiagramBar($legend_x, $legend_y, 0, $data_title);
```

Notar que en el segundo caso es pasado un valor escalar (0) al método DiagramBar() en lugar del array de datos, por lo tanto la clase escoge el array interno de datos formado por las llamadas a GatherData.

3.3.2.2.3 CAsBarDiagram métodos y variables

InitData(\$legend_x, \$legend_y) - Este método vacía el array de datos interno. Esto es necesario si se usa el objeto var CAsBarDiagram para gráficar múltiples barras, antes de recolectar datos para la segunda, tercera, etc barras. La dimensión interna de datos llega a ser igual a count(\$legend_x), count(\$legend_y).

GatherData(\$sqltext, \$legend_x, \$legend_y [, \$position_y]) - Ejecuta la sentencia SQL enviada en la conexión MySQL actualmente abierta y llena el array interno con los datos resultantes. Se pueden llenar todas las columnas del array si la sentencia SQL devuelve tres campos:

El primero es la "Llave" para el eje X, el segundo para el eje Y, y el tercer campo es el valor de la posición Y que se está llenando del array de datos.

```
SELECT MONTH(ventas_fecha) AS mes, producto,
SUM(ventas_valor) FROM sales WHERE
YEAR(ventas_fecha)=2005 GROUP BY mes,producto
```

Si se quiere llenar sólo una fila del array de datos, se debe pasar una sentencia SQL de dos campos y un cuarto parámetro opcional, definiendo la posición Y de la columna de datos que está siendo llenada.

DiagramBar(\$legend_x, \$legend_y, [\$data0, [\$data_title]]) - Es el método principal, dibuja el gráfico de barras. Si el tercer parámetro es omitido o se envía un valor escalar, el array de datos interno es interpretado. En caso de ser enviado un array, CAsBarDiagram lo usa en lugar de los datos internos almacenados si es llamado antes GatherData().

\$data - Array de datos bidimensional. \$data[x][y] – valores para 'Y' en la sección del gráfico.

\$showdigits - Activa la representación numérica bajo las barras (tabla de representación numérica de los datos del array)

\$ShowPercents - Al asignar un cadena que no está vacía a ésta variable, aparecerán los valores de porcentajes: una columna adicional es generada

en la parte de representación numérica bajo las barras. Esta fila contendrá valores porcentuales de los dos números previos $\$data[n-1]/\$data[n]$.

```
$graph->ShowPercents = 'Porcentajes';
```

```
$graph->DiagramBar($legend_x, $legend_y, 0, $data_title);
```

Las siguientes tres variables, son usadas para convertir los títulos de leyenda a enlaces URL.

legendx_url - Esta es una variable de clase, si se quiere que los títulos de leyendas en el gráfico sirvan de enlaces URL. Solamente colocar el URL deseado en ésta variable:

```
$graph = new CAsBarDiagram;
```

```
$graph->legendx_url = 'detailedinfo.php?info_id={ID}';
```

Después, los títulos de leyendas se convierten en etiquetas `<A HREF...>`'s a la URL indicada. El "{ID}" será sustituido con el valor del título o con el valor del elemento `legendx_id` del array.

legendx_onClick - Esta variable tiene un evento "onClick", si se desea correr algún código JavaScript en lugar de abrir una URL, colocar el evento aquí. Por ejemplo, para abrir otra ventana que contenga datos detallados, se puede llenar la variable de la siguiente manera:

```
$graph->legendx_onClick = "window.open('details.php?id={ID}',
'_blank','height=300,width=400');";
```

legendx_id - {ID} array de valores que serán usados para crear leyendas de anclaje. Si no es inicializado, el título de leyenda X es substituido por valores {ID} en los hrefs y definiciones "onClick".

Otras características acerca del uso de la clase son:

- Los datos pueden contener valores negativos, en este caso la parte negativa será graficada.
- Se puede establecer el ancho de la barra con la variable \$bwidth antes de hacer el llamado a DiagramBar();
- Si el array de datos contiene muchas líneas el ancho será ajustado automáticamente (\$bwidth variable). El ancho máximo es 1024 pixeles (para la resolución estandar de un monitor 1024 x 768).

Si se quiere desactivar este comportamiento, comentar en as-diagrams.php la línea de código: "**var \$autoshrink = 1024;**"

También se puede cambiar el máximo de ancho de pantalla aquí.

- El eje Y es siempre dividido en 4 partes, y cada parte tiene una altura de 60 pixeles (por lo tanto, la máxima altura del eje Y es 240 px). El

archivo imagen "bar-bg240.png" es usado como background por el área principal del gráfico. Si se quiere cambiar el background, modificar éste archivo.

- Si se quiere modificar el color de las barras o su orden, editar el código de as-diagrams.php o sustituir los archivos bar-*.png.
- Si se planea trabajar con valores pequeños o no enteros se debe ajustar el número de dígitos decimales con la variable \$precision, setear el valor deseado antes del llamado a DiagramBar():

```
$precision = 4;
```

La precisión decimal por defecto es 2.

3.4 Pruebas e Instalación

3.4.1 Pruebas

Se realizaron todas las pruebas necesarias de forma local antes de la puesta en marcha del sistema por la Universidad.

3.4.2 Instalación

Contenido del disco de Instalación: El disco de instalación del sistema SEE contiene:

- Un directorio con los archivos del sitio Web llamado “**extension**”.
- Un archivo denominado **extension.dump** donde se encuentra la estructura de la base de datos.
- Programa de instalación **PHP 4.2.2**. (Para S.O. Windows y Solaris)
- Programa de instalación **MySQL 4.1.14**. (Para S.O. Windows y Solaris)

Pasos para la Instalación del Sistema:

- Como primer paso, procedemos con la instalación del compilador de PHP obligatorio para el funcionamiento de la interfaz del sistema. Instalamos la versión de PHP 4.2.2 con el programa de instalación del mismo (si en el sistema operativo previamente no se encuentra instalado o existe una versión inferior). Luego, modificamos el archivo “**php.ini**” que se instaló en el sistema operativo, con la finalidad de permitir el paso de variables entre formularios; seteamos en “**On**” la opción “**register_globals**” y grabamos la modificación.

register_globals = On

- Ahora instalamos el motor de base de datos MySQL. Instalamos la versión de MySQL 4.1.14 con el programa de instalación del mismo (si en el sistema operativo previamente no se encuentra instalado o existe una versión inferior).
- Grabamos el sitio Web que contiene los archivos del sistema en el directorio respectivo:
 - **Windows.** “**c:/inetpub/wwwroot/extension**”.
 - **Solaris.** “**/home/usuario/extension**”.
- Para cargar la base de datos del sistema es necesario copiar el archivo “**extension.dump**” en el directorio “**bin**” contenido en el directorio de MySQL.
 - **Windows.** “**c:/mysql/bin**”.
 - **Solaris.** “**/usr/local/mysql/bin**”.
- Por último ingresamos a MySQL, creamos la base de datos con el nombre “**extension**” y cargamos el archivo “**extension.dump**” para crear la estructura de la base.

