UNIVERSIDAD DEL
AZUAY

Facultad de Administración
Escuela de Ingeniería de Sistemas

"Automatización del Registro de Inventario Turístico Caso: Provincia del Azuay"

Monografía propuesta para la obtención del
Título de Ingeniero de Sistemas

Autores:
Jéssica Priscila Herrera Urgilés
Andrea Paulina Tenesaca Quizhpe

Director: Ing. Pablo Esquivel

Cuenca-Ecuador
2007

Indice de Contenidos

Capítulo 4: MANUAL DEL USUARIO

Capítulo 5: CONCLUSIONES Y BIBLIOGRAFÍA

Dedicatoria

Este proyecto lo dedico a la persona mas importante en mi vida, mi hija Natalia, a mis padres por todo su esfuerzo y sacrificio, y a mis hermanos y familiares que en cada momento me han apoyado y alentado.

Jessica Priscila Herrera Urgilés

Dedicatoria

Lo dedico principalmente a mi hija la razón de mi vida y a mi familia que en todo momento me apoyó y compartió sacrificios, esfuerzos y principalmente han sido mi guía en el transcurso de toda mi vida.

Andrea Paulina Tenesaca Quizhpe

Agradecimiento

Nuestros más sinceros agradecimientos a nuestras familias, amigos y profesores que han sabido compartir su compresión y conocimientos para la culminación de este proyecto.

También extendemos un agradecimiento de forma especial al Ing. Pablo Esquivel por su amistad y guía en el desarrollo de este trabajo.

Las opiniones y comentarios vertidos en el presente documento son absoluta responsabilidad de las autoras del mismo.

Jessica Herrera

Andrea Tenesaca

RESUMEN

La página Web de inventario turístico brinda información sobre aquellos lugares que, por sus cualidades naturales y/o culturales, motivan el desarrollo del turismo en el Ecuador. Su intención es ofrecer a la institución y a los estudiantes vinculadas al sector turismo, así como al público en general, una herramienta útil para la planificación turística, mostrando diversas alternativas turísticas y la difusión del Ecuador como un destino turístico único y variado.

Una combinación de herramientas como: MYSQL, PHP y Javascript, está siendo implementado para sustituir el sistema anterior por una aplicación en tiempo real para manipular fácilmente la información conceptual y gráfica a través de la página web de la universidad. El producto final es el resultado de un trabajo conjunto entre los estudiantes del último año de Sistemas y la Facultad de Turismo.

ABSTRACT

The tourist inventory web page presents information about those places that, due to their natural and/or cultural qualities, contribute to the development of tourism in Ecuador. Its intention is to offer the institution, the students related to the tourism field, and the general public, a useful tool for tourist planning, showing the various alternatives that the country has to offer.

A combination of tools such as MYSQL, PHP, and Javascript is being implemented to replaced the former system with a real time application to easily manipulate the conceptual and graphic information through the University of Azuay web page. The final product is the result of a combined work of students in their last year of Systems and Tourism Schools

Introducción

La necesidad de automatizar el almacenamiento de información de un inventario turístico y pasar a una nueva etapa en que los datos recolectados dejan de ser un peso mas en papel para llevarlos e implementarlos como una aplicación Web en la tecnología que nos rodea , ha sido la razón para la realización de este proyecto.

La automatización del registro de inventario Turístico; caso Provincia del Azuay ha surgido de la necesidad de los alumnos de le Escuela de Turismo, debido a que la gestión a realizar con los datos obtenidos de sus visitas se lleva manualmente en fichas de papel sin opciones estándar ni validaciones de la información a ingresar.
Por ello al realizar una aplicación Web programada en PHP, permitirá acceder desde cualquier computador con acceso a Internet, a la actualización de la información obtenida en una base de datos en MySQL, la cual constará con toda la información ingresada por los estudiantes.

Entre las características de esta aplicación web es considerable la facilidad de uso que ofrece, eficacia, rapidez y seguridad de que los datos han sido almacenados correctamente y en tiempo real en la base de datos. Por lo que para poder actualizar los datos se requerirá de su identificación y contraseña, para llevar el registro

adecuado de cada usuario, sea este un estudiante o persona autorizada como administrador, y de toda la información ingresada hasta el momento.

Con la información existente en la base de Datos, se permitirá la administración de la misma a personas debidamente autorizadas.
Entre las metas a alcanzar en el desarrollo de esta monografía está en el fomentar, liderar, y realizar inventarios turísticos que sirvan de base para elaborar productos turísticos en una región, clasificando y evaluando los atractivos para que de esta manera se pueda realizar consultas que permitirán conocer los lugares turísticos visitados por los estudiantes en cada uno de los cantones de las diferentes provincias, principalmente el Azuay. Y de esta manera permitir que la información esté disponible en la página Web de la Universidad del Azuay.

Capítulo 1
INVENTARIO TURÍSTICO

1.1 Introducción

A raíz de la velocidad en la información y el interés cada vez mas marcado de visitar regiones con múltiples atractivos naturales y culturales, los países subdesarrollados son más visitados por los extranjeros de países desarrollados y por sus nacionales, presentándose la necesidad de conocer nuestros propios atractivos turísticos para así mejorar la calidad de los servicios prestados en cada destino.

Se alude al turismo como aquella actividad que implica la utilización temporal de un espacio distinto al de residencia habitual, donde se pretende desarrollar un conjunto de actividades recreativas a partir del uso de unos recursos de base. Existe espacio potencialmente turístico cuando se produce la valoración social de ciertos bienes ambientales. En la medida que los bienes son valorados por distintos segmentos del mercado, se constituyen en atractivos turísticos y conforman el patrimonio turístico de una determinada localización.

En el siguiente capítulo se muestran los esquemas bajo los cuales se basó este proyecto para realizar el inventario turístico, donde se integran los aspectos claves

que califican a un atractivo turístico, con el fin de poder identificar y determinar los sitios turísticos visitados por los estudiantes de nuestra Universidad.
También se describen una serie de conceptos básicos necesarios para el desarrollo del presente proyecto y los pasos que se siguieron para el análisis y validación de los datos obtenidos para así obtener un inventario claro y optimo.

1.2 Definición de inventario Turístico

Un Inventario Turístico se define como una herramienta necesaria en el quehacer turístico al poder aprovechar las potencialidades de una región investigando, analizando y generando información para el desarrollo de proyectos a nivel micro y macro en su localidad. El levantamiento del inventario turístico permitirá, entre otras cosas, identificar las fortalezas y debilidades de los productos y servicios que se ofrecen en la zona.

Para realizar un inventario turístico primero se seleccionan los aspectos clave que califican a un atractivo turístico y se procede a agruparlos según el área a la cual se enfoquen. Por lo tanto un atractivo turístico puede ser analizado y clasificado en función de cinco componentes básicos: Localización, Variables Internas, Variables Externas, Facilidades Turísticas y Características Generales.

1.2.1 Métodos de Información

La fuente de información provino de archivos digitales en los que se llevaba el inventario turístico hasta el momento, dicha información reflejaba falta de organización y agilidad en las búsquedas o reportes que se requerían convirtiéndose de esta manera en una información que no satisfacía los requerimientos de las personas interesadas. Anexo 1

1.3 Análisis de datos necesarios para el registro

Para llevar a cabo el presente proyecto de monografía fue necesario conocer temas y conceptos básicos que se necesitaron para estructurar el modelo de inventario turístico, a continuación se detalla cada uno ellos:

Turismo.- Desde sus orígenes, el término "turismo" ha sido asociado a la acción de "viajar por placer". Aún hoy, muchas personas lo entienden exclusivamente de esta forma sin tener en cuenta sus otras motivaciones y dimensiones.

Para la Organización Mundial del Turismo (OMT), el turismo comprende las actividades que realizan las personas durante sus viajes y estancias en lugares distintos al de su residencia habitual por menos de un año y con fines de ocio, negocios, estudio, entre otros.

El turismo es, en la práctica, una forma particular de emplear el tiempo libre y de buscar recreación, por lo tanto es un fenómeno social que tiene un impacto económico favorable para las comunidades receptoras, y que consiste en el desplazamiento de personas por diversos motivos, desde su punto de residencia fija a otros lugares en donde se constituye en la población flotante de ese lugar, sin participar en los mercados de trabajo y por más de 24 horas pero menos de seis meses.

Turista.- Es toda persona que se desplaza, estimulada por una o varias motivaciones y realiza al menos un pernocte fuera de su residencia permanente, independiente del lapso de tiempo transcurrido, no percibiendo sus ingresos habituales en el lugar visitado.

Atractivo Turístico.- Son todos los recursos turísticos que cuentan con las condiciones necesarias para ser visitados y disfrutados por el turista, es decir, que cuentan con planta turística, medios de transporte, servicios complementarios e infraestructura básica.
Nuevos atractivos para el turismo son los parques nacionales, y los deportes al aire libre, así como el estudio de las tradiciones, leyendas y atractivos turísticos tales como la arquitectura y la antropología serán factores de atracción para el turismo

Servicios Turísticos.- Aquellos servicios que satisfacen las necesidades de los turistas. Los establecimientos que están directamente relacionados con brindar este tipo de atenciones son las empresas de Servicios Turísticos. Las empresas turísticas son sociedades, u organizaciones estructuradas en variedades de comercio, las cuales

tienen como objetivo comercializar personales de servicios que satisfagan las necesidades del turista. (Gráfico 1)

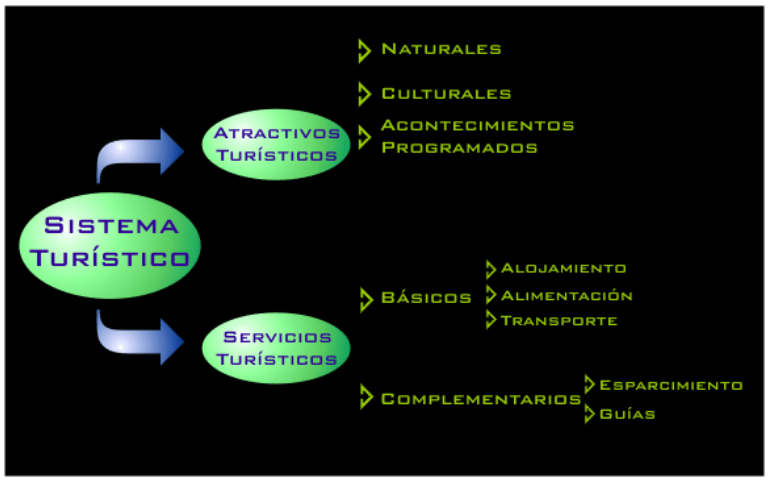


Gráfico 1 Servicios Turísticos

Empresas turísticas de alimentación.

El conjunto de empresas que conforman la Industria Turística responde a los múltiples servicios que deben prestarse a las personas en desplazamiento (turistas). La importancia que tiene las empresas prestatarias de servicios turísticos se debe a la diversidad de funciones que cumplen y a la interrelación entre ellas; ya que su operación debe ser en conjunto para responder a las necesidades del turista.

Infraestructura: Dotación de redes de servicios: de comunicación, de transporte, de telefonía, etc., es decir, desde las rutas hasta los servicios públicos y privados que garanticen que el turista acceda al lugar y tenga una estadía confortable.

Entre estos encontramos:

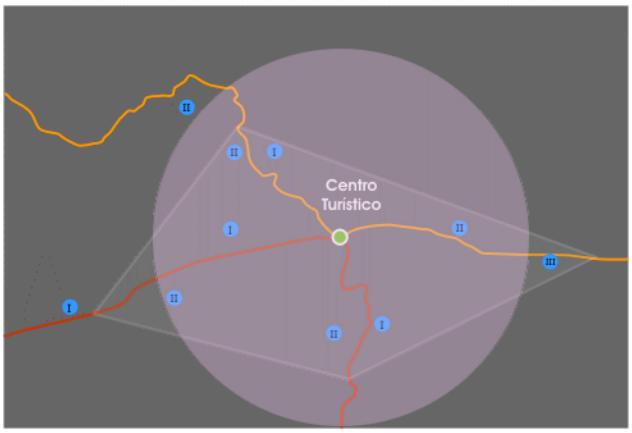
- Rutas de acceso. Pueden ser terrestre, aérea, lacustre, marítima, redes viales, aeropuertos, terminales, etc.
- Comunicaciones. Abarca los servicios de teléfono, correo, telégrafo, internet, correo electrónico, etc.
- Equipamiento urbano. Comprende los servicios agua, desagüe, alcantarillado y energía eléctrica.

Accesibilidad: Es importante que los accesos sean transitables en todo el año o la mayor parte del mismo, que no requieran mantenimiento periódico (en el caso de que se trate de caminos consolidados o de tierra) y en lo posible a corta distancia de las rutas principales asfaltadas.