```
shell> mysql
```

```
mysql> CREATE DATABASE extension;
```

```
mysql> USE extension;
```

```
mysql> SET FOREIGN_KEY_CHECKS=0;
```

```
mysql> SOURCE extension.dump;
```

Importante. El nombre de la base de datos debe ser idéntico al que se indica.
Antes de cargar la base de datos es necesario setear en 0 la verificación de claves foráneas que tiene MySQL.

4 CAPÍTULO IV. VERIFICACIÓN Y VALIDACIÓN DE RESULTADOS

4.1 Verificación

La hipótesis planteada fue que con la implementación de un Sistema Informático para la Dirección de Estudiantes, se facilitará el control y aprobación de ejecución de las Actividades de Extensión Universitaria y la obtención de Información estadística de los estudiantes.

En el período de evaluación realizado, se comprobó la hipótesis planteada, llegando a cumplir los objetivos planteados de manera muy satisfactoria en beneficio de las actividades realizadas por la Dirección de Estudiantes.

El sistema reveló un ahorro de tiempo en la obtención de información referente a las actividades de Extensión Universitaria y Estadísticas de suma importancia para la Dirección de Estudiantes.

Los reportes obtenidos a partir del uso del sistema ayudan a la Dirección de Estudiantes en el control y obtención de información relevante para el mejoramiento de su organización y del servicio a los estudiantes de la Universidad.

4.2 Validación

4.3 Conclusiones

- Se logró la automatización en el control de Actividades de Extensión Universitaria y generación de Gráficos Estadísticos sobre información útil para la Dirección de Estudiantes.
- Se obtuvo un ahorro de tiempo en el proceso de recopilación de información y reportes de Actividades de Extensión de los Estudiantes así como en la obtención de información a nivel estadístico.
- Se consiguió facilitar el análisis de información de mucho interés para la Dirección de Estudiantes a través de reportes estadísticos.
- La utilización de software y herramientas de libre disposición y de alto nivel de compatibilidad permiten que el sistema tenga su propia independencia y brinde las mejores ventajas en relación a su costo de implementación.

4.4 Recomendaciones

- Para el correcto funcionamiento del sistema es importante un buen entendimiento del tipo de información que éste maneja, de modo que la misma se registre y manipule correctamente.

- Para los usuarios administradores es importante revisar el manual de usuario para poder dar mantenimiento de forma adecuada a la base de datos.
- Es necesario que el administrador del sitio Web y la base de datos del sistema revise el manual técnico, diccionario de datos, mapa del sitio y procesos de instalación que este documento contiene con el propósito de mantener la integridad del sistema y poder darle mantenimiento de ser necesario.
- En el proceso de montaje de la base de datos es de suma importancia mantener el nombre de la base de datos exactamente igual al que indica el proceso de instalación.
- No se deben descuidar la obtención periódica de respaldos que garanticen la integridad de la información valiosa contenida en la Base de Datos manejada por el Sistema.
- Es importante tener en cuenta los requerimientos de hardware y software especificados en el Manual Técnico incluido en éste documento, así como la instalación del Sistema en conjunto con la Base de Datos del SED (Sistema de Evaluación Docente) indispensable para su funcionamiento.

ANEXOS

A. Manual Técnico

Ámbito Y Funciones

ÁMBITO

El Sistema de Control de Actividades de Extensión Universitaria y Estadísticas está dirigido a la Dirección de Estudiantes de la PUCESA y tomará el nombre de SEE.

PROBLEMAS A SOLUCIONAR

- Brindará un control Automatizado del proceso de control de actividades de Extensión Universitaria y obtención de estadísticas sobre información de admisión de los alumnos de la Universidad.
- Proveerá de forma rápida información referente a las actividades de extensión e información estadística de los registros de admisión de los estudiantes.
- Ofrecerá una fácil manipulación e interacción con la información registrada.

FUNCIONES

Proceso de Registro y Mantenimiento de Actividades de los Alumnos

Es el que permite el registro y manipulación de la información referente a las actividades realizadas por los alumnos para su posterior utilización en los reportes.

Proceso de Registro y Mantenimiento de Información obtenida en la Admisión

Es el que permite el registro y manipulación de la información referente a resultados de Examen de Aptitud Académica y Encuestas obtenidas en la admisión de los estudiantes para su posterior utilización en los reportes estadísticos.

Modificación de Notas Máximas por bloques de preguntas para el examen de Admisión

Este proceso nos ayudará para la correcta validación de los datos ingresados en lo que respecta a los resultados del examen de aptitud académica.

Generación de Reportes

El SEE proporciona al usuario reportes sobre información registrada acerca de las actividades de Extensión Universitaria de acuerdo a criterios como pueden ser estudiante, carrera, tipo de actividad y estado de actividad (Realizado o Aprobado).

Generación de Estadísticas

El SEE proporciona al usuario reportes estadísticos de barras sobre información registrada en la admisión de los estudiantes como son resultados del examen de aptitud académica, colegios, bachilleratos, medios de obtención de información acerca de la Universidad, servicios que motivaron el ingreso, y preferencias de radioemisoras; de acuerdo a criterios como pueden ser la carrera y la fecha de realización del examen de aptitud y encuestas.

Mantenimiento Base de Datos

Esta función agrupa las tareas referentes al mantenimiento de los datos que son necesarios para el registro y manejo de actividades de extensión y estadísticas.

Especificación de Requisitos de Software SRS

1 INTRODUCCIÓN

1.1 PROPÓSITO

1.1.1 PROPÓSITO DEL DOCUMENTO

Este documento tiene como objetivo el dar a conocer los alcances y delimitaciones que el Sistema de Extensión y Estadísticas tiene al ser implementado en la Universidad. Describe datos técnicos para el administrador referentes al front end y back end del sistema (instalación del sistema y respaldo de datos).

1.1.2 AUDIENCIA A LA QUE VA DIRIGIDO

El presente documento va dirigido a la administración del Centro de Cómputo de la Universidad, encargada de la administración del servidor donde se encuentra alojado el sitio Web de la PUCESA, en especial a los miembros encargados de dar hospedaje y mantenimiento al sistema.

1.2 ALCANCE

1.2.1 IDENTIFICACIÓN DEL PRODUCTO MEDIANTE UN NOMBRE

El sistema de extensión y estadísticas para el presente documento se lo denomina **SEE**.

1.2.2 QUÉ HACE Y QUÉ NO HACE EL PRODUCTO

Sistematizará el control y la obtención de información acerca de las actividades realizadas y estadísticas de registros de admisión de los alumnos para la Dirección de Estudiantes de la PUCESA. El sistema brinda la posibilidad de mantener la integridad de los datos en la creación, modificación y actualización de la base de datos. Genera reportes dependiendo del criterio establecido por el usuario. *Ver también Ámbito y Funciones.*

Los procesos de respaldos y recuperación no se realizan de forma automática, este proceso deberá ser ejecutado de forma manual por el encargado de administrar el equipo donde se encuentra hospedado el sistema. *Ver también Operaciones de Backup y recuperación.*

1.2.3 APLICACIONES DEL SOFTWARE: BENEFICIOS, OBJETIVOS Y METAS

Los beneficios y objetivos que ofrece el sistema son básicamente los de automatizar y sistematizar los procesos de control y obtención de información sobre actividades de extensión y registros de admisión de los alumnos.

1.3 DEFINICIÓN, ACRÓNIMOS Y ABREVIATURAS

SEE → Sistema de Extensión y Estadísticas.

1.4 REFERENCIAS

[IEEE 93] IEEE Std 830 1993 (revision of IEEE Std 830 1984). Software engineering Standards Committee of the IEEE Computer Society.

1.5 VISIÓN GENERAL

1.5.1 DESCRIPCIÓN DEL CONTENIDO DEL RESTO DEL DOCUMENTO

Este documento estará enfocado para el correcto funcionamiento del sistema.

1.5.2 ORGANIZACIÓN DEL DOCUMENTO

Estas especificaciones de requisitos software se organizan según las recomendaciones del estándar [IEEE 93].

2 DESCRIPCIÓN GENERAL

2.1 PERSPECTIVA DEL PRODUCTO

2.1.1 INDICAR SI ES UN PRODUCTO INDEPENDIENTE O PARTE DE UN SISTEMA MAYOR

Software dependiente de un compilador de código PHP, motor de base de datos MySQL, módulos o librerías para la conexión de PHP con el motor de base de datos. Los registros referentes a los estudiantes, como son Nombre y Cédula a partir de los cuales se realiza el registro de los datos de actividades y admisión, son tomados de la base “evaluacion” del sistema de Evaluación Docente(SED).

2.1.2 INTERFACES DE SISTEMAS

Interfaz Web interpretada por cualquier browser del lado del cliente.

2.1.3 LIMITACIONES DE MEMORIA

Interfaz. La velocidad de presentación de las pantallas del sistema depende del tipo de conexión a Internet que el cliente tenga.

Base de Datos. MySQL es un motor de base de datos pequeño, optimiza los recursos de memoria.

2.1.4 OPERACIONES

2.1.4.1 MODO DE OPERACIÓN DE LOS DISTINTOS GRUPOS DE USUARIOS

Administradores. Son los encargados de la administración de usuarios que accederán al sistema, además del mantenimiento de la base de datos del mismo.

No Administradores. La única característica que lo diferencia de los Usuarios Administradores es que no tienen la funcionalidad de administrar los usuarios que accederán al sistema.

2.1.4.2 FUNCIONES RESPALDOS DEL PROCESAMIENTO DE DATOS

El encargado de administrar el equipo donde se encuentra hospedado el sistema es el responsable de realizar las funciones de respaldo de datos de forma manual. *Ver también Operaciones de Backup y Recuperación.*

2.1.4.3 OPERACIONES DE BACKUP Y RECUPERACIÓN

Interfaz. La totalidad de los archivos que comprenden al SEE deben hospedarse en servidor Web. Para Windows copiamos toda la carpeta “**extension**” dentro del directorio “**wwwroot**”. Para Solaris es necesario copiarla en el directorio “**var/apache**”. Para obtener un respaldo del sitio solo es necesario copiar toda la carpeta que contiene a los archivos del sistema.

Base de Datos. Para obtener un respaldo de la base de datos necesariamente se lo debe realizar de forma manual ejecutando una función propia de MySQL que genera un archivo de respaldo **.dump**.

```
mysql> exit;
```

```
shell> bye
```

```
shell> mysqldump extension > extension.dump
```

Para recuperar una base de datos es necesario copiar el archivo de respaldo “**extension.dump**” dentro del directorio raíz donde se encuentra situado MySQL, luego manualmente dentro de la consola del motor de base de datos MySQL, lo primero es crear la base de datos para luego cargar las tablas y datos de la base usando la instrucción “**source**”.

```
shell> mysql
mysql> CREATE DATABASE extension;
mysql> USE extension;
mysql> SOURCE extension.dump;
```

2.1.5 REQUERIMIENTOS PARA ADAPTARSE A LA UBICACIÓN

2.1.5.1 DATOS O SECUENCIAS DE INICIALIZACIÓN ESPECÍFICOS DE CUALQUIER LUGAR, MODO DE OPERACIÓN

- Instalación de Motor de Base de Datos Mysql, versión que soporte tablas InnoDB – versión 4.1.14 o superior.
- Instalación del compilador PHP, versión 4.2.2 o superior.
- (Windows) Configuración del Internet Information Server [IIS] de Windows versión 4 o superior.
- (Solaris) Instalación del Apache y módulos de conexión de PHP con MySQL.

- Configuración del archivo php.ini para permitir al compilador de PHP el pase de variables a través de formularios.

2.1.5.2 CARACTERÍSTICAS QUE DEBEN SER MODIFICADAS PARA UNA INSTALACIÓN EN PARTICULAR

Dependiendo del equipo donde se hospeda la base de datos, los parámetros que utiliza el sistema para la conexión con la base podrían cambiar. El sistema utiliza la clase “**conexion**” programada en PHP para realizar esta función, la clase posee la función de conexión “**mysql_connect ()**” que abre una conexión a un servidor MySQL. Esta función se encuentra definida de forma local, por nombre del servidor “**localhost**” y nombre del usuario “**root**” por defecto.

```
mysql_connect ("localhost", "root");
```

Si es necesario cambiar estos valores, la estructura de la función es:

```
mysql_connect ([string server], [string username], [string password]);
```

La página del sitio Web que se debería modificar es: **conexion.php**.

2.2 FUNCIONES DEL PRODUCTO

- Registro de Actividades de Extensión
- Registro de Datos de Admisión
- Generación de Reportes
- Generación de Estadísticas
- Mantenimiento Base de Datos

Ver también Ámbito – Funciones.

2.3 CARACTERÍSTICAS DEL USUARIO

Debe tener conocimiento sobre el manejo, control y administración del sistema.

2.4 RESTRICCIONES

- El sistema solo podrá almacenar datos de alumnos que se encuentren registrados en el sistema de Evaluación Docente.
- El sistema no maneja dinámicamente el ingreso de nuevas categorías en bloques de preguntas para el examen de aptitud.

2.5 SUPOSICIONES Y DEPENDENCIAS

Depende del correcto funcionamiento del servidor Web o equipo donde se encuentre alojado, instalación de los compiladores básicos para la funcionalidad del sistema y el motor de base de datos, además de la instalación de su base de datos en conjunto con la del Sistema de Evaluación Docente(SED) debido a su interacción con el maestro de estudiantes de la misma.

2.6 REQUISITOS PARA FUTURAS VERSIONES DEL SISTEMA

Si se desea instalar una nueva versión del sistema o nueva base de datos, es necesario que todos los componentes, librerías o módulos de ésta sean soportados por el software y hardware del servidor Web donde será instalado.

3 REQUISITOS ESPECÍFICOS

3.1 REQUISITOS DE INTERFAZ EXTERNA

3.1.1 INTERFACES DE USUARIO

La interfaz Web del usuario puede ser interpretada por cualquier browser.

Esta consta de menús, formularios, navegador y buscador interno del sistema.

3.1.2 INTERFACES DE SOFTWARE

PHP se ejecuta y conecta del lado del servidor con MySQL.

3.2 REQUISITOS FUNCIONALES

Windows:

- PHP versión 4.2.2
- MySQL versión 4.1.14
- Internet Information Server versión 4 o superior

Solaris:

- PHP versión 4.2.2
- MySQL versión 4.1.14
- Apache

3.2.1 FLUJOS DE INFORMACIÓN

3.2.1.1 DIAGRAMA DE FLUJO DE DATOS NIVEL 0

Ver diagrama de flujo de Datos Nivel 0 del Capítulo 3 del Desarrollo de Proyecto.

3.2.1.2 DIAGRAMA DE FLUJO DE DATOS NIVEL 1

Ver diagrama de flujo de Datos Nivel 1 del Capítulo 3 del Desarrollo de Proyecto.

3.3 REQUISITOS DE RENDIMIENTO

Espacio en disco para alojamiento de la base de datos de mínimo 10 Mb.

B. Manual de Usuario

INTRODUCCION

El objetivo de éste manual es el de servir como instructivo para la utilización del Sistema de Extensión y Estadísticas SEE por parte de los usuarios de la Dirección de Estudiantes de la PUCESA, su contenido ayudará a la comprensión de su funcionamiento e interacción con el mismo.

INGRESO AL SISTEMA

El ingreso al Sistema, se realiza a través de una validación de usuario y contraseña:

| | |
|---|----------------------|
| Usuario: | <input type="text"/> |
| Clave: | <input type="text"/> |
| <input type="button" value="Ingresar"/> | |

En la página principal, ingresamos el Usuario, la Clave de ingreso y presionamos el botón “Ingresar”.

Si el Usuario y la Clave son validados por el sistema, se nos permitirá visualizar el menú de opciones del sistema disponibles para el usuario ingresado.

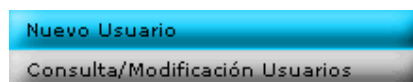


USUARIOS

Esta opción es la única que es exclusiva para los usuarios Administrador, encierra las operaciones relacionadas con la administración de Usuarios del Sistema; está conformado por las siguientes opciones:



Ingreso de un Nuevo Usuario del Sistema. Si queremos ingresar un nuevo Usuario del Sistema, hacemos “clic” en la opción “Nuevo Usuario” del menú “USUARIOS”.



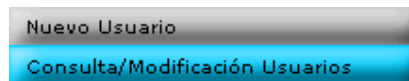
Esta opción, nos enviará al formulario de Ingreso de Nuevo Usuario.

Insertar Usuario

| | |
|-----------------------------|---|
| Codigo:* | <input type="text" value="4"/> |
| Usuario:* | <input type="text"/> máx. 10 caracteres |
| Clave:* | <input type="text"/> máx. 10 caracteres |
| Ingresar Clave Nuevamente:* | <input type="text"/> |
| Nombre:* | <input type="text"/> |
| Administrador:* | <input type="text" value="N"/> |

Ingresamos los datos solicitados y presionamos el botón “Insertar” para guardarlos. Los campos que tienen el símbolo asterisco “*” son requeridos. De existir algún error u omisión en el ingreso, una vez presionado el botón “Insertar” aparecerá por pantalla un listado de los campos que necesitarán ser corregidos.

Modificar/Eliminar un Usuario Existente. Si queremos modificar o eliminar los datos de un Usuario existente, hacemos “clic” en la opción “Consulta/Modificación Usuarios” del menú “USUARIOS”.



Esta opción, nos enviará a la página de búsqueda de Usuarios.



Podemos ingresar una Clave de Búsqueda para filtrar los usuarios, y presionamos el botón “Buscar”, o hacemos “clic” en el enlace “Mostrar Todos” para listar todos los Usuarios existentes.

| Reg. # | CODIGO | USUARIO | CLAVE | NOMBRE | ADMINISTRADOR | FECHA ULTIMO INGRESO | Nuevo |
|--------|--------|---------|---------|---------|---------------|----------------------|---|
| 1 | 3 | USUARIO | usuario | USUARIO | S | 0000-00-00 | Editar Borrar |

El enlace “[Nuevo](#)”, nos enviará al formulario de Ingreso de Nuevo Usuario.

El enlace “[Editar](#)”, nos enviará al formulario de Modificación de Usuario, en la cual deberemos modificar los datos deseados y presionar el botón “Modificar” para guardar los cambios.

El enlace “ [Borrar](#) ”, nos enviará al formulario de Eliminación de Usuario, en la cual deberemos presionar el botón “Borrar” para eliminar el registro.

MANTENIMIENTO

Esta opción contempla todas aquellas operaciones relacionadas al Mantenimiento de Datos en el Sistema, es decir, Ingreso, Modificación y Eliminación. Se encuentra dividida en tres bloques:



Ingresar/Modificar/Eliminar Registros de Actividades y Admisión de Alumnos. Si queremos ingresar, modificar o eliminar los datos de Actividades y de Admisión de Alumnos, hacemos “clic” en la opción “Alumno” del menú “MANTENIMIENTO”.



Esta opción, nos enviará a la página de búsqueda de Alumnos.

[Mostrar Todos](#)

Filtrar:

Escoger Carrera

Mostrar Todos los Registros

Carrera:

Apellido/Cédula:

Clave de Búsqueda para Apellido o Cédula

Realizar Búsqueda

Podemos escoger una carrera, ingresar una Clave de Búsqueda de Apellido o Cédula para filtrar los alumnos, y presionamos el botón “Buscar”, o hacemos “clic” en el enlace “Mostrar Todos” para listar todos los Alumnos existentes.

Listado de Alumnos

Resultados para : "180300888-5" ← Condiciones Filtrado

| Reg.# | CARRERA | CEDULA | APELLIDOS y NOMBRES | Ingresar Resultados |
|-------|------------------------|-------------|----------------------------|--|
| 1 | INGENIERIA EN SISTEMAS | 180300888-5 | RUBIO PUTIG DANILO EDUARDO | Detalles Exa.Adm./Encuesta |

Página de 1

Navegador

Número de Registros Encontrados

¡Se Encontró 1 Registro!

Ver Detalles del Alumno

Ingresar Resultados del Exámen de Admisión e Información de Encuestas

Enlaces Rápidos

[Mantenimiento Alumnos](#)
[Salir](#)

El enlace “ [Detalles](#) ”, nos enviará a la página de visualización de Detalles de los registros de Actividades y Admisión del Alumno.

Detalles Alumno

| | |
|----------------------|----------------------------------|
| Carrera: | Ingenieria en Sistemas |
| Cédula: | 180300888-5 |
| Apellidos y Nombres: | RUBIO PUIG DANILO EDUARDO |
| Colegio: | BOLÍVAR |
| Dirección: | PARIS N4395 Y VICTOR HUGO(QUITO) |
| Bachillerato: | FÍSICO MATEMÁTICO |
| e-mail: | danilo@hotmail.com |
| Teléfono: | 098082376 |
| Fecha de Nacimiento: | 1977-01-31 |
| Estado Civil: | SOLTERO |

Actividades

| Reg.# | TIPO | FECHA DESDE | FECHA HASTA | ESTADO | HORAS/NIVELES | RESPONSABLE | LUGAR | OBSERVACIONES | Nuevo |
|-------|-----------|-------------|-------------|--------|---------------|---------------|------------------------------|-------------------------------|--|
| 1 | EXTENSIÓN | 2007-03-01 | 2007-03-30 | R | 200 | JIMENA ANDINO | JARDIN DE INFANTES INGAHURCO | TRABAJO EN EL REFERIDO JARDIN | Detalles Editar Borrar |