La jerarquización: La capacidad de atracción no es siempre la misma en todos los atractivos turísticos, por ello pueden ser jerarquizados. Para ello, la capacidad de atracción se cuantifica y mide atendiendo al impacto que puedan generar en el turismo internacional o en el interno, este proceso es conocido como jerarquización.

La valoración social da lugar a la jerarquización de los atractivos en función de su capacidad para atraer mercados, de manera que cuanto más lejanos sean los mercados que acuden a visitarlo mayor será su jerarquía. Adoptando la jerarquización realizada por CICATUR/OEA(1983), los atractivos turísticos, materia prima del sector, se clasifican en cuatro jerarquías:

- **Atractivos Jerarquía IV**: Excepcionales capaces por sí solos de atraer una corriente importante de visitantes actuales o potenciales del mercado internacional.
- . **Atractivos Jerarquía III:** Excepcionales capaces por sí solos de atraer una corriente del mercado interno y en menor porcentaje que los atractivos Jerarquía IV del turismo receptivo.
- **Atractivos Jerarquía II:** Atractivos con algún rasgo llamativo, capaces de interesar a visitantes, ya sea del mercado interno o receptivo que hubiesen llegado a la zona por otras motivaciones turísticas
- **Atractivos Jerarquía I:** Atractivos sin méritos suficientes para considerarlos en las jerarquías anteriores. Pero que forman parte del patrimonio turístico como complemento de otros atractivos de mayor jerarquía.



Cercanía a un Centro Urbano de apoyo: Siempre es necesario el auxilio de servicios complementarios que se consumen en el lugar de prestación de servicios como la alimentación, guías locales, eventuales servicios médicos y expendio de combustibles entre otros. Es importante considerar que en ocasiones algunos de estos servicios son adquiridos o comprados en y desde el centro turístico y es aquí donde las localidades no turísticas cercanas a los atractivos deben insistir y competir. Una de las claves del desarrollo sustentable es que las poblaciones locales se beneficien con la explotación de los recursos ubicados en sus cercanías, propiciando de este modo su preservación.

La planta turística

Es el conjunto de instalaciones, equipos, empresas y personas que prestan servicio al turismo y fueron creados para este fin. Estos se clasifican en:

1. Alojamiento. Según la OMT, éste se divide en hoteleros y extrahoteleros. Los primeros están conformados por:

- Hoteles que ofrecen alojamiento con o sin servicios complementarios (alimentación, congresos y eventos). Estos dependen de la categoría de cada establecimiento que se puede clasificar de 5 a 1 estrellas, según la reglamentación vigente en nuestro país, siendo el de 5 estrellas el que brinda mayores y mejores servicios complementarios.
- Hoteles-apartamento que pueden ofrecer todos los servicios de los hoteles, pero que cuentan adicionalmente con instalaciones y equipamiento para la conservación, preparación y consumo de alimentos fríos y calientes. Van de 5 a 3 estrellas[3].
- Moteles: Se encuentran ubicados en las carreteras o autopistas, poseen entrada independiente desde el exterior al alojamiento y pueden tener garaje individual o parqueo colectivo. En el Perú aún no se cuenta con esta clasificación.
- Hostal o pensión: Cumplen las funciones de un hotel, pero no alcanzan las condiciones mínimas indispensables para considerárseles como tales. Puede ser de 3 a 1 estrellas.

Los extrahoteleros agrupan a:

  - Campamentos que son terrenos de uso privado en los que se instalan sistemas de alojamiento, tales como tiendas de campañas o caravanas móviles. Cuentan con instalaciones comunes (sanitarios, lugares para el aseo, el lavado y el tendido de ropa, zonas recreativas).
  - Casa de alojamiento, normalmente particulares, que prestan servicios en época de alta demanda turística y que contribuyen a ampliar la oferta de un determinado destino.
  - Casas rurales, comprende las habitaciones de las comunidades campesinas que han sido adaptadas especialmente para recibir a turistas.
  - Apartamentos que han sido adaptados para permanencias largas.

Restaurantes.- Son aquellos establecimientos que expenden comidas y bebidas preparadas al público en el mismo local, prestando el servicio en las condiciones

señaladas en el reglamento de restaurantes y de acuerdo a las normas sanitarias correspondientes.

Capacidad de Carga*:* El primer paso es analizar el número óptimo de visitantes simultáneos que puede recibir el atractivo. Con este fin será necesario calcular la capacidad receptiva del mismo. Según Boullón(1994) la capacidad receptiva de un ambiente natural específico se puede calcular mediante estándares obtenidos por la combinación de tres tipos de capacidad: la material, la psicológica y la ecológica.

1.4 Análisis para la validación de datos

Se procedió a realizar un análisis de cada uno de los datos con el objetivo de asignar prioridades a cada uno de ellos como obligatoriedad, valor especifico, opcional, etc., de acuerdo al estudio de la información necesaria para llevar a cabo el inventario, dando de esta manera facilidad, agilidad y sobretodo siendo una guía para el alumno al momento de ingresar los datos en la ficha evitando así información repetida, errónea o inconsistente.

Código: automático
Nombre, Descripción: obligatorio en todas las tablas
Observación, Causa, Recomendación: opcional, campo destinado a comentarios u observaciones.
Mapa: campo en el cual se ingresará un nombre de un archivo de imagen.
Latitud: campo destinado al ingreso de un número compuesto, obligatorio.
Longitud: campo destinado al ingreso de un número compuesto, obligatorio.
Altitud: campo destinado al ingreso de un número compuesto, obligatorio.
Temperatura (máxima y mínima): campo destinado al ingreso de un número compuesto.
Capacidad: campo destinado al ingreso de un número compuesto que indica la capacidad de carga.
RangoInf: Valor numérico más bajo permitido, obligatorio.
RangoSup: Valor numérico más alto permitido, obligatorio.
Época de Sol y E. de Lluvia: Meses que comprende una determinada época, obligatorio.

Esparcimiento: Campo de selección opcional.

Transporte Público y Privado: Campo de selección opcional.

Habitaciones, Plazas, Mesas, Pisos: Valor numérico que indica el número de ocupaciones de un determinado atractivo.

Tipo: Clasificación de un determinado aspecto.

Km: valor numérico que indica una distancia aproximada.

1.5 Estructuración de los datos

Se realizó una valoración de los atractivos turísticos clasificando sus características y estructurándolas de la siguiente manera:

1.6 Conclusiones

Al implementar el Inventario Turístico Digital de la Provincia del Azuay se cuenta con una herramienta fundamental para la planificación del desarrollo turístico de Cuenca y la región, ya que se podrá obtener información válida y eficaz en el momento en que se lo requiera, de esta manera los estudiantes de la escuela de turismo agilitarán sus actividades y podrán utilizar dicha información en proyectos futuros que lleven al desarrollo y modernización de la ciudad y el País.

Capítulo 2
MySQL

2.1 Introducción

Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su ínter actuación con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

2.2 Análisis del modelo Entidad_Relación

El análisis del modelo Entidad_Relación se basa en un estudio previo de las tablas necesarias en la base de datos del sistema y su relación entre sí, con el fin de cumplir con todos los requerimientos planteados por parte de los usuarios. Como se puede observar en el Anexo 2.

2.3 Diseño del modelo Entidad - Relación

El propósito de esta sección es preparar las especificaciones técnicas de diseño para la base de datos, la misma que será adaptable a futuros requerimientos y expansiones. Los campos, tablas y los tipos de datos contenidos en ellas han sido concebidas de acuerdo al análisis y estudio explicado anteriormente. La base de datos para este sistema se denomina **bdituristico** desarrollada en MySQL, la cual esta formada por 43 tablas propiamente en la base y 38 tablas en el modelo. Ver Anexo 3.

2.4 Generalidades de MySql

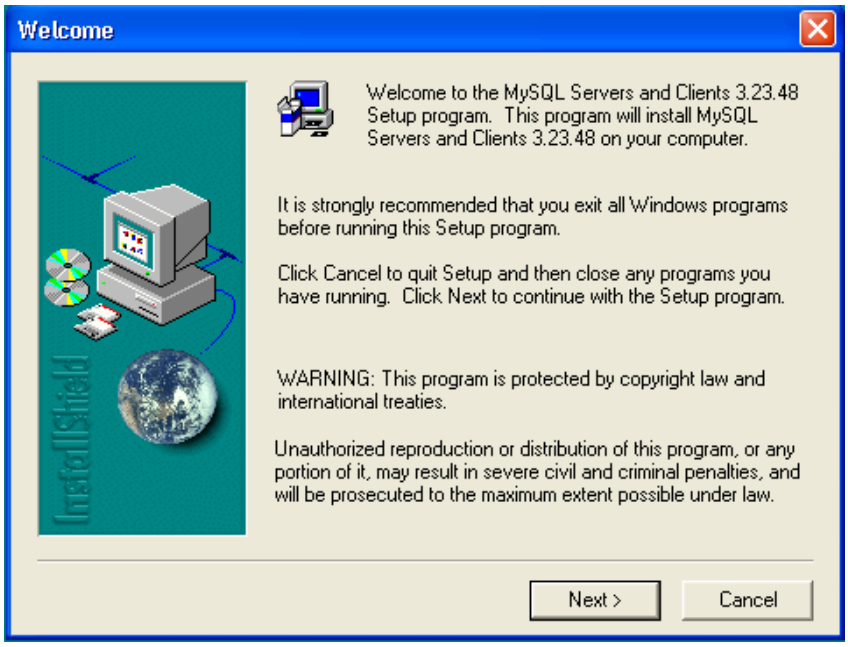
MySQL es la base de datos open source más popular y, posiblemente, mejor del mundo. Su continuo desarrollo y su creciente popularidad está haciendo de MySQL un competidor cada vez más directo de gigantes en la materia de las bases de datos como Oracle.
MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados bases de datos.

También es muy destacable, la condición de open source de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.
Para poder manejar MySql debemos tener unos conocimientos mínimos del lenguaje SQL.

2.5 Instalación y configuración por primera vez en windows

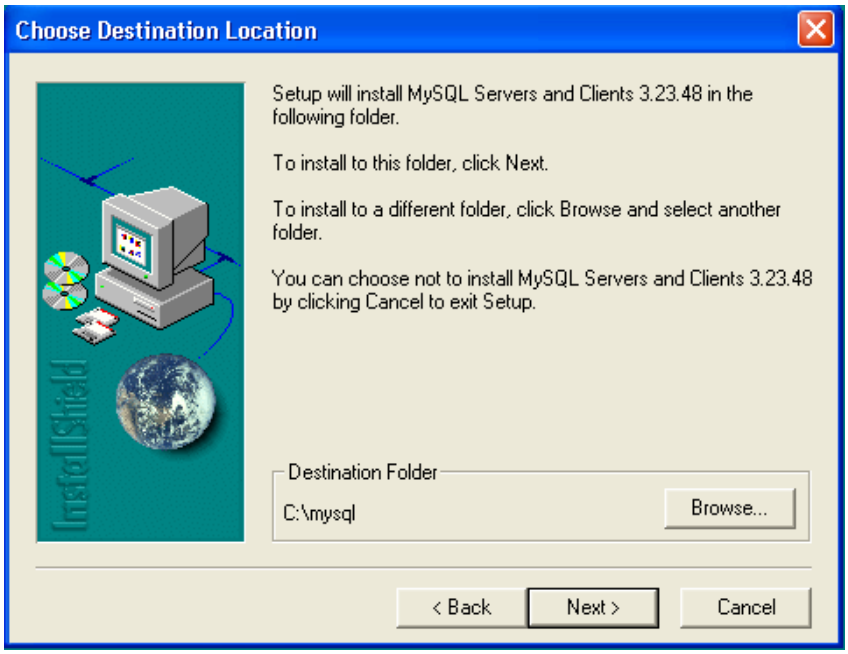
Una vez que tengamos el archivo de instalación mysql- 3_23_34-win.zip debemos descomprimirlo, por ejemplo, podemos ubicarlo en el directorio "C:\mysqlinst\". Una vez descomprimido vamos a este directorio y ejecutamos el programa de instalación "Setup.exe" donde indicaremos las rutas que queremos para su instalación. Como se muestra a continuación:

La primera pantalla a mostrarse es la siguiente con la cual se inicia la instalacion del software, donde se tendrá que hacer click en siguiente:

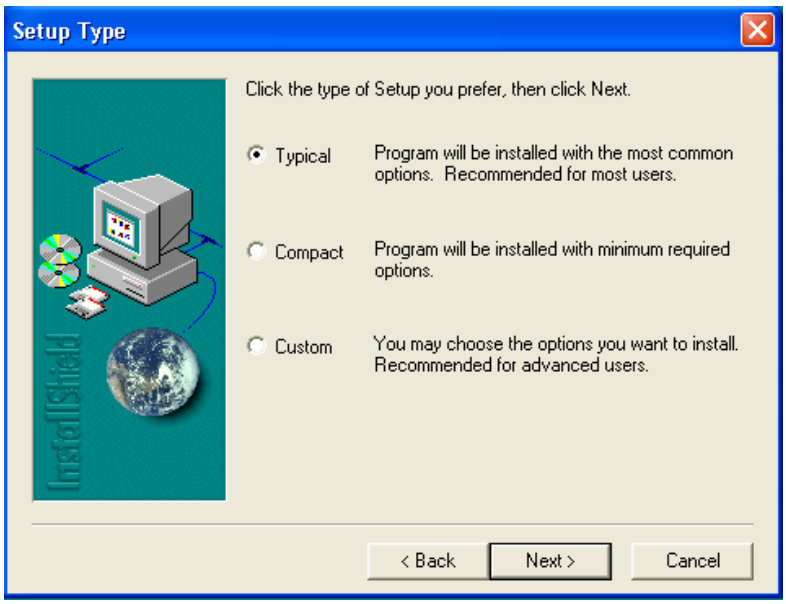


La siguiente pantalla nos muestra mayor información sobre el software que estamos instalando, las rutas que utiliza en su instalación y otras opciones de configuración.