Resultado Examen de Admisión

| PALAB. EXCLU. | USO CONECT. | ELIM. ORAC. | COMP. LECT. | COMP. ORAC. | COHER. REDAC. | SERIES NUM. | ANALISIS FIG. | RAZON. MAT. | RAZON. LOG. | SUFIC. DATOS. | CULT. GENER. | TOTAL | FECHA | Editar Borrar |
|---------------|-------------|-------------|-------------|-------------|---------------|-------------|---------------|-------------|-------------|---------------|--------------|-------|------------|---|
| 8/8 | 6/6 | 8/8 | 6/6 | 6/6 | 6/6 | 4/4 | 2/2 | 2/2 | 2/2 | 0/0 | 0/0 | 50 | 2007-07-10 | |

Encuesta

| ORIGEN INFORMACIÓN PUCESA | RADIOEMISORA | SERVICIO MOTIVO INGRESO | COMENTARIO |
|--|--|--|---|
| -FAMILIA-AMIGOS-TRÍPTICOS, AFICHES-CONFERENCIAS EN EL COLEGIO-PRENSA-RADIO | -DISNEY 94.1-BRUJA 97.3-LA MEGA 102.5-HOT 106-RUMBA 88.9 | -INFRAESTRUCTURA-BENEFICIOS ESTUDIANTILES-CARRERAS-NIVEL ACADÉMICO-PERSONAL DOCENTE-AMBIENTE-PRESTIGIO-UBICACIÓN-COSTOS-HORARIOS-PRINCIPIOS CRISTIANOS | Editar Borrar |

- En el bloque “Actividades”:


El enlace “[Nuevo](#)”, nos enviará al formulario de Ingreso de Nueva Actividad.

Insertar Registro

| | |
|---|--|
| Carrera: | <input type="text" value="Ingenieria en Sistemas"/> |
| Cedula: | <input type="text" value="180300888-5"/> |
| Apellidos y Nombres: | <input type="text" value="RUBIO PUIG DANILO EDUARDO"/> |
| Tipo: * | <input type="text" value=""/> Tipos Ya Registrados: - EXTENSIÓN |
| Fecha Desde: * | <input type="text" value=""/> |
| Fecha Hasta: * | <input type="text" value=""/> |
| Aprobado/Realizado: * | <input type="text" value=""/> |
| Horas/Niveles: * | <input type="text" value=""/> |
| Responsable: * | <input type="text" value=""/> |
| | Nuevo: |
| | Código: <input type="text" value="196"/> |
| | Responsable: * <input type="text" value=""/> |
| Teléfono: <input type="text" value=""/> | <input type="text" value=""/> |
| Lugar: * | <input type="text" value=""/> |
| | Nuevo: |
| | Código: <input type="text" value="205"/> |
| | Lugar: * <input type="text" value=""/> |
| | Dirección: * <input type="text" value=""/> |
| Teléfono: <input type="text" value=""/> | <input type="text" value=""/> |
| Observaciones: | <input style="width: 100%;" type="text" value=""/> |
| Fecha Estado: | <input type="text" value="2007-7-24"/> |

Horario

| | Entrada Mañana | Salida Mañana | Entrada Tarde | Salida Tarde |
|-----------|---|---|---|---|
| UNIFICADO | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> |
| | hh:mm | hh:mm | hh:mm | hh:mm |
| Lunes | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> |
| Martes | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> |
| Miércoles | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> |
| Jueves | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> |
| Viernes | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> |
| Sábado | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> | <input type="text" value=""/> : <input type="text" value=""/> |

 **Nota.** La descripción insertada en el campo “Observaciones”, corresponde a la descripción de la actividad realizada, y ésta será la que aparecerá como detalle al momento de generar un “Certificado de Actividades de los Alumnos”.

Ingresamos los datos solicitados y presionamos el botón “Insertar” para guardarlos. Los campos que tienen el símbolo asterisco “*” son requeridos.

De existir algún error u omisión en el ingreso, una vez presionado el botón “Insertar” aparecerá por pantalla un listado de los campos que necesitarán ser corregidos.

El enlace “ [Detalles](#) ”, nos enviará a la página de visualización de Detalles de Actividad del alumno.

El enlace “ [Editar](#) ”, nos enviará al formulario de Modificación de Actividad, en la cual deberemos modificar los datos deseados y presionar el botón “Modificar” para guardar los cambios.

El enlace “ [Borrar](#) ”, nos enviará al formulario de Eliminación de Actividad, en la cual deberemos presionar el botón “Borrar” para eliminar el registro.

- En el bloque “Resultados Examen de Admisión” y “Encuesta”:

El enlace “ [Editar](#) ”, nos enviará al formulario de Modificación del registro de Admisión y Encuesta, en la cual deberemos modificar los datos deseados y presionar el botón “Modificar” para guardar los cambios.

El enlace “ [Borrar](#) ”, nos enviará al formulario de Eliminación del registro de Admisión y Encuesta, en la cual deberemos presionar el botón “Borrar” para eliminar el registro.

El enlace “ [Exa.Adm./Encuesta](#) ”, nos enviará al formulario de Ingreso de Resultados de Examen de Admisión e Información de Encuestas para las Estadísticas. Es el único enlace para éste propósito, si se lo puede visualizar, nos indica que se encuentra pendiente el ingreso de la información, caso contrario, ya fueron ingresados.

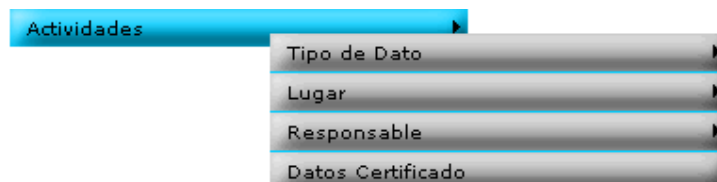
Examen Admisión/Encuesta

| | |
|---------------------------|---|
| Carrera: | Ingeniería en Sistemas |
| Cédula: | 180300888-5 |
| Apellidos y Nombres: | RUBIO PUIG DANILO EDUARDO |
| EXÁMEN ADMISIÓN | |
| Palabras Excluidas:* | <input type="checkbox"/> /8 |
| Uso de Conectores:* | <input type="checkbox"/> /6 |
| Eliminar Oraciones:* | <input type="checkbox"/> /8 |
| Compresión Lectora:* | <input type="checkbox"/> /6 |
| Completar Oraciones:* | <input type="checkbox"/> /6 |
| Coherencia de Redacción:* | <input type="checkbox"/> /6 |
| Serie Numéricas:* | <input type="checkbox"/> /4 |
| Análisis de Figuras:* | <input type="checkbox"/> /2 |
| Razonamiento Matemático:* | <input type="checkbox"/> /2 |
| Razonamiento Lógico:* | <input type="checkbox"/> /2 |
| Suficiencia Datos:* | <input type="checkbox"/> /0 |
| Cultura General:* | <input type="checkbox"/> /0 |
| Fecha:* | <input type="text"/> <small>RR</small> |
| ENCUESTA | |
| Colegio:* | <input type="text"/> |
| Dirección:* | <input type="text"/> |
| Bachillerato:* | <input type="text"/> |
| E-mail: | <input type="text"/> |
| Teléfono: | <input type="text"/> |
| Fecha Nacimiento: | <input type="text"/> <small>RR</small> |
| Estado Civil: | <input type="text"/> |
| Origen Información: | FAMILIA AMIGOS TRIPTICOS, AFICHES CONFERENCIAS EN EL COLEGIO PRENSA RADIO OTROS |
| Radioemisora: | DISNEY 94.1 BRUJA 97.3 LA MEGA 102.5 HOT 106 RUMBA 88.9 OTRA |
| Servicio: | INFRAESTRUCTURA BENEFICIOS ESTUDIANTILES CARRERAS NIVEL ACADÉMICO PERSONAL DOCENTE AMBIENTE PRESTIGIO UBICACIÓN COSTOS HORARIOS PRINCIPIOS CRISTIANOS |
| Comentario: | <input type="text"/> |

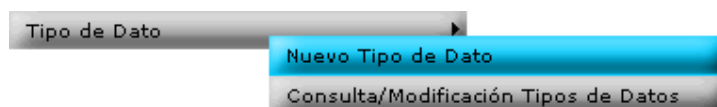
Insertar

Ingresamos los datos solicitados y presionamos el botón “Insertar” para guardarlos. Los campos que tienen el símbolo asterisco “*” son requeridos. De existir algún error u omisión en el ingreso, una vez presionado el botón “Insertar” aparecerá por pantalla un listado de los campos que necesitarán ser corregidos.

El bloque de “Actividades” del menú “MANTENIMIENTO”, tiene las siguientes opciones:



Ingreso de un Nuevo Tipo de Actividad. Si queremos ingresar un nuevo Tipo de Actividad, hacemos “clic” en la opción “Nuevo Tipo de Dato” del menú “MANTENIMIENTO -> Actividades -> Tipo de Dato”.



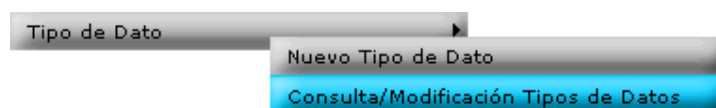
Esta opción, nos enviará al formulario de Ingreso de Nuevo Tipo de Dato.

Insertar Tipo

| | |
|----------------------|----------------------|
| Código:* | <input type="text"/> |
| Descripción:* | <input type="text"/> |

Ingresamos los datos solicitados y presionamos el botón “Insertar” para guardarlos. Los campos que tienen el símbolo asterisco “*” son requeridos. De existir algún error u omisión en el ingreso, una vez presionado el botón “Insertar” aparecerá por pantalla un listado de los campos que necesitarán ser corregidos.

Modificar/Eliminar un Tipo de Actividad Existente. Si queremos modificar o eliminar los datos de un Tipo de Actividad existente, hacemos “clic” en la opción “Consulta/Modificación Tipos de Datos” del menú “MANTENIMIENTO -> Actividades -> Tipo de Dato”.



Esta opción, nos enviará a la página de búsqueda para filtrar el Tipo de Actividad que queramos modificar o eliminar.



Podemos ingresar una Clave de Búsqueda para filtrar los Tipos, y presionamos el botón “Buscar”, o hacemos “clic” en el enlace “Mostrar Todos” para listar todos los Tipos existentes.

| Reg.# | CODIGO | DESCRIPCION | Nuevo |
|-------|--------|-------------|---|
| 1 | E | EXTENSIÓN | Editar Borrar |

El enlace “[Nuevo](#)”, nos enviará al formulario de Ingreso de Nuevo Tipo de Dato.

El enlace “[Editar](#)”, nos enviará al formulario de Modificación de Tipo de Dato, en la cual deberemos modificar los datos deseados y presionar el botón “Modificar” para guardar los cambios.

El enlace “[Borrar](#)”, nos enviará al formulario de Eliminación de Tipo de Dato, en la cual deberemos presionar el botón “Borrar” para eliminar el registro.

Ingreso de un Nuevo Lugar. Si queremos ingresar un nuevo Lugar, hacemos “clic” en la opción “Nuevo Lugar” del menú “MANTENIMIENTO -> Actividades -> Lugar”.



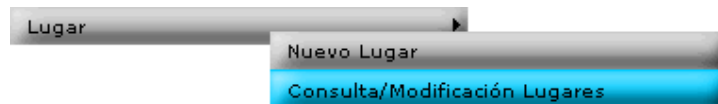
Esta opción, nos enviará al formulario de Ingreso de Nuevo Lugar.

Insertar Lugar

| | |
|--------------------|----------------------------------|
| Código:* | <input type="text" value="205"/> |
| Lugar:* | <input type="text"/> |
| Dirección:* | <input type="text"/> |
| Teléfono: | <input type="text"/> |

Ingresamos los datos solicitados y presionamos el botón “Insertar” para guardarlos. Los campos que tienen el símbolo asterisco “*” son requeridos. De existir algún error u omisión en el ingreso, una vez presionado el botón “Insertar” aparecerá por pantalla un listado de los campos que necesitarán ser corregidos.

Modificar/Eliminar un Lugar Existente. Si queremos modificar o eliminar los datos de un Lugar existente, hacemos “clic” en la opción “Consulta/Modificación Lugares” del menú “MANTENIMIENTO -> Actividades -> Lugar”.



Esta opción, nos enviará a la página de búsqueda para filtrar el Lugar que queramos modificar o eliminar.



Podemos ingresar una Clave de Búsqueda para filtrar los Lugares, y presionamos el botón “Buscar”, o hacemos “clic” en el enlace “Mostrar Todos” para listar todos los Lugares existentes.

| Reg.# | CODIGO | LUGAR | DIRECCION | TELEFONO | Nuevo |
|-------|--------|---------------------|--------------|----------|---|
| 1 | 1 | ESCUELA DE TEOLOGIA | TROPEZON S/N | 03414604 | Editar Borrar |

El enlace “[Nuevo](#)”, nos enviará al formulario de Ingreso de Nuevo Lugar.

El enlace “[Editar](#)”, nos enviará al formulario de Modificación de Lugar, en la cual deberemos modificar los datos deseados y presionar el botón “Modificar” para guardar los cambios.

El enlace “ [Borrar](#) ”, nos enviará al formulario de Eliminación de Lugar, en la cual deberemos presionar el botón “Borrar” para eliminar el registro.

Ingreso de un Nuevo Responsable. Si queremos ingresar un nuevo Responsable, hacemos “clic” en la opción “Nuevo Responsable” del menú “MANTENIMIENTO -> Actividades -> Responsable”.



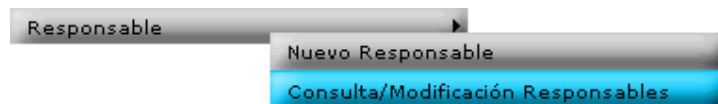
Esta opción, nos enviará al formulario de Ingreso de Nuevo Responsable.

Insertar Responsable

| | |
|----------------------|----------------------------------|
| Código:* | <input type="text" value="196"/> |
| Responsable:* | <input type="text"/> |
| Teléfono: | <input type="text"/> |

Ingresamos los datos solicitados y presionamos el botón “Insertar” para guardarlos. Los campos que tienen el símbolo asterisco “*” son requeridos. De existir algún error u omisión en el ingreso, una vez presionado el botón “Insertar” aparecerá por pantalla un listado de los campos que necesitarán ser corregidos.

Modificar/Eliminar un Responsable Existente. Si queremos modificar o eliminar los datos de un Responsable existente, hacemos “clic” en la opción “Consulta/Modificación Responsables” del menú “MANTENIMIENTO -> Actividades -> Responsable”.



Esta opción, nos enviará a la página de búsqueda para filtrar el Responsable que queramos modificar o eliminar.



Podemos ingresar una Clave de Búsqueda para filtrar los Responsables, y presionamos el botón “Buscar”, o hacemos “clic” en el enlace “Mostrar Todos” para listar todos los Responsables existentes.

| Reg.# | CODIGO | RESPONSABLE | TELEFONO | Nuevo |
|-------|--------|------------------------|----------|---|
| 1 | 1 | ESCUELA DE LINGUISTICA | 03825865 | Editar Borrar |

El enlace “[Nuevo](#)”, nos enviará al formulario de Ingreso de Nuevo Responsable.

El enlace “[Editar](#)”, nos enviará al formulario de Modificación de Responsable, en la cual deberemos modificar los datos deseados y presionar el botón “Modificar” para guardar los cambios.

El enlace “[Borrar](#)”, nos enviará al formulario de Eliminación de Responsable, en la cual deberemos presionar el botón “Borrar” para eliminar el registro.

Modificar Datos para el Certificado de Actividades. Si queremos modificar los datos del Certificado de Actividades Realizadas, hacemos “clic” en la opción “Datos Certificado” del menú “MANTENIMIENTO -> Actividades -> Datos Certificado”.



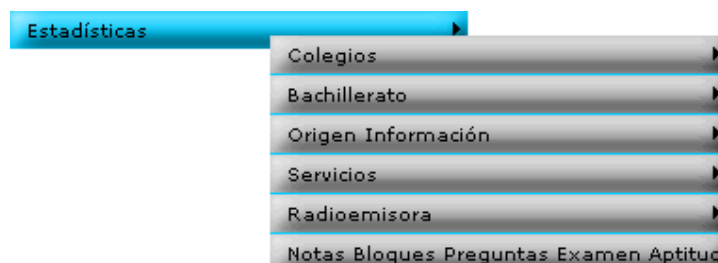
Esta opción, nos enviará al formulario de Modificación de Datos del Certificado de Actividades.

Modificar Datos Certificado

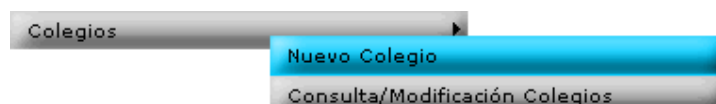
| | |
|----------|---|
| Horas:* | <input type="text" value="200"/> |
| Cargo:* | <input type="text" value="DIRECTORA DE ESTUDIANTES"/> |
| Nombre:* | <input type="text" value="ING. MSC. MARÍA ELENA SANDOVAL"/> |

Modificamos los datos deseados y presionamos el botón “Modificar” para guardar los cambios.

El bloque de “Estadísticas” del menú “MANTENIMIENTO”, tiene las siguientes opciones:



Ingreso de un Nuevo Colegio. Si queremos ingresar un nuevo Colegio, hacemos “clic” en la opción “Nuevo Colegio” del menú “MANTENIMIENTO -> Estadísticas -> Colegios”.



Esta opción, nos enviará al formulario de Ingreso de Nuevo Colegio.

Insertar Colegio

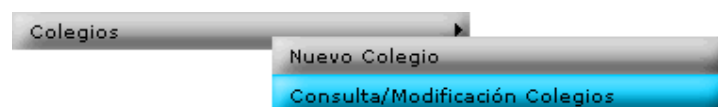
| | |
|-----------|---------------------------------|
| Código:* | <input type="text" value="16"/> |
| Colegio:* | <input type="text"/> |
| Activo:* | <input type="text" value="S"/> |

Insertar

✍ **Nota.** El campo “Activo”, nos permite Activar “S” o Desactivar “N”, la representación del “Colegio” en los gráficos estadísticos.

Ingresamos los datos solicitados y presionamos el botón “Insertar” para guardarlos. Los campos que tienen el símbolo asterisco “*” son requeridos. De existir algún error u omisión en el ingreso, una vez presionado el botón “Insertar” aparecerá por pantalla un listado de los campos que necesitarán ser corregidos.

Modificar/Eliminar un Colegio Existente. Si queremos modificar o eliminar los datos de un Colegio existente, hacemos “clic” en la opción “Consulta/Modificación Colegios” del menú “MANTENIMIENTO -> Estadísticas -> Colegios”.



Esta opción, nos enviará a la página de búsqueda para filtrar el Colegio que queramos modificar o eliminar.



Podemos ingresar una Clave de Búsqueda para filtrar los Colegios, y presionamos el botón “Buscar”, o hacemos “clic” en el enlace “Mostrar Todos” para listar todos los Colegios existentes.

| Reg. # | CODIGO | COLEGIO | ACTIVO | |
|--------|--------|---------|--------|--|
| 1 | 1 | BOLÍVAR | S | Nuevo Editar Borrar |

El enlace “[Nuevo](#)”, nos enviará al formulario de Ingreso de Nuevo Colegio.

El enlace “[Editar](#)”, nos enviará al formulario de Modificación de Colegio, en la cual deberemos modificar los datos deseados y presionar el botón “Modificar” para guardar los cambios.

El enlace “[Borrar](#)”, nos enviará al formulario de Eliminación de Colegio, en la cual deberemos presionar el botón “Borrar” para eliminar el registro.

Ingreso de un Nuevo Bachillerato. Si queremos ingresar un nuevo Bachillerato, hacemos “clic” en la opción “Nuevo Bachillerato” del menú “MANTENIMIENTO -> Estadísticas -> Bachillerato”.



Esta opción, nos enviará al formulario de Ingreso de Nuevo Bachillerato.

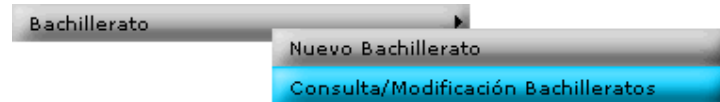
Insertar Bachillerato

| | |
|-----------------------|--------------------------------|
| Código:* | <input type="text" value="8"/> |
| Bachillerato:* | <input type="text"/> |
| Activo:* | <input type="text" value="S"/> |