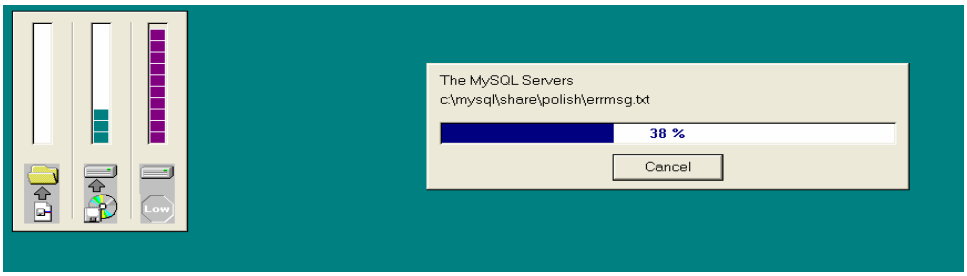
Aquí tendremos que escoger el directorio en el cual deseamos que se instale nuestro software.



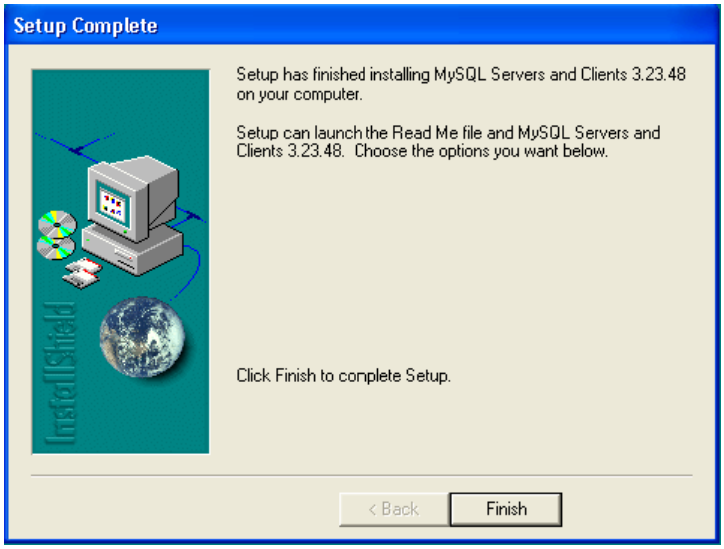
A continuación se pedirá escoger el tipo de instalación que deseamos realizar según nuestras necesidades y preferencias.



Luego de haber escogido el tipo de instalación, se procesará hasta el 100% .



Al finalizar se mostrará la pantalla de finalización de la instalación.



Una vez terminada la instalación los archivos ejecutables se encuentra en el directorio "\bin\". MySQL se comporta como un servidor, por tanto para poder tener acceso al SGBD (sistema gestor de bases de datos) deberemos tenerlo en funcionamiento, para hacer esto deberemos ejecutar el archivo "\bin\mysqld.exe". Para acceder al sistema debemos hacerlo con el programa cliente, que es "/bin/mysql.exe". Luego ejecutamos la aplicación c:\mysql\bin\winmysqladmin.exe

Esta aplicación nos permitirá administrar la base de datos. La primera vez que se ejecuta nos va a pedir un usuario y una clave en este caso vamos a poner root y admin. En la parte inferior derecha de la pantalla aparecerá un semáforo con la luz verde encendida esto indica que el servidor está activo, para apagar el servidor damos un click con el botón derecho sobre el gráfico del semáforo y aparecerá un menú, elegimos la opción shutdown Both. Esta herramienta también sirve para crear bases de datos en modo gráfico.

2.6 Conexión 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:

```
shell> 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:

```
shell>mysql -h turistico -u root -p
Enter password: ******

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5563 to server version: 3.23.41

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Este prompt nos indica que mysql está listo para recibir comandos.

Algunas instalaciones permiten que los usuarios se conecten de manera anónima al servidor corriendo en la máquina local. Si es el caso de nuestra máquina, debemos de ser capaces de conectarnos al servidor invocando a mysql sin ninguna opción:

```
shell> mysql
```

Después de que nos hemos conectado de manera satisfactoria, podemos desconectarnos en cualquier momento al escribir "quit", "exit", o presionar CONTROL+D.

Para ingresar al modo de comandos y administrar la base de datos se va al directorio:

C:\mysql\bin\mysql –u root –p

Como password se pone admin, esto hará que salga el prompt de mysql:
mysql>

Podemos ver las bases de datos que tenemos en el sistema escribiendo lo siguiente:
> show databases;

Crear una base de datos
En el prompt de mysql:
Mysql> USE bdituristico
ERROR 1049: Unknown database ' bdituristico '
mysql>
El mensaje anterior indica que la base de datos no ha sido creada, por lo tanto necesitamos crearla.
mysql> CREATE DATABASE bdituristico;
Query OK, 1 row affected (0.00 sec)
mysql> USE bdituristico
Database changed
mysql>
Bajo el sistema operativo Unix, los nombres de las bases de datos son sensibles al uso de mayúsculas y minúsculas (no como las palabras clave de SQL), por lo tanto debemos tener cuidado de escribir correctamente el nombre de la base de datos. Esto es cierto también para los nombres de las tablas.
Al crear una base de datos no se selecciona ésta de manera autómatica; debemos hacerlo de manera explícita, por ello usamos el comando USE en el ejemplo anterior.

La base de datos se crea sólo una vez, pero nosotros debemos seleccionarla cada vez que iniciamos una sesión con mysql. Por ello es recomendable que se indique la base de datos sobre la que vamos a trabajar al momento de invocar al monitor de MySQL. Por ejemplo:

```
shell>mysql -h turistico -u root -p bdituristico
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17 to server version: 3.23.38-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer
```

mysql>Observar que " bdituristico " no es la contraseña que se está proporcionando desde la línea de comandos, sino el nombre de la base de datos a la que deseamos acceder. Si deseamos proporcionar la contraseña en la línea de comandos después de la opción "-p", debemos hacerlo sin dejar espacios (por ejemplo, -padmin, no como -p admin). Sin embargo, escribir nuestra contraseña desde la línea de comandos no es recomendado, ya que es bastante inseguro. Usar una base de datos.

Conociendo como escribir y ejecutar sentencias, se puede de acceder a una base de datos.

Primeramente usaremos la sentencia SHOW para ver cuáles son las bases de datos existentes en el servidor al que estamos conectados:

```
mysql> SHOW DATABASES;
+----------+
| Database |
+----------+
| mysql    |
| test     |
+----------+
2 rows in set (0.00 sec)
mysql>
```

Es probable que la lista de bases de datos que veamos sea diferente en nuestro caso, pero seguramente las bases de datos "mysql" y "test" estarán entre ellas. En particular, la base de datos "mysql" es requerida, ya que ésta tiene la información de los privilegios de los usuarios de MySQL. La base de datos "test" es creada durante

la instalación de MySQL con el propósito de servir como área de trabajo para los usuarios que inician en el aprendizaje de MySQL.

Cabe anotar también que es posible que no veamos todas las bases de datos si no tenemos el privilegio SHOW DATABASES.

Si la base de datos "test" existe, hay que intentar accesar a ella:

```
mysql> USE test
Database changed
mysql>
```

Observar que USE, al igual que QUIT, no requieren el uso del punto y coma, aunque si se usa éste, no hay ningún problema. El comando USE es especial también de otra manera: éste debe ser usado en una sola línea.

Sintaxis MySQL

Una vez conectados ya al servidor MySQL, aún cuando no hemos seleccionado alguna base de datos para trabajar. Lo que haremos a continuación es escribir algunos comandos para conocer el funcionamiento de mysql.

```
mysql> SELECT VERSION(), CURRENT_DATE;
mysql>
```

Este comando ilustra distintas cosas acerca de 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 (como en el ejemplo anterior) las etiquetas en las columnas son la expresión en sí.

MySQL muestra cuántas filas fueron regresadas y cuanto tiempo tardó en ejecutarse la consulta, lo cual puede darnos una idea de la eficiencia del servidor, aunque estos valores pueden ser un tanto imprecisos ya que no se muestra la hora del CPU, y

porque pueden verse afectados por otros factores, tales como la carga del servidor y la velocidad de comunicación en una red.

Las palabras clave pueden ser escritas usando mayúsculas y minúsculas.

Las siguientes consultas son equivalentes:

```
mysql> SELECT VERSION(), CURRENT_DATE;
mysql> select version(), current_date;
mysql> Select version(), current_DATE;
```

2.6.1 Creación de tablas

Crear la base de datos es la parte principal, pero en este momento la base de datos está vacía, como lo indica el comando SHOW TABLES:

```
mysql> SHOW TABLES;
Empty set (0.00 sec)
```

Usaremos la sentencia CREATE TABLE para indicar como estarán conformados los registros de nuestra información.

```
mysql> CREATE TABLE nombre_tabla(
    -> nombre_campo TIPO_DE_DATO(20));
Query OK, 0 rows affected (0.02 sec)
mysql>
```

Ahora que hemos creado la tabla, la sentencia SHOW TABLES debe producir algo como:

```
mysql> SHOW TABLES;
1 row in set (0.00 sec)
mysql>
```

Para verificar que la tabla fue creada, usaremos la sentencia DESCRIBE:

```
mysql> DESCRIBE nombre_tabla;
6 rows in set (0.01 sec)
```

Podemos observar la creación de cada una de las tablas necesarias para nuestra base de datos en el Anexo 4.

2.6.1.1 Recuperar información de una tabla

La sentencia SELECT es usada para obtener la información guardada en una tabla. La forma general de esta sentencia es:

SELECT LaInformaciónQueDeseamos FROM DeQueTabla WHERE CondiciónASatisfacer

Aquí, LaInformaciónQueDeseamos es la información que queremos ver. Esta puede ser una lista de columnas, o un * para indicar "todas las columnas".
DeQueTabla indica el nombre de la tabla de la cual vamos a obtener los datos. La cláusula WHERE es opcional. Si está presente, la CondiciónASatisfacer especifica las condiciones que los registros deben satisfacer para que puedan ser mostrados.

2.6.1.2 Seleccionando todos los datos

La manera más simple de la sentencia SELECT es cuando se recuperan todos los datos de una tabla:

mysql> SELECT * FROM tabla;

Esta forma del SELECT es útil si deseamos ver los datos completos de la tabla, por ejemplo, para asegurarnos de que están todos los registros después de la carga de un archivo.

A modo de resumen, proponemos además las operaciones más básicas :

Instrucción	Descripción
Show databases;	Muestra el conjunto de bases de datos presentes en el servidor
Use nombre_de_la_base	Determina la base de datos sobre la que vamos a trabajar
Create Database	Crea una nueva bd con el nombre especificado

nombre_de_la_base;	
Drop Database nombre_de_la_base;	Elimina la base de datos del nombre especificado
Show tables;	Muestra las tablas presentes en la base de datos actual
Describe nombre_de_la_tabla;	Describe los campos que componen la tabla
Drop Table nombre_de_la_tabla;	Borra la tabla de la base de datos
Load Data Local Infile "archivo.txt" Into Table nombre_de_la_tabla;	Crea los registros de la tabla a partir de un fichero de texto en el que separamos por tabulaciones todos los campos de un mismo registro.
Quit	Salir de MySQL

2.7 DBDesigner 4

Es un Sistema totalmente visual de diseño de bases de datos, que combina características y funciones profesionales con un diseño simple, muy claro y fácil de usar, a fin de ofrecer un método efectivo para gestionar las bases de datos.