📌 **Nota.** El campo “Activo”, nos permite Activar “S” o Desactivar “N”, la representación del “Bachillerato” en los gráficos estadísticos.

Ingresamos los datos solicitados y presionamos el botón “Insertar” para guardarlos. Los campos que tienen el símbolo asterisco “*” son requeridos. De existir algún error u omisión en el ingreso, una vez presionado el botón “Insertar” aparecerá por pantalla un listado de los campos que necesitarán ser corregidos.

Modificar/Eliminar un Bachillerato Existente. Si queremos modificar o eliminar los datos de un Bachillerato existente, hacemos “clic” en la opción “Consulta/Modificación Bachilleratos” del menú “MANTENIMIENTO -> Estadísticas -> Bachillerato”.



Esta opción, nos enviará a la página de búsqueda para filtrar el Bachillerato que queramos modificar o eliminar.



Podemos ingresar una Clave de Búsqueda para filtrar los Bachilleratos, y presionamos el botón “Buscar”, o hacemos “clic” en el enlace “Mostrar Todos” para listar todos los Bachilleratos existentes.

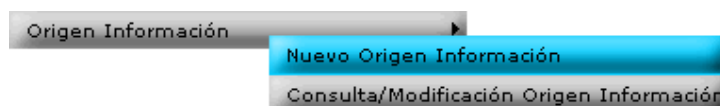
| Reg.# | CODIGO | BACHILLERATO | ACTIVO | Nuevo |
|-------|--------|-------------------|--------|---|
| 1 | 1 | FÍSICO MATEMÁTICO | S | Editar Borrar |

El enlace “[Nuevo](#)”, nos enviará al formulario de Ingreso de Nuevo Bachillerato.

El enlace “[Editar](#)”, nos enviará al formulario de Modificación de Bachillerato, en la cual deberemos modificar los datos deseados y presionar el botón “Modificar” para guardar los cambios.

El enlace “[Borrar](#)”, nos enviará al formulario de Eliminación de Bachillerato, en la cual deberemos presionar el botón “Borrar” para eliminar el registro.


Ingreso de un Nuevo Origen de Información. Si queremos ingresar un nuevo Origen de Información, hacemos “clic” en la opción “Nuevo Origen Información” del menú “MANTENIMIENTO -> Estadísticas -> Origen Información”.



Esta opción, nos enviará al formulario de Ingreso de Nuevo Origen de Información.

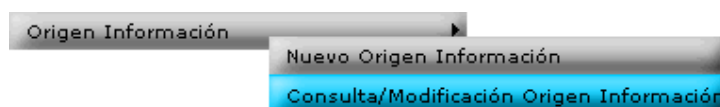
Insertar Origen Información

| | |
|----------------------|----------------------------------|
| Código:* | <input type="text" value="8"/> |
| Origen Información:* | <input type="text"/> |
| Activo:* | <input type="button" value="S"/> |

 **Nota.** El campo “Activo”, nos permite Activar “S” o Desactivar “N”, la representación del “Origen Información” en los gráficos estadísticos.

Ingresamos los datos solicitados y presionamos el botón “Insertar” para guardarlos. Los campos que tienen el símbolo asterisco “*” son requeridos. De existir algún error u omisión en el ingreso, una vez presionado el botón “Insertar” aparecerá por pantalla un listado de los campos que necesitarán ser corregidos.

Modificar/Eliminar un Origen de Información Existente. Si queremos modificar o eliminar los datos de un Origen de Información existente, hacemos “clic” en la opción “Consulta/Modificación Origen Información” del menú “MANTENIMIENTO -> Estadísticas -> Origen Información”.



Esta opción, nos enviará a la página de búsqueda para filtrar el Origen de Información que queramos modificar o eliminar.



Podemos ingresar una Clave de Búsqueda para filtrar los Orígenes de Información, y presionamos el botón “Buscar”, o hacemos “clic” en el enlace “Mostrar Todos” para listar todos los Orígenes de Información existentes.

| Reg. # | CODIGO | ORIGEN INFORMACION | ORIGEN ACTIVO | |
|--------|--------|--------------------|---------------|--|
| 1 | 1 | FAMILIA | S | Nuevo Editar Borrar |

El enlace “[Nuevo](#)”, nos enviará al formulario de Ingreso de Nuevo Origen Información.

El enlace “[Editar](#)”, nos enviará al formulario de Modificación de Origen de Información, en la cual deberemos modificar los datos deseados y presionar el botón “Modificar” para guardar los cambios.

El enlace “[Borrar](#)”, nos enviará al formulario de Eliminación de Origen de Información, en la cual deberemos presionar el botón “Borrar” para eliminar el registro.

Ingreso de un Nuevo Servicio. Si queremos ingresar un nuevo Servicio, hacemos “clic” en la opción “Nuevo Servicio” del menú “MANTENIMIENTO - > Estadísticas -> Servicios”.



Esta opción, nos enviará al formulario de Ingreso de Nuevo Servicio.

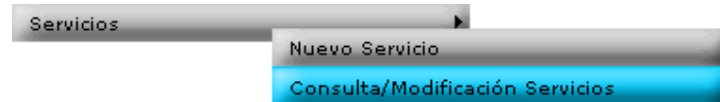
Insertar Servicio

| | |
|-------------------|---------------------------------|
| Código:* | <input type="text" value="12"/> |
| Servicio:* | <input type="text"/> |
| Activo:* | <input type="text" value="S"/> |

📌 **Nota.** El campo “Activo”, nos permite Activar “S” o Desactivar “N”, la representación del “Servicio” en los gráficos estadísticos.

Ingresamos los datos solicitados y presionamos el botón “Insertar” para guardarlos. Los campos que tienen el símbolo asterisco “*” son requeridos. De existir algún error u omisión en el ingreso, una vez presionado el botón “Insertar” aparecerá por pantalla un listado de los campos que necesitarán ser corregidos.

Modificar/Eliminar un Servicio Existente. Si queremos modificar o eliminar los datos de un Servicio existente, hacemos “clic” en la opción “Consulta/Modificación Servicios” del menú “MANTENIMIENTO -> Estadísticas -> Servicios”.



Esta opción, nos enviará a la página de búsqueda para filtrar el Servicio que queremos modificar o eliminar.



Podemos ingresar una Clave de Búsqueda para filtrar los Servicios, y presionamos el botón “Buscar”, o hacemos “clic” en el enlace “Mostrar Todos” para listar todos los Servicios existentes.

| Reg.# | CODIGO | SERVICIO | ACTIVO | Nuevo |
|-------|--------|-----------------|--------|---|
| 1 | 1 | INFRAESTRUCTURA | S | Editar Borrar |

El enlace “[Nuevo](#)”, nos enviará al formulario de Ingreso de Nuevo Servicio.

El enlace “[Editar](#)”, nos enviará al formulario de Modificación de Servicios, en la cual deberemos modificar los datos deseados y presionar el botón “Modificar” para guardar los cambios.

El enlace “[Borrar](#)”, nos enviará al formulario de Eliminación de Servicios, en la cual deberemos presionar el botón “Borrar” para eliminar el registro.

Ingreso de una Nueva Radioemisora. Si queremos ingresar una nueva Radioemisora, hacemos “clic” en la opción “Nueva Radioemisora” del menú “MANTENIMIENTO -> Estadísticas -> Radioemisora”.



Esta opción, nos enviará al formulario de Ingreso de Nueva Radioemisora.

Insertar Radio

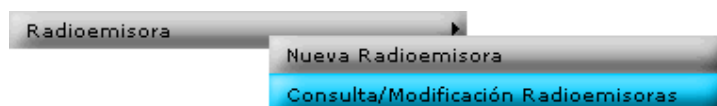
| | |
|----------|--------------------------------|
| Código:* | <input type="text" value="7"/> |
| Radio:* | <input type="text"/> |
| Activo:* | <input type="text" value="S"/> |

Insertar

📌 **Nota.** El campo “Activo”, nos permite Activar “S” o Desactivar “N”, la representación de la “Radio” en los gráficos estadísticos.

Ingresamos los datos solicitados y presionamos el botón “Insertar” para guardarlos. Los campos que tienen el símbolo asterisco “*” son requeridos. De existir algún error u omisión en el ingreso, una vez presionado el botón “Insertar” aparecerá por pantalla un listado de los campos que necesitarán ser corregidos.

Modificar/Eliminar una Radioemisora Existente. Si queremos modificar o eliminar los datos de una Radioemisora existente, hacemos “clic” en la opción “Consulta/Modificación Radioemisoras” del menú “MANTENIMIENTO -> Estadísticas -> Radioemisora”.



Esta opción, nos enviará a la página de búsqueda para filtrar la Radioemisora que queramos modificar o eliminar.



Podemos ingresar una Clave de Búsqueda para filtrar la Radioemisora, y presionamos el botón “Buscar”, o hacemos “clic” en el enlace “Mostrar Todos” para listar todas las Radioemisoras existentes.

| Reg.# | CODIGO | RADIO | ACTIVO | Nuevo |
|-------|--------|-------------|--------|---|
| 1 | 1 | DISNEY 94.1 | S | Editar Borrar |

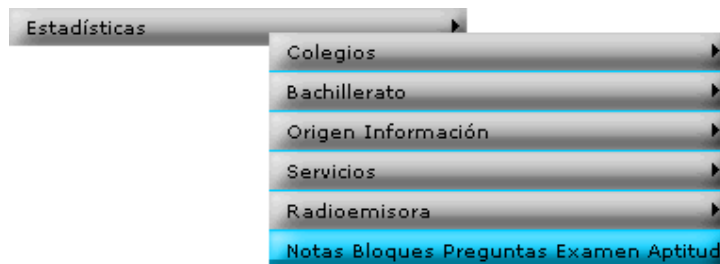
El enlace “[Nuevo](#)”, nos enviará al formulario de Ingreso de Nueva Radioemisora.

El enlace “[Editar](#)”, nos enviará al formulario de Modificación de Radioemisoras, en la cual deberemos modificar los datos deseados y presionar el botón “Modificar” para guardar los cambios.