DBDesigner permite administrar la base, diseñar tablas, hacer peticiones SQL manuales y mucho más, como ingeniería inversa en MySQL, Oracle, MSSQL y otras bases de datos ODBC, modelos XML y soporte para la función drag-and-drop. Además, cuenta con una interfaz profesional y detallados manuales de uso.
Se trata de un completo sistema visual para el desarrollo de aplicaciones de bases de datos que integra diseño, modelado, creación y mantenimiento de fácil uso y optimizado para MySQL.

Características:

- Está disponible para Linux y Ms Windows.
- La presentación al usuario es análoga a la que presenta otro software:
- Tiene un navegador del diseño igual al que tiene Adobe Illustrator® and photoshop®.

- Paletas de herramientas fijas o flotantes.
- Objetos: tablas, relaciones, etiquetas, áreas e imágenes.
- Permite usar "drag and drop".
- Menús "Pop-up".
- Editores avanzados.
- Funcionalidad ilimitada hacer/deshacer.
- Funciones "copy / cut / paste" al/del portapapeles (XML, DLL).
- Funciones para alinear elementos.
- Modo diseño y modo consulta.
- Posibilidad de realizar ingeniería inversa con bases de datos de MySQL, Oracle, MSSQL o cualquier base de datos ODBC.
- Generación del esquema de la base de datos definida por el usuario.
- Sincronización del modelo con la base de datos.
- Soporta índices.
- Sustitución automática de "foreign keys".
- Elaboracion de documentación.
- Impresión del modelo según varios formatos (incluye gráfica y XML).
- Soporta todos los tipos de campos de MySQL.
- También soporta tipos de campos definidos por el usuario.
- Control de versiones.
- Una consola para construir sentencias de SQL.
- Un histórico de los comandos SQL.
- Almacenamiento de los comandos SQL con el modelo.

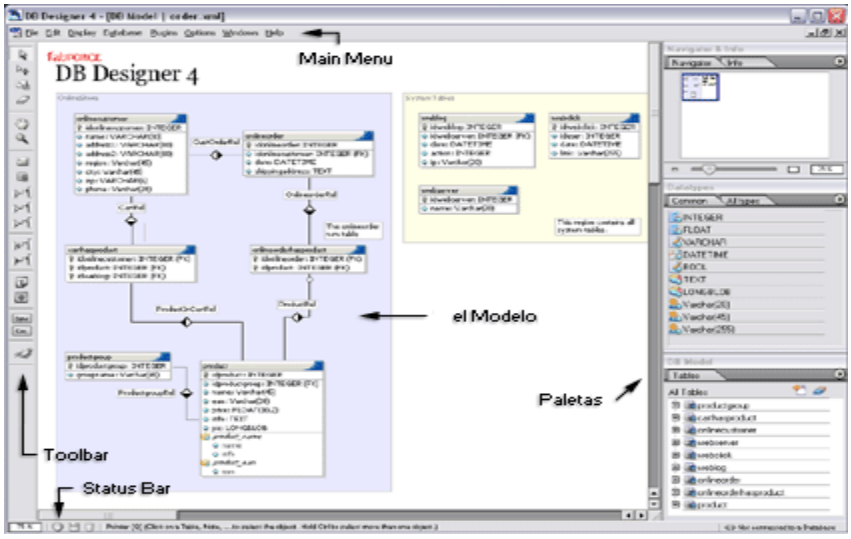
Fundamentos : La clave del uso de DBDesigner 4 es entender los fundamentos de bases de datos.

Modelos y Bases de Datos : En DBDesigner 4, siempre trabajas en un modelo. Un modelo es una visualización de la meta-información almacenada en una base de datos (e.g. Tablas e Índices, Relaciones, ...) Aunque es posible guardar datos iniciales en las tablas directamente en el modelo, sólo se representa la meta-información, no los datos en si mismos.

Puedes crear y mantener tantos modelos como necesites, conteniendo un número ilimitado de objetos. Un objeto puede ser una tabla de base de datos con columnas e índices, una relación entre dos tablas, una nota, ...
Los modelos pueden ser diseñados colocando estos objetos en el lienzo de modelo o pueden ser extraídos de una base de datos existente usando la función de ingeniería inversa.

Para crear la base de datos el modelo puede ser exportado como un Script SQL de Creates o ser creado directamente desde DBDesigner 4 usando la función de sincronización. La función de sincronización también se usa para modificar la base de datos automáticamente cuando cambia el modelo.
Cuando DBDesigner cambia al Modo Consulta el modelo puede ser usado para construir complejas consultas SQL y editar los datos de las tablas.
Los modelos se guardan como fichero XML o pueden ser almacenados directamente en la base de datos activando el acceso distribuido al modelo.

La Interfaz de Usuario: La interfaz de usuario se basa en estándares de software de diseño. Esto hace que crear tus modelos de base de datos sean muy sencillos.

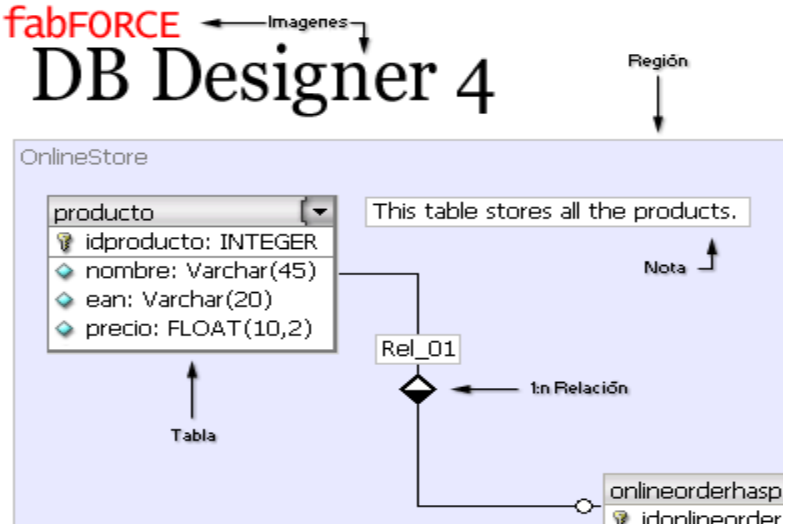


GUI de DBDesigner

A parte de los familiares menús de ventana, barras de desplazamiento y barras de estado DBDesigner 4 provee varias paletas (ventanas flotantes) que pueden ser usadas para acceder a funciones más rapidamente.

DBDesigner 4 tiene soporte para el Interfaz de Múltiples Documentos (MDI) que te permite abrir un número ilimitado de modelos al mismo tiempo. Puedes moverte entre modelos y usar comandos estándar de copiar y pegar para intercambiar objetos entre los modelos.

Modelado: El modelo de base de datos se crea colocando diversos objetos en el lienzo, especificando sus atributos y relaciones.
Para colocar un objeto se ha de seleccionar la herramienta adecuada de la Barra de Herramientas. Además de los objetos más importantes como Tablas y Relaciones (1:1, 1:1 generalización, 1:n, 1:n no identificadora, n:m) puedes usar Notas, Imagenes y Regiones para ayudar a "entender" la estructura del modelo.



Los Objetos

Tablas: Las Tablas representan tablas de bases de datos. La Tabla se muestra de forma similar a una ventana. El nombre de la Tabla se indica en el título, las columnas se muestran debajo y están indicadas con un icono. Un icono en forma de llave indica que la tabla es una clave primaria.

Clave Primaria: Normalmente una o más columnas están definidas como la Clave Primaria (PK) de la tabla. Estas columnas no pueden contener dos o más valores iguales. Esto hace posible identificar claramente cada registro en la tabla mediante la Clave Primaria (e.g. idproducto)

Índices: Para hacer la base de datos encontrar un registro específico más rápidamente, es posible definir en un índice en una o más columnas. Los índices también se emplean para mejorar la velocidad cuando se realiza un JOIN entre una o más tablas.
Relaciones: Las relaciones pueden realizarse únicamente entre dos tablas. Definen la relación entre tablas y pueden crear referencias de Claves Foráneas. Las tablas pueden conectarse con relaciones uno-a-uno (e.g. persona-dirección), uno-a-muchos (grupodeproducto - producto) o muchos-a-muchos (e.g. empleado - reunión).

Notas: Las notas son simplemente cajas de texto que contienen información acerca de una tabla o estructura. Pueden colocarse en cualquier lugar en el modelo para proveer la información que se necesite.
Imágenes: Las imágenes pueden colocarse en el modelo para visualizar información adicional.
Regiones: Las regiones proveen espacio para tablas con los mismos atributos. Los atributos se asignan a la región y se aplican automáticamente a todas las tablas de esa región. Además pueden ser usadas por los plugins para aplicar funciones específicas a un grupo de tablas.

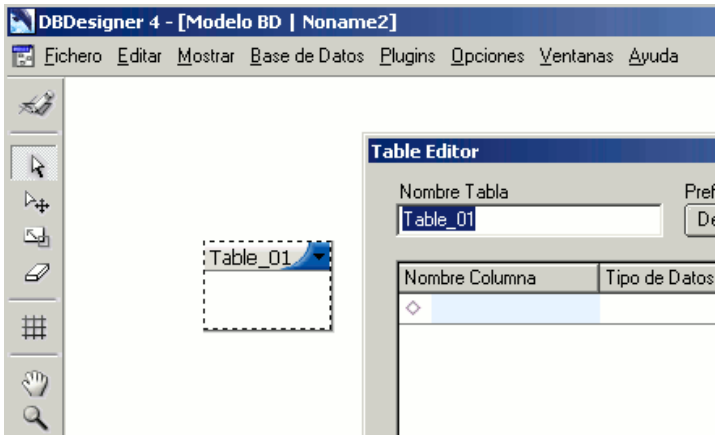
Ejecución del Modelado: Antes de empezar a modelar la base de datos el Modo Diseño debe seleccionarse. Para cambiar el Modo de Trabajo actual pulsa el icono de Modo de Trabajo en la Paleta de Herramientas.

Colocar una nueva tabla.- Para colocar una nueva tabla en el modelo selecciona la Herramienta de Tabla de la Barra de Herramientas o pulsa [T] en el teclado. El cursor del ratón cambiará para reflejar la herramienta seleccionada actualmente.

Seleccionando la Herramienta de Tabla

Para colocar la tabla en el modelo pulsa el botón izquierdo. Se creará una nueva tabla. Se llamará [table_XX]. La esquina superior izquierda estará en la posición donde se pulsó con el ratón. Después de que la tabla se coloque la herramienta seleccionada cambiará de nuevo a la Herramienta Puntero. El cursor también cambiará.

Editar una tabla: Para editar una tabla existente hay que asegurarse que la Herramienta Puntero esté seleccionada. Haciendo una doble pulsación en la tabla con el botón izquierdo del ratón. Se mostrará el Editor de Tablas.



Llamando al editor de tablas

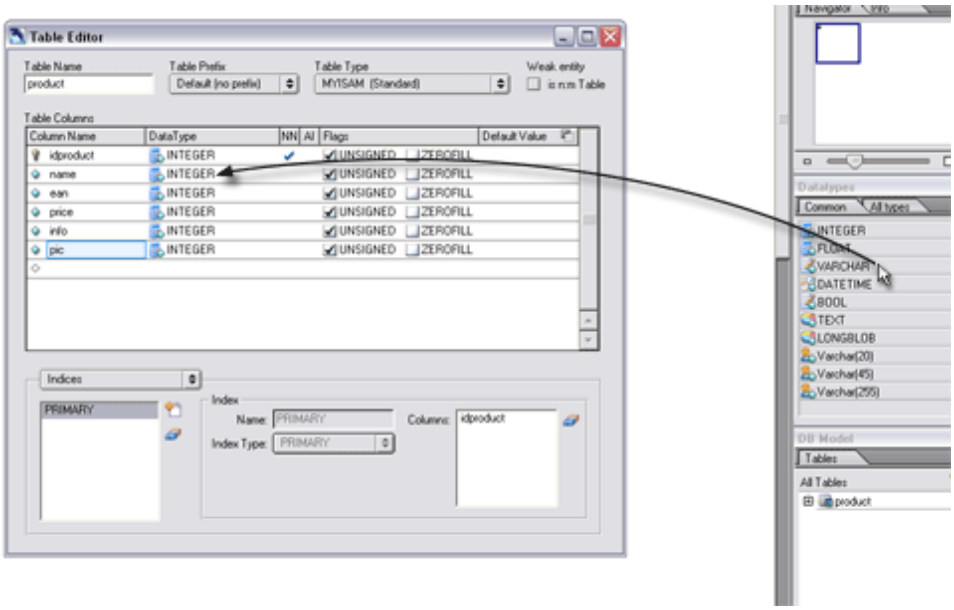
Definiendo nombres y columnas de tablas: Ingresar el nombre de la tabla (p.e ITParametros) y se pulsa la tecla enter. El foco cambiará a la primera columna. Se llamará id+nombre_de_tabla por defecto. Para aceptar este nombre se pulsa enter o sobreescribirlo antes de pulsar enter.

Pulsa la tecla Esc después de asignar nombre a la última columna. Las columnas pueden ordenarse arrastrando y soltando.

Asignación de tipos de datos de las columnas

Asignando tipos de datos

Todas las columnas se crearán con el tipo por defecto.



Asignación de tipos de datos

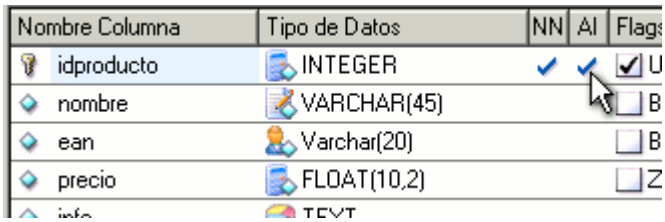
Para cambiar el tipo de datos de una columna ir a l a Paleta de Tipos de Datos y arrastrar el tipo apropiado en la columna en el Editor de Tablas. Si el tipo asignado tiene parámetros (p.e. VARCHAR(xxx)) entra el valor deseado.

Para cambiar el parámetro de un tipo haz una doble pulsación en el tipo de datos y se ingresa el nuevo valor.

Asignando los parámetros del tipo de datos

Clave Primaria, AuntoInc y Opciones

La primera columna se define automáticamente como clave primaria de la tabla indicada con un Icono de Llave a la izquierda del nombre de columna. Para eliminar una columna de la clave primaria se pulsa el Icono de la Llave.

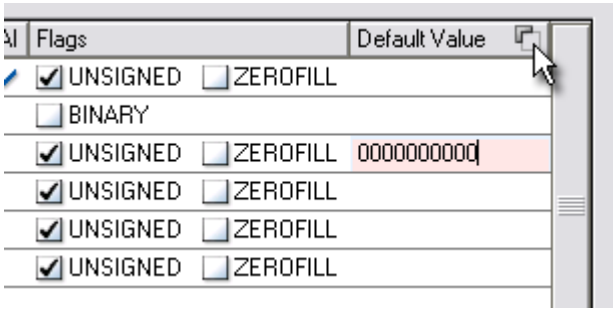


Asignando la propiedad de Auto Incremento

Para hacer una columna autoincrementable tras la inserción pulsa la columna AI de su fila. Cada tipo de datos tiene opciones específicas. Pueden activarse y desactivarse pulsándolas.

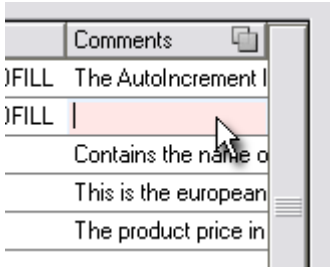
Valores por defecto y Comentarios

Para asignar un valor por defecto a una columna se hace una doble pulsación en la columna Valor por Defecto y entra el valor. Pulsar Enter para aplicar los cambios.



Asignando Valores por Defecto

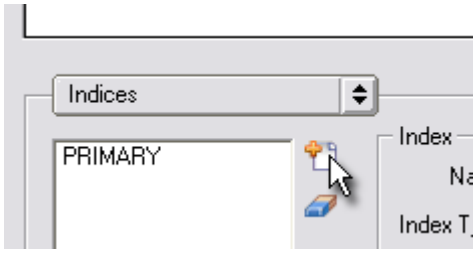
Para mostrar los Comentarios de la columna pulsar el icono a la derecha del título de la columna Valor por Defecto. Para editar un comentario haz una doble pulsación en la columna Comentarios.



Editar los comentarios de columna

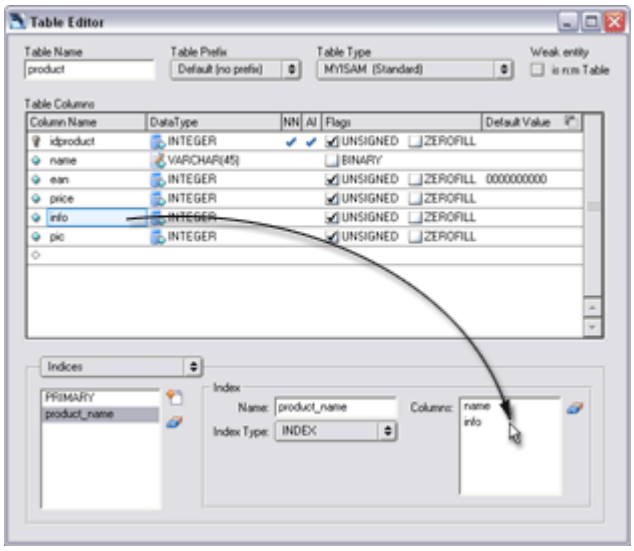
Crear Índices de Tabla

Cambiar a la pestaña de Índices y pulsar el Icono Más para añadir un nuevo índice. Aparecerá un diálogo de Introducción de Cadenas. Se debe ingresar el nombre del índice y pulsar enter. Se mostrará el nuevo índice. Seleccionar el tipo de índice.



Crear un nuevo índice de tabla

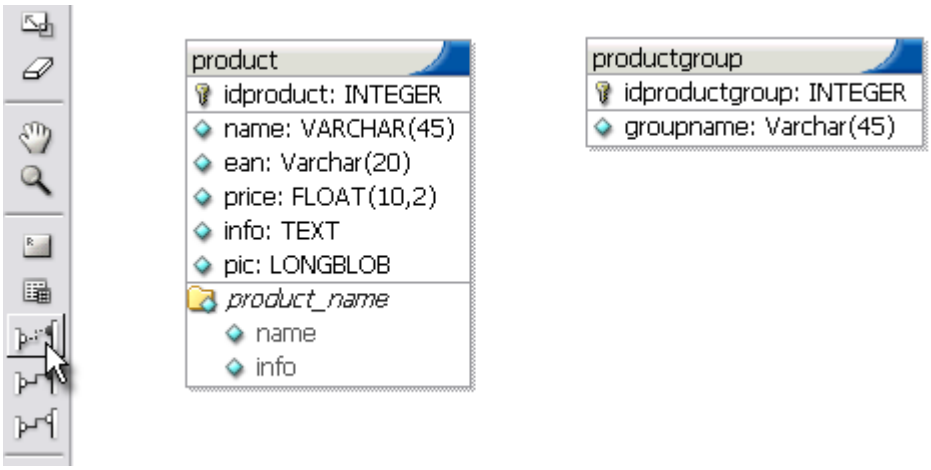
Para añadir una columna al índice empieza arrastrando la columna hacia abajo a la Lista de Columnas y suéltela. Se puede reordenar las columnas arrastrando las columnas. Para eliminar una columna pulsa el botón Eliminar.



Añadir columnas al índice

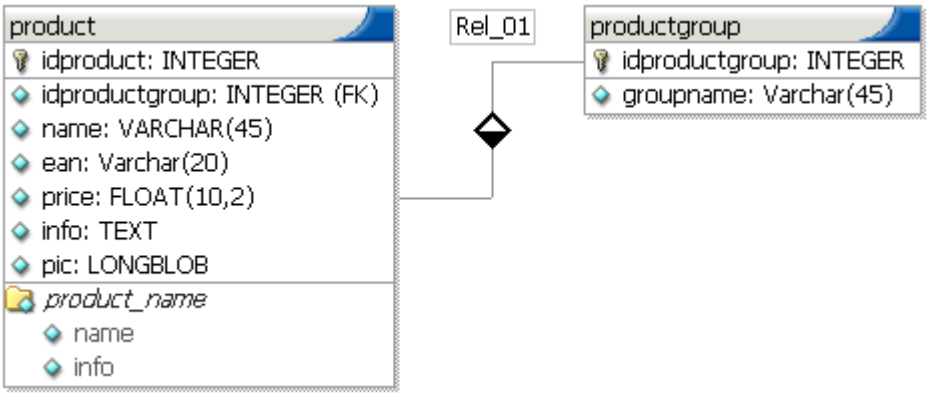
Hacer una relación:

Añade otra tabla al modelo usando la Herramienta de Tabla (p.e. grupoproducto) y define las columnas de la tabla. Ahora selecciona la Herramienta Relación 1:n (Relación No Identificadora) de la Barra de Herramientas.



Seleccionar la Herramienta de Selección 1:n (Relación No identificadora):

Para establecer la relación pulsa en la primera tabla (grupoproducto) y luego en la segunda tabla (producto). Teniendo en cuenta que la segunda tabla (producto) ahora tiene una clave foránea identificada por (FK) en la parte derecha del tipo de dato.



Tablas conectadas por una relación

Creación y Mantenimiento de Bases de Datos

Exportación SQL Tradicional: Como cualquier otra herramienta de modelado de bases de datos DBDesigner 4 puede exportar el modelo como un script SQL que puede ser ejecutado en cualquier herramienta de mantenimiento de bases de datos, como la línea de comandos de MySql.

Todos los comandos SQL CREATE TABLE y los Insert Estándar pueden exportarse como en cualquier otra herramienta de modelado de bases de datos a un fichero de script SQL .También es posible generar todos los comandos SQL DROP TABLE.

¿Qué es la Sincronización de Bases de Datos?

Con DBDesigner 4 se puede simplificar la tarea de crear y mantener la base de datos. DBDesigner 4 ofrece la posibilidad de conectarse a un servidor MySQL y crear y sincronizar una base de datos con el modelo diseñado.

Con sincronización queremos decir que DBDesigner 4 busca todas las tablas en la base de datos existente y comprueba las diferencias. Si la tabla existe en el modelo pero no en la base de datos, se ejecuta el comando SQL CREATE TABLE necesario. Si la tabla no existe en el modelo pero sí en la base de datos puede ser borrada dependiendo de las opciones del usuario.

Si la tabla existe en el modelo y en la base de datos también, se comparan todos los campos y si hay una diferencia, se ejecuta el comando SQL ALTER TABLE apropiado.

¿Qué es Ingeniería Inversa?

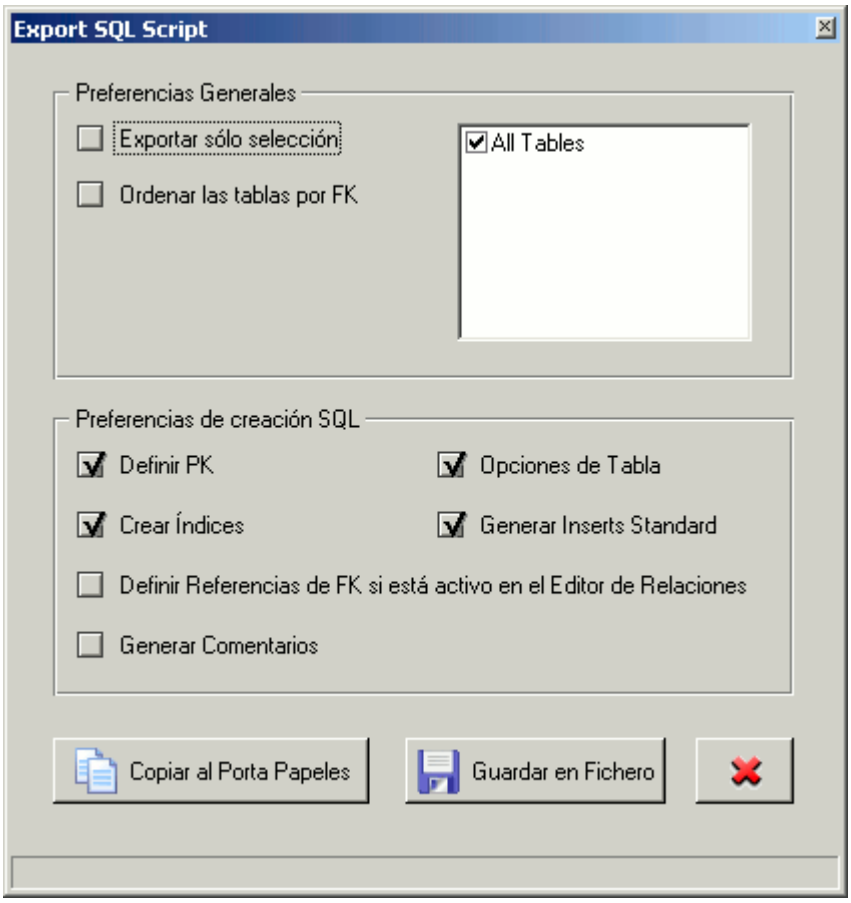
Realizar la ingeniería inversa en una base de datos significa conectarse al servidor de bases de datos, elegir una base de datos existente y construir automáticamente un modelo basado en la meta información de la base de datos.