El enlace “[Borrar](#)”, nos enviará al formulario de Eliminación de Radioemisoras, en la cual deberemos presionar el botón “Borrar” para eliminar el registro.

Modificar Notas Máximas por Bloques de Preguntas del Examen de Aptitud.

Si deseamos modificar las notas máximas por cada bloque de preguntas del examen de aptitud, hacemos “clic” en la opción “Notas Bloques Preguntas Examen Aptitud” del menú “MANTENIMIENTO -> Estadísticas”.



Esta opción, nos enviará al formulario de Modificación de Notas Máximas del Examen de Admisión.

Modificar Notas Máximas de Examen Admisión

| | |
|---------------------------|---|
| Palabras Excluidas:* | 8 |
| Uso de Conectores:* | 6 |
| Eliminar Oraciones:* | 8 |
| Compresión Lectora:* | 6 |
| Completar Oraciones:* | 6 |
| Coherecia de Redacción:* | 6 |
| Series Numéricas:* | 4 |
| Análisis de Figuras:* | 2 |
| Razonamiento Matemático:* | 2 |
| Razonamiento Lógico:* | 2 |
| Suficiencia Datos:* | 0 |
| Cultura General:* | 0 |

Modificar

Modificamos los datos deseados y presionamos el botón “Modificar” para guardar los cambios.

REPORTES

Corresponde a las opciones de Reportes por pantalla e impresión de Registros de Actividades, Admisión y Encuestas de los Alumnos.



Reporte y Certificado de Actividades de los Alumnos. Si queremos obtener un listado por pantalla y para impresora de las Actividades de los Alumnos, hacemos “clic” en la opción “Actividades” del menú “REPORTES”.



Esta opción, nos enviará a la página de búsqueda de Alumnos.

 A screenshot of a search form titled 'Filtros para el Reporte:'. The form contains several input fields and a search button. Red arrows point to specific elements with labels:

- 'Escoger Carrera' points to the 'Carrera' dropdown menu.
- 'Mostrar Todos los Registros' points to the 'Mostrar Todos' link.
- 'Escoger Actividad' points to the 'Tipo' dropdown menu.
- 'Escoger Estado Aprobado o Realizado' points to the 'Estado "A"probado/"R"ealizado:' dropdown menu.
- 'Clave de Búsqueda para Apellido o Cédula' points to the 'Apellido/Cédula:' text input field.

 The 'Buscar' button is located to the right of the 'Apellido/Cédula:' field.

Podemos escoger una carrera, tipo de actividad y su estado, ingresar una Clave de Búsqueda de Apellido o Cédula para filtrar los alumnos y actividades para el reporte, y presionamos el botón “Buscar”, o hacemos “clic” en el enlace “Mostrar Todos” para listar todos los Alumnos existentes y sus actividades.

Reporte Registros de Actividades

Resultados para : "180339261-0"

| * CARRERA | INGENIERÍA COMERCIAL | | CEDULA | 180339261-0 | | APELLIDOS Y NOMBRES | TIRADO GAVILANES TANNYA FERNANDA | |
|-----------|----------------------|-------------|-------------|---------------|------------|-----------------------------|--|--|
| Reg. # | ACTIVIDAD | FECHA DESDE | FECHA HASTA | APROB./REALI. | HORAS/NIV. | RESPONSABLE | LUGAR | OBSERVAC. |
| 1 | EXTENSIÓN | 2006-03-17 | 2006-03-17 | R | 125 | ING. GINO ZAMORA | ESCUELA DE ADMINISTRACION DE EMPRESAS PUCESA | SEGÚN INFORME DEL DIRECTOR DE ESCUELA |
| 2 | EXTENSIÓN | 2007-01-31 | 2007-01-31 | R | 200 | P. JUAN CARLOS ACOSTA | DEPARTAMENTO DE PASTORAL | SEGÚN INFORME DEL DEPARTAMENTO DE PASTORAL |
| Total | | | | | 325 | Certificado | | |

Página 1 de 1

¡Se Encontraron 2 Registros!

Imprimir

Al hacer “clic” en el botón “Imprimir”, nos enviara a la versión para impresora del Reporte de Registros de Actividades.

El enlace “ [Certificado](#) ”, nos enviará a la versión para impresora del Certificado de Actividades Realizadas por el Alumno al que corresponde el enlace.

CERTIFICADO

Por solicitud de RUBIO PUIG DANILLO EDUARDO con C.I. 180300888-5, estudiante de la Escuela de INGENIERIA EN SISTEMAS y luego de un análisis exhaustivo de los archivos de la Dirección de Estudiantes y comparándolo con los Datos arrojados por el sistema, certifico que hasta el 8 de agosto del 2007, ha realizado:

Extensión 200 horas/niveles

200 TRABAJOS EN EL REFERIDO JARDIN

Sin otro particular, me suscribo.

Atentamente,

Ing. Msc. María Elena Sandoval
DIRECTORA DE ESTUDIANTES

[Imprimir](#)

Reporte de Registros de Admisión y Encuestas. Si queremos obtener un listado por pantalla y para impresora de los registros de Admisión y Encuestas de los Alumnos, hacemos “clic” en la opción “Datos Personales/Admisión/Encuestas” del menú “REPORTES”.



Esta opción, nos enviará a la página de búsqueda de Alumnos.

[Mostrar Todos](#)

Filtrar:

Escoger Carrera

Carrera:

Mostrar Todos los Registros

Apellido/Cédula: Buscar

Clave de Búsqueda para Apellido o Cédula

Realizar Búsqueda

Podemos escoger una carrera, ingresar una Clave de Búsqueda de Apellido o Cédula para filtrar los alumnos, y presionamos el botón “Buscar”, o hacemos “clic” en el enlace “Mostrar Todos” para listar todos los registros de Admisión de Alumnos existentes.

Reporte Registros de Admisión

Resultados para : "180428626"

| No. 1 | CARRERA | INGENIERÍA COMERCIAL | CÉDULA | 180428626-6 | APELLIDOS y NOMBRES | ALVAREZ CORDOVA MARIA AUGUSTA | | | | | | | |
|---|---------------------|--------------------------------|---|---------------------|----------------------|-------------------------------|----------------------|--------------------|--------------------|----------------------|---------------------|--------------|--------------|
| Datos Personales | | | | | | | | | | | | | |
| DIRECCIÓN | TELÉFONO | FECHA NAC. | E-MAIL | ESTADO CIVIL | COLEGIO | BACHILLERATO | | | | | | | |
| AV. RUMIÑAHUI ENTRE ROCAFUERTE Y MONTALVO PILLARO | 032873018 | 1988-06-01 | magus1211@yahoo.com | SOLTERO | AMBATO | FÍSICO MATEMÁTICO | | | | | | | |
| Resultado Examen de Admisión | | | | | | | | | | | | | |
| PALAB. EXCLU. | USD CONECT. | ELIM. DRAC. | COMP. LECT. | COMP. DRAC. | COHER. REDAC. | SERIES NUM. | ANALISIS FIG. | RAZON. MAT. | RAZON. LOG. | SUFIC. DATOS. | CULT. GENER. | TOTAL | FECHA |
| 6/8 | 2/6 | 6/8 | 6/6 | 4/6 | 4/6 | 4/4 | 2/2 | 2/2 | 0/2 | 0/0 | 0/0 | 36 | 2006-02-10 |
| Encuesta | | | | | | | | | | | | | |
| ORIGEN INFORMACIÓN PUCESA | RADIOEMISORA | SERVICIO MOTIVO INGRESO | COMENTARIO | | | | | | | | | | |
| -FAMILIA-TRÍPTICOS, AFICHES | -DISNEY 94.1 | -HORARIOS | más publicidad , parqueadero abierto para los estudiantes la biblioteca mas libros interesantes | | | | | | | | | | |

Página 1 de 1

¡Se Encontró 1 Registro!

En la parte inferior del reporte encontramos la opción para impresión “Bloques a Imprimir”, en el cual marcamos “” los bloques del reporte que deseamos aparezcan en la versión para imprimir.

| Bloques a Imprimir | |
|-----------------------------|--------------------------|
| Datos Personales: | <input type="checkbox"/> |
| Resultados Examen Admisión: | <input type="checkbox"/> |
| Resultados Encuesta: | <input type="checkbox"/> |

Al hacer “clic” en el botón “Imprimir”, nos enviara a la versión para impresora del Reporte de Registros de Admisión en el cual aparecerán los bloques que hayamos marcado.

ESTADÍSTICAS

Corresponde a las opciones de Reportes Estadísticos de Barras por pantalla e impresión de Registros de Admisión y Encuestas de los Alumnos. Tiene las siguientes opciones:



Estadísticas de Resultados de Examen de Aptitud Académica. Si queremos visualizar el gráfico estadístico de barras de los resultados de los estudiantes en el Examen de aptitud académica por bloques de preguntas, hacemos “clic” en la opción “Aptitud Académica” del menú “ESTADÍSTICAS”.



Esta opción, nos enviará a la página de filtro para estadísticas.

Filtrar:

Escoger Carrera

Carrera:

▼

Fecha Desde:

📅

Fecha Hasta:

📅

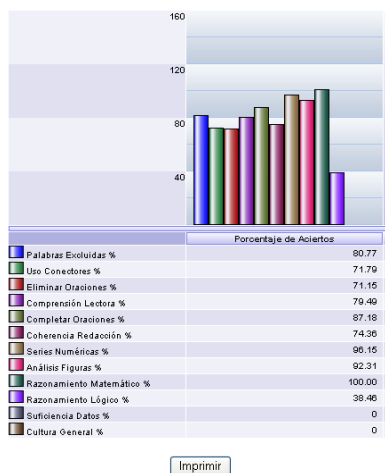
Escoger Fecha

Graficar

Podemos escoger la carrera y el rango de fechas en las que fueron realizados el Examen de Admisión y la Encuesta para filtrar los datos que serán graficados, y presionamos el botón “Graficar”, que nos enviará a la página del gráfico de barras.

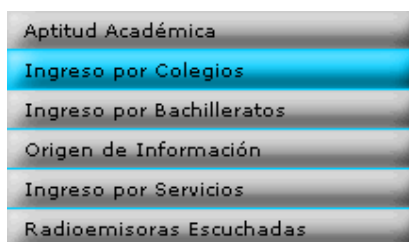
Todos los Alumnos

PORCENTAJE DE ACIERTOS EXAMEN DE ADMISION POR BLOQUES DE PREGUNTAS



Si deseamos imprimir el gráfico, presionamos el botón “Imprimir”.

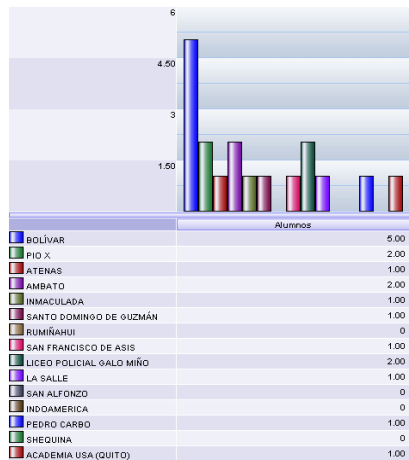