En DBDesigner 4 toda la información de las tablas se extrae de la meta información y las relaciones entre tablas se extraen de nombres de tablas y campos. Las tablas se colocan en el modelo en orden alfabético siguiendo un esquema de cuadrícula definido por el usuario.

El proceso de ingeniería inversa es posible en MySQL, Oracle y cualquier otra base de datos accesible mediante ODBC.

Exportar Creates SQL

Para exportar el script create SQL en el modelo actual de base de datos selecciona Fichero-Exportar-Script Create SQL ... Aparecerá el diálogo de Exportación SQL.



Diálogo de exportación de Creates SQL

Para exportar los Creates SQL a un fichero, pulsar el botón [Guardar en Fichero]. Se preguntará el nombre del fichero y el destino. Pulsar [Guardar] para escribir el script en el disco.

Opciones Create SQL

La salida puede ser personalizada usando las siguientes Opciones Generales y las Opciones de Create SQL.

Exportar sólo tablas seleccionadas

Marca esta opción para exportar sólo las tablas seleccionadas. Las demás tablas no serán creadas en el fichero de script.

Ordenar Tablas por Clave Foránea

Se usa esta opción para cambiar el orden de creación. Por defecto las tablas se crean en orden alfabético. Cuando se usa las claves foráneas es necesario cambiar el orden

en que se crean las tablas. Las tablas sin relaciones que apunten a ellas han de ser creadas primero. Las demás tablas se crearán solo cuando existan todas las tablas de origen. Si hay una colección de relaciones cíclica, las tablas no pueden ser creadas. Se mostrará un mensaje de error. Aún así puedes exportar las tablas en orden alfabético.

Definir Claves Foráneas

Activa esta opción si quieres activar la creación de claves primarias.

Crear Índices

Activar esta opción si se quiere activar la creación de índices. No incluye las claves primarias. Activar la opción Definir Claves Foráneas para crear claves primarias.

Opciones de Generación de Tabla

Usar esta opción para habilitar las opciones de tabla en los comandos SQL CREATE TABLE. Es necesario solo si se ha especificado las Opciones de las Tablas para la tabla en el modelo.

Conexiones a la Base de Datos

Varias funciones en DBDesigner 4 usan conexiones a Bases de Datos. Se usan para establecer una conexión a una base de datos al seleccionar el servidor y base de datos apropiados.

Crear una nueva Conexión de Base de Datos

Una nueva Conexión de Base de Datos se crea en el Diálogo de Conexiones de Bases de Datos.

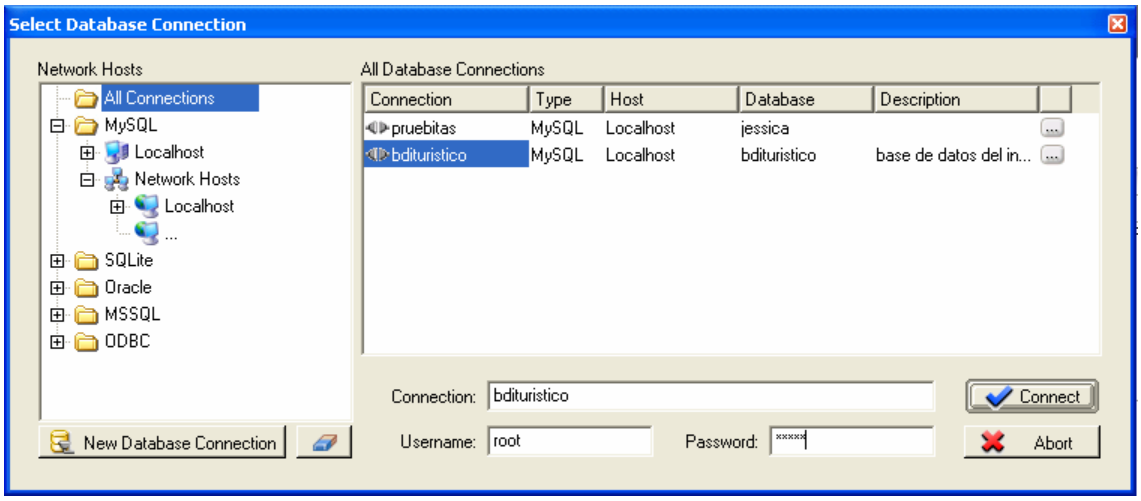
Para crear una nueva Conexión a Base de Datos, se pulsa sobre el botón de Nueva Conexión a Base de Datos. Se mostrará el Diálogo de Parámetros de Conexión. Especificar toda la información necesaria y pulsar OK para añadir la conexión a la lista de conexiones.

Conectar a una Base de Datos

Para conectarse a una base de datos se usa el Diálogo de Conexión a Bases de Datos, igual que al crear una nueva Conexión de Bases de Datos.

Selecciona la conexión apropiada de la lista de conexiones. Ingresar un nombre de usuario y un password y pulsar el botón Conectar para establecer la conexión

El Diálogo de Conexión a Bases de Datos



Diálogo de Conexión a Bases de Datos

El Diálogo de Conexión a Bases de Datos tiene tres áreas, el Árbol de Servidores de Red, La Lista de Conexiones y la sección de Usuario/Password.

Árbol de Servidores de Red

El Árbol de Servidores de Red muestra todos los servidores y sus bases de datos. Se usa como filtro para mostrar conexiones y para crear nuevas conexiones a servidores de bases de datos.

Cuando el primer nodo, llamado [Todas las Conexiones] está seleccionado, todas las conexiones introducidas se mostrarán en la Lista de Conexiones.

Para mostrar todas las conexiones al servidor local de MySQL selecciona [MySQL]-[Localhost]

Para mostrar todas las conexiones a servidores MySQL en la red selecciona [MySQL]-[Network Hosts]

Para mostrar todas las conexiones a un servidor MySQL específico de una red selecciona el nombre del servidor en el nodo [MySQL]-[Network Hosts]

Mostrar bases de Datos del Servidor

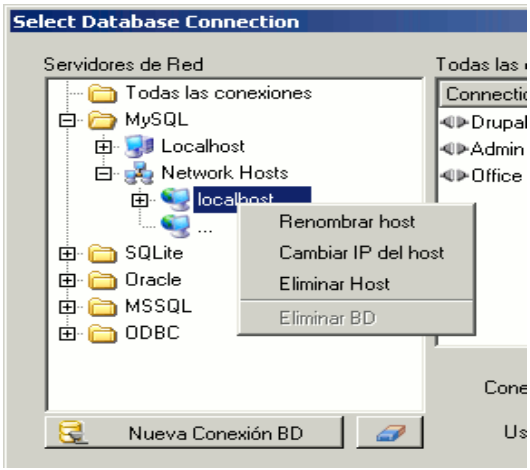
Para mostrar las bases de datos de un servidor dado pulsar en el icono [+] a la izquierda del nombre del servidor. DBDesigner 4 preguntará por el nombre de usuario y password que se utilizarán para validarse. Se debe tener en cuenta que el usuario especificado debe tener los permisos adecuados para ejecutar un comando SQL SHOW DATABASES.

Ingresar un nuevo Servidor

Después de la instalación de DBDesigner 4 es posible conectar únicamente al servidor local de MySQL. Para añadir un nuevo servidor de red se hace lo siguiente.

Cambiar los parámetros del Servidor

Para cambiar el nombre o la dirección IP de un servidor pulsar con el botón derecho en un Servidor. Se mostrará el menú popup de servidores.



Menú popup de servidores

Seleccionar la función que se quiera realizar del menú.

Eliminar un Servidor

Para eliminar un Servidor selecciona [Eliminar Host] del menú popup.

Crear una nueva base de datos

Es posible crear una nueva base de datos desde el Diálogo de Conexión a Base de Datos. Para crear una nueva base de datos muestra todos los servidores de bases de datos como ya se ha explicado. Se pulsa el último nodo bajo el nodo del Servidor que

tiene el texto [...]. Aparecerá el Diálogo de Nueva Base de Datos. Introduce el nombre de la base de datos y pulsar enter. Se creará la base de datos.

Eliminar base de datos

Es posible eliminar una base de datos desde el Diálogo de Conexión a Base de Datos. Para eliminar una base de datos muestra los servidores de bases de datos como ya se ha explicado. Pulsar sobre el nodo de base de datos con el botón derecho para mostrar el menú popup. Seleccionar [Eliminar Base de Datos].

Hay que tener en cuenta que una vez que la base de datos ha sido eliminada no puede ser restaurada. Para recuperarla se necesitará una copia de seguridad.

Lista de Conexiones

La Lista de Conexiones muestra las conexiones seleccionadas en el Arbol de Servidores de Red. Pulsar en la conexión deseada para poner la conexión en la Sección de Usuario.

Crear una nueva conexión

Para crear una nueva conexión seleccionar el Servidor al que se desea conectar del Arbol de Servidores de Red. Visualizar los Servidores de base de datos. Ahora arrastrar la base de datos a la que se quiere conectar en la Lista de Conexiones. Y se creará una nueva conexión.

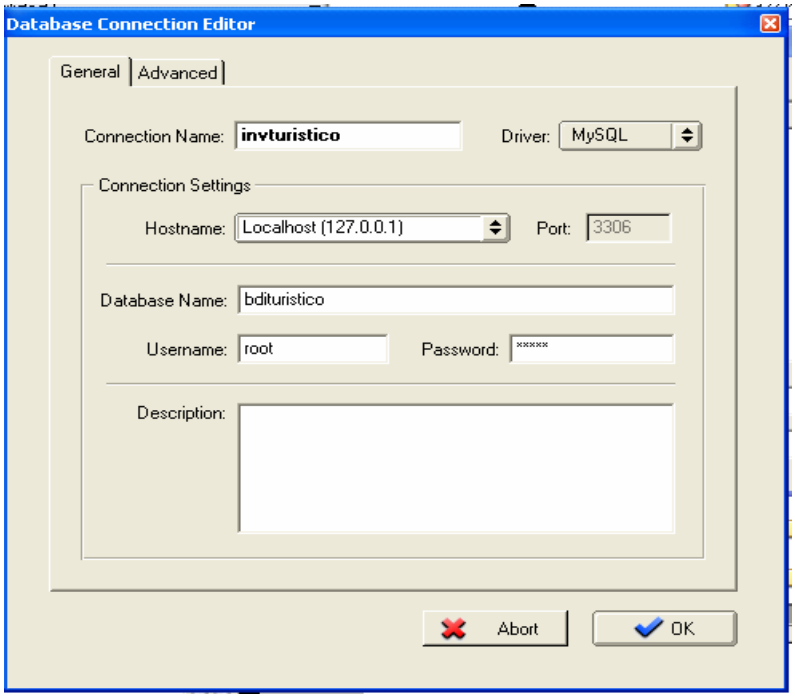
En lugar de arrastrar la base de datos se puede seleccionar la base de datos con el botón izquierdo y pulsar el botón [Nueva Conexión a la Base de Datos seleccionada].

La sección de Usuario

Cuando se selecciona una conexión a base de datos en la Lista de Conexiones se muestra el nombre de la conexión en la Sección de Usuario y DBDesigner 4 pregunta por el password de usuario. El password nunca se guarda en la conexión de base de datos por motivos de seguridad. Se debe pulsar return o el botón [Connect] para establecer la conexión con la base de datos. Si se realiza correctamente la conexión el diálogo de cierra.

Diálogo de Parámetros de Conexión

El Diálogo de Parámetros de Conexión se usa para cambiar la conexión.



Diálogo de Parámetros de Conexión

Nombre de la Conexión

Cada Conexión de Base de Datos está definida por un nombre único.

IP de Host

Se ingresa la dirección IP del servidor o su nombre de red. Es sólo necesario para conexiones con MySQL.

Nombre de la base de datos

Ingresar el nombre de la base de datos. Cuando se use el Driver de MySQL este es el nombre que se usará en el comando CREATE DATABASE SQL.

Driver

Se debe seleccionar un driver de base de datos de la lista desplegable. Cuando se selecciona un nuevo driver todos los valores se restablecen a su valor inicial.

Nombre de Usuario

Especifica el nombre de usuario usado para conectar a la base de datos.

Password

Especifica el password a usar para conectar a la base de datos.

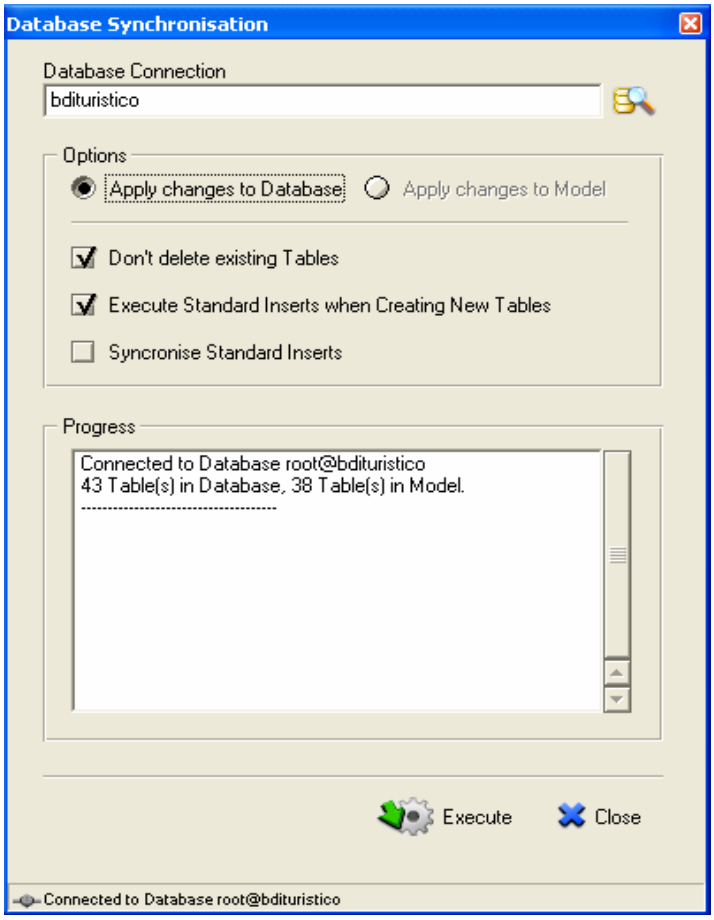
Descripción

Entra una descripción con algo de información acerca de la conexión a la base de datos.

Sincronización de la Base de Datos

Para sincronizar el modelo con una base de datos selecciona [Base de datos]->[Sincronización con la Base de Datos] del menú principal. La sincronización también puede llamarse pulsando el botón [Sync] en la paleta de Herramientas. Se debe tener en cuenta que un modelo vacío no puede sincronizarse.

Se mostrará el Diálogo de Conexión a Base de Datos. Hay que seleccionar la conexión a la base de datos con la que se va a hacer la sincronización. Ingresar el password de usuario y pulsar [Conectar] para establecer la conexión.



Database Syncronisation

Opciones de Sincronización

Cuando se establezca la conexión a la base de datos de forma correcta se mostrará el Diálogo de Sincronización con Bases de Datos.

Conexión a la Base de Datos

El nombre de la conexión de la base de datos se mostrará en la parte de arriba. Para conectar a una base de datos diferente pulsar el botón derecho en el nombre de conexión. El Diálogo de Conexión a la Base de Datos se volverá a mostrar y podrá seleccionarse una nueva conexión.

Aplicar cambios a la Base de Datos

Esta opción está seleccionada por defecto. La base de datos se modificará para reflejar los cambios hechos en el modelo. El modelo no se modificará al ejecutar la sincronización.

Aplicar cambios al Modelo

Para modificar el modelo en lugar de la base de datos selecciona esta opción. La base de datos no se modificará al ejecutar la función de sincronización.

No eliminar Tablas existentes

Seleccionar esta opción para mantener las tablas en la base de datos que no están presentes en el modelo. Si esta opción no está seleccionada esas tablas serán borradas.

Ejecuta Inserts Estándar cuando se Crean Nuevas Tablas

Si se activa esta opción los Inserts Estándar se insertarán en el nuevo crear tabla. Esta opción está seleccionada por defecto.

Ejecutando la sincronización

La sincronización no puede deshacerse. Antes de ejecutarla comprueba todas las opciones seleccionadas. Verifica la Lista de Progreso que muestra la información de la conexión y el número de tablas en la base de datos y el modelo. La información de la conexión se mostrará [usuario@basededatos].

Para ejecutar la sincronización pulsar el botón [Ejecutar] al final del diálogo.

La ejecución puede vigilarse a través de la Lista de Progreso. Se listarán todas las comprobaciones y cambios.

Opciones Generales

Usa funciones específicas de MySQL

Seleccionar esta opción cuando se realice la ingeniería inversa en una base de datos MySQL. Las funciones específicas de MySQL como DESCRIBE TABLE se usarán para generar una copia más exacta de la estructura de la tabla. No usar esta opción con ninguna otra base de datos.

Capítulo 3
PHP

3.1 Introducción

PHP es un lenguaje interpretado que sirve principalmente para realizar páginas html dinámicas, no es case sensitive (no distingue mayúsculas de minúsculas), salvo bugs en el tratamiento de objetos y variables.
En PHP no se declaran las variables y no tienen un tipo fijo, sino que una misma variable puede almacenar a lo largo de su vida valores de todo tipo (números, cadenas...).

3.2 Generalidades de PHP

PHP es un lenguaje de programación muy potente que, junto con html, permite crear sitios web dinámicos. Php se instala en el servidor y funciona con versiones de Apache, Microsoft IIs, Netscape Enterprise Server y otros.
La forma de usar php es insertando código php dentro del código html de un sitio web. Cuando un cliente (cualquier persona en la web) visita la página web que contiene éste código, el servidor lo ejecuta y el cliente sólo recibe el resultado. Su ejecución, es por tanto en el servidor, a diferencia de otros lenguajes de programación que se ejecutan en el navegador. Php permite la conexión a numerosas bases de datos, incluyendo MySQL, Oracle, ODBC, etc. Y puede ser ejecutado en la mayoría de los sistemas operativos (Windows, Mac OS, Linux, Unix.

Antes de proceder a la instalación de PHP se debe tener ya instalado Apache, es necesario por lo tanto conocer su definición y algunos beneficios que esta herramienta nos brinda.

3.3 Apache

Definición.- Apache está diseñado para ser un servidor web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y los diferentes entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades, o que una misma característica o funcionalidad sea

implementada de diferente manera para obtener una mayor eficiencia. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios web elegir que características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor.

Como Funciona: Apache extiende este diseño modular hasta las funciones más básicas de un servidor web. El servidor viene con una serie de Módulos de MultiProcesamiento que son responsables de conectar con los puertos de red de la máquina, aceptar las peticiones, y generar los procesos hijo que se encargan de servirlas

Beneficios: Apache puede soportar de una forma más fácil y eficiente una amplia variedad de sistemas operativos. En concreto, la versión de Windows de Apache es mucho más eficiente, porque el módulo mpm_winnt puede usar funcionalidades nativas de red en lugar de usar la capa POSIX como hace Apache 1.3. Este beneficio se extiende también a otros sistemas operativos que implementan sus respectivos MPMs.

El servidor puede personalizarse mejor para las necesidades de cada sitio web. Por ejemplo, los sitios web que necesitan más que nada escalibildad pueden usar un MPM hebrado como worker, mientras que los sitios web que requieran por encima de otras cosas estabilidad o compatibilidad con software antiguo pueden usar prefork. Además, se pueden configurar funcionalidades especiales como servir diferentes hosts con diferentes identificadores de usuario (perchild).

3.4 Instalación de PHP

Para instalar php se desempaqueta el archivo php-4.1.2-Win32.zip en algún directorio de windows puede ser c:\php se busca el archivo PHP4TS.DLL y se copia este en el directorio c:\windows\system, luego copie todos los archivos que están en c:\php\php-4.1.2-Win32\dlls en la carpeta c:\windows\system. Si utiliza Windows XP copiarlos también a la carpeta c:\windows\system32

Se crea la carpeta C:\Archivos de programa\Apache Group\Apache\php\ y se copia todo lo que esta en C:\php\php-4.1.2-Win32\ incluido las carpetas a C:\Archivos de programa\Apache Group\Apache\php\

En el directorio c:\php\php-4.1.2-Win32\ existen dos archivos el php.ini-recommended y el php.ini-dist el primer archivo es la configuración de php con algunos cambios para optimizar su ejecución el segundo archivo es la configuración de php con las opciones por default, se va ha escoger el primer archivo y se lo copiara en el directorio c:\windows cambiándolo de nombre a php.ini.

Editamos el archivo php.ini y buscamos la línea extension_dir = ./ y la cambiamos a:

extension_dir = c:\Archivos de programa\Apache Group\Apache\php\extensions

esto nos permite agregar algunas capacidades extras a php. También buscamos la línea:

doc_root =

y le modificamos para que quede de la siguiente forma:

doc_root = c:\Archivos de programa\Apache Group\Apache\htdocs

se busca la siguiente linea
register_globals = On

se cambia a

register_globals = Off

buscamos las siguientes líneas:

;extension=php_zlib.dll
;extension=php_ldap.dll

borramos el; de cada línea para que quede de la siguiente forma:

extension=php_zlib.dll
extension=php_ldap.dll

Esto es para agregar esto módulos extras a php hay más módulos extras que se los puede activar quitando; del comienzo de la línea. Se guarda el archivo php.ini con los cambios realizados. A continuación se edita el archivo httpd.conf que es el archivo de configuración de apache que esta ubicado en el directorio:

C:\Archivos de Programa\Apache Group\Apache\conf\httpd.conf

Se busca la siguiente línea:

DirectoryIndex index.html

Se modifica esta línea por lo siguiente:

DirectoryIndex home.htm index.html index.php index.php3
ScriptAlias /php/ "C:/Archivos de programa/Apache Group/Apache/php/"
AddType application/x-httpd-php3 .php3
AddType application/x-httpd-php3-source .phps
AddType application/x-httpd-php .php .php4
AddType application/x-httpd-php-source .phps
AddHandler cgi-script .cgi
AddHandler php3-script .php3
AddHandler php-script .php .php4
Action php3-script /php/php.exe
Action php-script /php/php.exe

Se guarda el archivo http.conf y se borra el directorio c:\php con todo lo que esta dentro de él.

Variables y operadores: Todas las variables deben precedidas por signo dólar ($), y le asignamos contenido con el signo igual (=). Con las variables, PHP distingue entre mayúsculas y minúsculas, por lo que no es lo mismo $myvar que $Myvar, éstas son dos variables totalmente distintas.

El uso de la barra invertida, como en \n, no es obligatorio, pero ayuda a la depuración del código que enviamos al navegador, además del \n existen otros usos:

\" Carácter dobles comillas
\\ Carácter barra invertida
\n Nueva línea
\r Retorno de carro
\t Tabulador horizontal

Constantes: Las constantes son similares a las variables, con la salvedad de que no llevan el signo dólar delante, y sólo la podemos asignar una vez. Para definir una constantes usaremos la función define como sigue:

```
<html>
<body>

<?php
define ("CONSTANTE", "Hola Mundo");
printf (CONSTANTE);
?>

</body>
</html>
```

PHP crea diversas constantes al arrancar, como PHP_VERSION que contiene la versión de PHP, TRUE que le asigna 1 o FALSE que le asigna 0.

Operadores Aritméticos:

$a + $b Suma
$a - $b Resta
$a * $b Multiplicación
$a / $b &ss=codigoenlinea>$a / $b División
$a % $b Resto de la división de $a por $b
$a++ Incrementa en 1 a $a
$a-- Resta 1 a $a

Operadores de Cadenas:

El único operador de cadenas que existen es el de concatenación, el punto. Sin embargo PHP dispone de toda una batería de funciones que nos permitirán trabajar cómodamente con las cadenas.

```
$a = "Hola";
$b = $a . "Mundo";   // Ahora $b contiene "Hola Mundo"
```

En este punto hay que hacer una distinción, la interpretación que hace PHP de las simples y dobles comillas. En el segundo caso PHP interpretará el contenido de la cadena.

```
$a = "Mundo";
echo = 'Hola $a';   //Esto escribirá "Hola $a"
echo = "Hola $a";  //Esto escribirá "Hola Mundo&q; //Esto escribirá "Hola Mundo"
```

Operadores de Comparación:

$a < $b $a menor que $b
$a > $b $a mayor que $b
$a <= $b $a menor o igual que $b
$a >= $b $a mayor o igual que $b
$a == $b $a igual que $b
$a != $b $a distinto que $b

Operadores Lógicos:

$a AND $b Verdadero si ambos son verdadero
$a && $b Verdadero si ambos son verdadero
$a OR $b Verdadero si alguno de los dos es verdadero
$a !! $b Verdadero si alguno de los dos es verdadero
$a XOR $b Verdadero si sólo uno de los dos es verdadero !$a Verdadero si $a es falso, y recíprocamente

Operadores de Asignación:

$a = $b Asigna a $a el contenido de $b
$a += $b Le suma a $b a $a
$a -= $b Le resta a $b a $a
$a *= $b Multiplica $a por $b y lo asigna a $a
$a /= $b Divide $a por $b y lo asigna a $a
$a .= $b Añade la cadena $b a la cadena $a

Sentencias de control: Las sentencias de control permiten ejecutar bloque de códigos dependiendo de unas condiciones. Para PHP el 0 es equivalente a Falso y cualquier otro número es Verdadero.