Estadísticas de Ingreso de Estudiantes por Colegio. Si queremos visualizar los gráficos estadísticos de barras del número y porcentaje de alumnos que ingresaron por Colegio, hacemos “clic” en la opción “Ingreso por Colegios” del menú “ESTADISTICAS”.



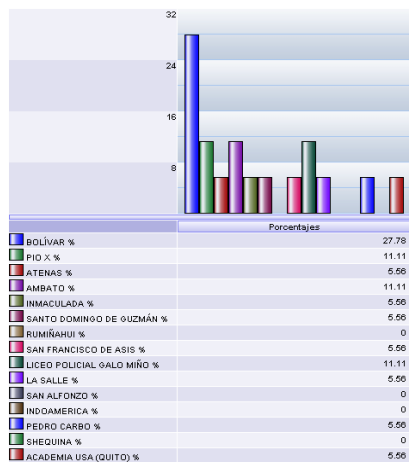
Esta opción, nos enviará a la página de filtro para estadísticas, donde podemos escoger la carrera y el rango de fechas en las que fue realizada la Encuesta para filtrar los datos que serán graficados, y presionamos el botón “Graficar”, que nos enviará a la página del gráfico de barras.

Todos los Alumnos

COLEGIOS POR ALUMNOS DE LA PUCESA (NUMERO)

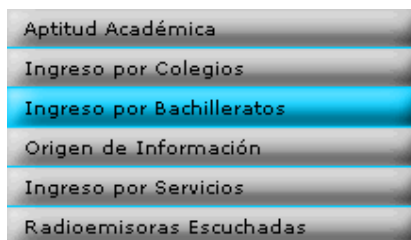


COLEGIOS POR ALUMNOS DE LA PUCESA (PORCENTAJE)



Si deseamos imprimir los gráficos, presionamos el botón “Imprimir”.

Estadísticas de Ingreso de Estudiantes por Bachillerato. Si queremos visualizar los gráficos estadísticos de barras del número y porcentaje de alumnos que ingresaron por Bachillerato, hacemos “clic” en la opción “Ingreso por Bachilleratos” del menú “ESTADISTICAS”.

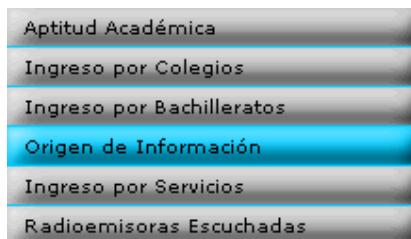


Esta opción, nos enviará a la página de filtro para estadísticas, donde podemos escoger la carrera y el rango de fechas en las que fue realizada la Encuesta para filtrar los datos que serán graficados, y presionamos el botón “Graficar”, que nos enviará a la página del gráfico de barras.

Si deseamos imprimir los gráficos, presionamos el botón “Imprimir”.

Estadísticas de Origen de la Información Obtenida acerca de la Universidad.

Si queremos visualizar los gráficos estadísticos de barras del número y porcentaje sobre orígenes de la información obtenida acerca de la Universidad por parte de los alumnos, hacemos “clic” en la opción “Origen de Información” del menú “ESTADISTICAS”.

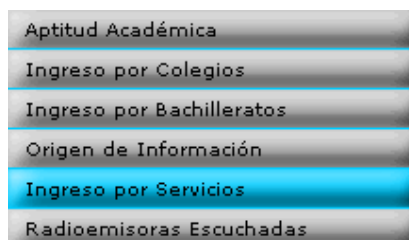


Esta opción, nos enviará a la página de filtro para estadísticas, donde podemos escoger la carrera y el rango de fechas en las que fue realizada la Encuesta para

filtrar los datos que serán graficados, y presionamos el botón “Graficar”, que nos enviará a la página del gráfico de barras.

Si deseamos imprimir los gráficos, presionamos el botón “Imprimir”.

Estadísticas de Servicios Ofrecidos que motivaron el Ingreso de los Estudiantes. Si queremos visualizar los gráficos estadísticos de barras del número y porcentaje de los servicios ofrecidos por la Universidad que motivaron el ingreso de los alumnos, hacemos “clic” en la opción “Ingreso por Servicios” del menú “ESTADISTICAS”.

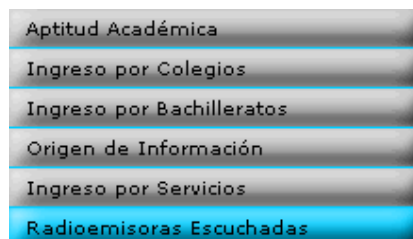


Esta opción, nos enviará a la página de filtro para estadísticas, donde podemos escoger la carrera y el rango de fechas en las que fue realizada la Encuesta para filtrar los datos que serán graficados, y presionamos el botón “Graficar”, que nos enviará a la página del gráfico de barras.

Si deseamos imprimir los gráficos, presionamos el botón “Imprimir”.

Estadísticas de Radioemisoras escuchadas por los Estudiantes. Si queremos visualizar los gráficos estadísticos de barras del número y porcentaje de las

radioemisoras escuchadas por los alumnos, hacemos “clic” en la opción “Radioemisoras Escuchadas” del menú “ESTADISTICAS”.



Esta opción, nos enviará a la página de filtro para estadísticas, donde podemos escoger la carrera y el rango de fechas en las que fue realizada la Encuesta para filtrar los datos que serán graficados, y presionamos el botón “Graficar”, que nos enviará a la página del gráfico de barras.

Si deseamos imprimir los gráficos, presionamos el botón “Imprimir”.





SALIR DEL SISTEMA

Para salir del sistema, hacemos “clic” en la opción “SALIR” del menú principal.



NAVEGADOR DE LISTADOS Y REPORTE

Para desplazarnos dentro de los listados y reportes, hacemos uso del navegador, ubicado en la parte inferior izquierda de la tabla donde se despliegan los datos; su función se describe a continuación:

-  **Primera.** Permite ubicarse en la primera página del formulario.
-  **Anterior.** Permite ubicarse en la página anterior del formulario.
-  **Siguiente.** Permite ubicarse en la página siguiente del formulario.
-  **Última.** Permite ubicarse en la última página del formulario.

Además nos da el detalle del número total de páginas y el número de página actual.

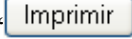
ENLACES RÁPIDOS


Son aquellos enlaces ubicados en la parte inferior de las páginas, sirven para tener rápido acceso a opciones relacionadas a la página en la que el usuario se encuentra.

MENSAJES

Cuando se realizan operaciones de Ingreso, Modificación o Eliminación de registros de la base, luego de realizada la misma, aparecerá por pantalla en caso de existir un error un mensaje en color “Rojo” junto a la imagen de error “

IMPRESIÓN

Las páginas que tienen ésta función, se identifican por el botón “” en la parte inferior de la misma, y son aquellas que se encuentran configuradas para ser impresas.

 **Nota.** Estas páginas cuentan con un control que funciona con Internet Explorer que imprime el documento en la impresora que el usuario tiene establecida como predeterminada.

DESCRIPCION Y USO DE OBJETOS DE FORMULARIOS

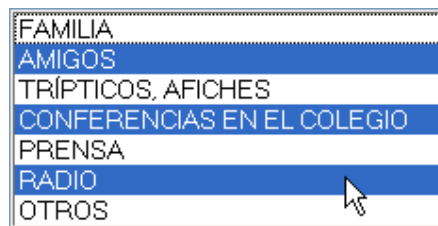
Campo de Texto. Su ingreso se lo realiza a través del uso del teclado una vez ubicado en el mismo.



Campo de Selección. Su ingreso se lo realiza a través del uso del ratón al hacer “clic” en el mismo y seleccionar una de las opciones.





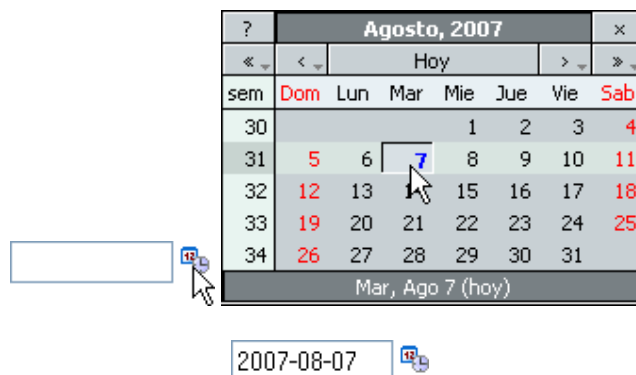
Campo de Selección Múltiple. Su ingreso se lo realiza a través del uso del ratón mientras se presiona la tecla “ctrl” haciendo “clic” una a una para seleccionar más de una opción.



Campo de Verificación. Su ingreso se lo realiza a través del uso del ratón al hacer “clic” en el mismo para seleccionarlo.



Campo de Fecha. Su ingreso se lo realiza a través del uso del ratón al hacer “clic” en el botón “” para activar el calendario y seleccionar una fecha; podemos acceder a la ayuda del uso del calendario al hacer “clic” en el botón “”.



BIBLIOGRAFÍA

Libros:

- Eve Andersson, Philip Greenspun, and Andrew Grumet. Software Engineering for Internet Applications. Massachusetts Institute of Technology, 2006.
- Roger S. Pressman, Ph.D. Software Engineering: A Practitioner's Approach. 5th Edition. McGraw-Hill, 2001.
- David A. Gustafson. Theory and Problems of Software Engineering. McGraw-Hill, 2002.
- Thomas A. Powell. Web Design: The Complete Reference. 2nd Edition. McGraw-Hill/Osborne, 2002.
- Ann Navarro. Effective Web Design. 2nd Edition. Sybex, 2001.
- Gavin Powell. Beginning Database Design. Indianapolis, Indiana: Wiley Publishing, Inc., 2006.
- David Sklar, Adam Trachtenberg. PHP Cookbook. 2nd Edition. O'Reilly, 2006.
- Rasmus Lerdorf, Peter MacIntyre, Kevin Tatroe. Programming PHP. 2nd Edition. O'Reilly, 2006.
- Vikram Vaswani. How to do everithing with PHP and MySQL. McGraw-Hill/Osborne, 2005.

Páginas Web:

- <http://www.angelfire.com/emo/tomaustin/Met/metinacap.htm>
Metodología de Investigación

- <http://www.sunfreeware.com/>
Contenidos para Servidores Sun
- <http://www.monografias.com/trabajos5/inso/inso.shtml>
Ingeniería de Software
- http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software
Ingeniería de Software
- http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol
Protocolo http(Hypertext Transfer Protocol)
- <http://lenguajes-de-programacion.com/programacion-web.shtml>
Programación Web
- <http://www.desarrolloweb.com/>
Artículos, Manuales Desarrollo Web
- <http://www.desarrolloweb.com/manuales/7/>
Manual Páginas Dinámicas
- <http://www.monografias.com/trabajos7/html/html.shtml>
Lenguaje de programación para paginas web HTML
- <http://www.php.net/manual/es/index.php>
Manual Oficial PHP
- <http://www.desarrolloweb.com/manuales/12/>
Programación en PHP
- <http://www.monografias.com/trabajos5/basede/basede.shtml>
Contenidos sobre Base de Datos
- http://www.hotscripts.com/PHP/Scripts_and_Programs/Graphs_and_Charts/index.html
PHP Graphs and Charts tools

- <http://www.mundotutoriales.com>
Tutoriales
- <http://www.solocursos.net/>
Cursos y Tutoriales on-line