IF...ELSE

La sentencia IF...ELSE permite ejecutar un bloque de instrucciones si la condición es Verdadera y otro bloque de instrucciones si ésta es Falsa. Es importante tener en cuenta q instrucciones si ésta es Falsa. Es importante tener en cuenta que la condición que evaluemos ha de estar encerrada entre paréntesis (esto es aplicable a todas la sentencias de control).

if (condición) {

Este bloque se ejecuta si la condición es VERDADERA

```
} else {

Este boque se ejecuta si la condición es FALSA

}
```

Existe una forma sencilla de usar la sentencia IF cuando no tenemos que usar el ELSE y solo tenemos que ejecutar una línea de código.

```
if ($a > 4) echo "$a es mayor que 4";
```

IF...ELSEIF...ELSE

La sentencia IF...ELSEIF...ELSE permite ejecuta varias condiciones en cascada. Para este caso veremos un ejemplo, en el que utilizaremos los operadores lógicos.

```
<?php

if ($nombre == ""){

echo "Tú no tienes nombre";

} elseif (($nombre=="eva") OR ($nombre=="Eva")) {

echo "

echo "Tu nombre es EVA";<

} else {

echo "Tu nombre es " . $nombre;

}
```

SWITCH...CASE...DEFAULT

Una alternativa a IF...ELSEIF...ELSE, es la sentencia SWITCH, la cuál evalúa y compara cada expresión de la sentencia CASE con la expresión que evaluamos, si llegamos al final de la lista de CASE y encuentra una condición Verdadera, ejecuta el código de bloque que haya en DEFAULT. Si encontramos una condición verdadera debemos ejecutar un BREAK para que la sentencia SWITCH no siga buscando en la lista de CASE. Veamos un ejemplo.

```
<?php

switch ($dia) {

case "Lunes":

echo "Hoy es Lunes";

break;

case "Martes":

echo "Hoy es Martes";

break;

case "Miercoles":

echo "Hoy es Miercoles";

break;

case "Jueves":
```

```
echo "Hoy es Jueves";

break;

case "Viernes":

echo "Hoy es Viernes";

break;

case "Sábado":

echo "Hoy es Sábado";

break;

case "Domingo":

echo "Hoy es Domingo";

break;

default:

default:

echo "Esa cadena no corresponde a ningún día de la semana";

}

?>
```

WHILE

La sentencia WHILE ejecuta un bloque de código mientras se cumpla una determinada condición.

```
<?php

$num = 1;

while ($num < 5) {

echo $num;

$num++

}

?>
```

Podemos romper un bucle WHILE utilizando la sentencia BREAK.

DO...WHILE

Esta sentencia es similar a WHILE, salvo que con esta sentencia primero ejecutamos el bloque de código y después se evalúa la condición, por lo que el bloque de código se ejecuta siempre al menos una vez.

```
<?php

$num = 1;

do {
```

```
echo $num;

if ($num == 3){

echo "Aquí nos salimos \n";

break

}

$num++

} while ($num < 5);

?>
```

FOR

El bucle FOR no es estrictamente necesario, cualquier bucle FOR puede ser sustituido fácilmente por otro WHILE. Sin embargo, el bucle FOR resulta muy útil cuando debemos ejecutar un bloque de código a condición de que una variable se encuentre entre un valor mínimo y otro máximo. El bucle FOR también se puede romper mediante la sentencia BREAK.

```
<?php

for ($num = 1; Snum <=5; $num++){

echo $num;

if ($num == 3){

echo "Aquí nos salimos \n";
```

```
break

}

}

?>
```

Las tablas: Las tablas (o array en inglés), son muy importantes en PHP, ya que generalmente, las funciones que devuelven varios valores, como las funciones ligadas a las bases de datos, lo hacen en forma de tabla.

En PHP disponemos de dos tipos de tablas. El primero sería el clásico, utilizando índices:

```
<?php

$ciudad[] = "París";

$ciudad[] = "París";

$ciudad[] = "Roma";

$ciudad[] = "Sevilla";

$ciudad[] = "Londres";

print ("yo vivo en " . $ciudad[2] . "<BR>\n");

?>
```

Esta es una forma de asignar elementos a una tabla, pero una forma más formal es utilizando la función array

```
<?php

$ciudad = array("París", "Roma", "Sevilla", "Londres");

//contamos el número de elementos de la tabla

$numelentos = count($ciudad);

//imprimimos todos los elementos de la tabla

for ($i=0; $i < $numelentos; $i++)

{

print ("La ciudad $i es $ciudad[$i] <BR>\n");

}

?>
```

Un segundo tipo, son las tablas asociativas, en las cuáles a cada elemento se le asigna un valor (key) para acceder a él.

Para entenderlo, que mejor que un ejemplo, supongamos que tenemos una tabla en al que cada elemento almacena el número de visitas a nuestra web por cada día de la semana.

Utilizando el método clásico de índices, cada día de la semana se representaría por un entero, 0 para lunes, 1 para martes, etc.

```
$visitas[0] = 200;

$visitas[1] = 186;
```

si usamos las tablas asociativas sería

$visitas["lunes"] = 200;

$visitas["martes"] = 186;

o bien,

$visitas = array("luodigo">$visitas = array("lunes"=>200; "martes"=>186);

Ahora bien, recorrer una tabla y mostrar su contenido es sencillo utilizando los índices, pero ¿cómo hacerlo en las tablas asociativas?. La manipulación de las tabas asociativas se a hace través de funciones que actúan sobre un puntero interno que indica la posición. Por defecto, el puntero se sitúa en el primer elemento añadido en la tabla, hasta que es movido por una función:

current - devuelve el valor del elemento que indica el puntero
pos - realiza la misma función que current
reset - mueve el puntero al primer elemento de la tabla
end - mueve el puntero al último elemento de la tabla
next - mueve el puntero al elemento siguiente
prev - mueve el puntero al elemento anterior
count&n count - devuelve el número de elementos de una tabla.

EACH
La función each()devuelve el valor del elemento actual, en este caso, el valor del elemento actual y su clave, y desplaza el puntero al siguiente, cuando llega al final devuelve FALSO, y termina el bucle while().

Tablas multidimensionales

Las tablas multidimensionales son simplemente tablas en las cuales cada elemento es a su vez otra tabla.

```
<?php

$calendario[] = array (1, "enero", 31);

$calendario[] = array (2, "febrero", 28);

$calendario[] = arra

$calendario[] = array (3, "marzo", 31);

$calendario[] = array (4, "abril", 30);

$calendario[] = array (5, "mayo", 31);



while (list($clave, $valor ) = each($calendario)){

{

$cadena = $varlor[1];

$cadena .= " es el mes número " . $valor[0];

$cadena .= "y tiene " . $varlor[2] . " días<BR>";

echo $cadena;

}

?>
```

La función list() es más bien un operador de asignación, lo que hace es asignar valores a unas lista de variables. En este caso los valores son extraídos de una tabla por la función each().

Los formularios: Los Formularios no forman parte de PHP, sino del lenguaje estándar de Internet, HTML. Lo que viene a continuación es HTML y no PHP.
Todo formulario comienza con la etiqueta <FORM ACTION="lo_que_sea.php" METHOD="post/get">. Con. Con ACTION indicamos el script que va procesar la información que recogemos en el formulario, mientras que METHOD nos indica si el usuario del formulario va ha enviar datos (post) o recogerlos (get). La etiqueta <FORM> indica el final del formulario.
A partir de la etiqueta <FORM> vienen los campos de entrada de datos que pueden ser:

Cuadro de texto:
<input type="text" name="nombre" size="20" value="jose">
Cuadro de texto con barras de desplazamiento:
<textarea rows="5" name="descripcion" cols="20">Es de color rojo</textarea>
Casilla de verificación:
<input type="checkbox" name="cambiar" value="ON">
Botón de opción:
<input type="radio" value="azul" checked name="color">
Menú desplegable:
<select size="1&qu class="codigo"><select size="1" name="dia">
<option selected value="lunes">lunes</option>
<option>martes</option>
<option value="miercoles">miercoles</option>
</select>
Botón de comando:
<input type="submit" value="enviar" name="enviar">
Campo oculto:
<input type="hidden" name="edad" value="55">
Este último tipo de campo resulta especialmente útil cuando queremos pasar datos ocultos en un formulario.

Como se ha observado todos lo tipos de campo tienen un modicador llamado name, que no es otro que el nombre de la variable con la cual recogeremos los datos en el script indicado por el modificador ACTION de la etiqueta FORM, con value establecemos un valor por defecto.

Las funciones: Muchas veces, cuando trabajamos en el desarrollo de una aplicación, nos surge la necesidad de ejecutar un mismo bloque de código en diferentes partes de nuestra aplicación. Una Función no es más que un bloque de código al que le pasamos una serie de parámetros y nos devuelve un valor. Como todos los lenguaje de programación, PHP trae una gran cantidad de funciones para nuestro uso, pero las funciones más importantes son las que nosotros creamos.

Para declarar una función debemos utilizar la instrucción function seguido del nombre que le vamos a dar, y después entre paréntesis la lista de argumentos separados por comas, aunque también habrá funciones que no recojan ningún argumento.

```
function nombre_de_funcion (arg_1, arg_2, ..., arg_n)
{
bloque de código
}
```

Cualquier instrucción válida de PHP puede aparecer en el cuerpo (lo que antes hemos llamado bloque de código) de una función, incluso otras funciones y definiciones de clases. En PHP no podemos redefinir una función previamente declarada, y además en PHP3, las funciones deben definirse siempre antes de que se invoquen, en PHP4 este requerimiento ya no existe.

La instrucción RETURN

Cuando invocamos una función, la ejecución del programa pasa a ejecutar las líneas de código que contenga la función, y una vez terminado, el programa continua su ejecución desde el punto en que fue llamada la función.

Existe una manera de terminar la ejecución de la función aunque aún haya código por ejecutar, mediante el uso de la instrucción return terminamos la ejecución del

código de una función y devolvemos un valor. Podemos tener varios return en nuestra función, pero por lo general, cuantos más return tengamos menos reutilizable será nuestra función.

Con la instrucción return puede devolverse cualquier tipo de valor, incluyendo tablas y objetos. PHP solo permite a las funciones devolver un valor, y para solventar este pequeño problema, si queremos que nuestra función devuelva varios tenemos que utilizar una tabla (array).

Parámetros de las funciones: Existen dos formas de pasar los parámetros a una función, por valor o por referencia. Cuando pasamos una variable por valor a una función, ocurra lo que ocurra en ésta en nada modificará el contenido de la variable. Mientras que si lo hacemos por referencia, cualquier cambio acontecido en la función sobre la variable lo hará para siempre.
En PHP, por defecto, las variables se pasan por valor. Para hacerlo por referencia debemos anteponer un ampersand (&) a la variable.

```
<?php
function suma ($x, $y)
{
$x = $x + 1;
return $x+$y;
}
$a = 1;
$b = 2;

//parámetros por valor
echo suma ($a, $b);    // imprimirá 4
echo $a;    // imprimirá 1

//parámetros por referencia
echo suma (&$a, $b);    // imprimirá 4
echo $a;      //imprimirá 2
?>
```

Si queremos que un parámetro de una función se pase siempre por referencia debemos anteponer un ampersand (&) al nombre del parámetro en la definición de la función. En PHP podemos definir valores por defecto para los parámetro de una función. Estos valores tienen que ser una expresión constante, y no una variable o miembro de una clase. Además cuando usamos parámetros por defectos, éstos deben estar a la derecha de cualquier parámetro sin valor por defecto, de otra forma PHP nos devolverá un error.

Llegados a este punto, damos un paso atrás y volvemos a las variables, para distinguir entre variables estáticas (static) y globales (global). Las variables estáticas se definen dentro de una función, la primera vez que es llamada dicha función la variable se inicializa, guardando su valor para posteriores llamadas.

```
<?php
function contador ()
{
static $count = 0;
$count = $count + 1;
return $count;
}
echo contador()."<BR>"; // imprimirá 1
echo contador()."<BR>"; // imprimirá 2
echo contador()."<BR>"; // imprimirá 3
?>
```

Las variables globales, no se pueden declarar dentro de una función, lo que hacemos es llamar a una variable que ya ha sido declarada, tomando el valor que tenga en ese momento, pudiendo ser modificado en la función.

```
<?php
var $a = 1;
```

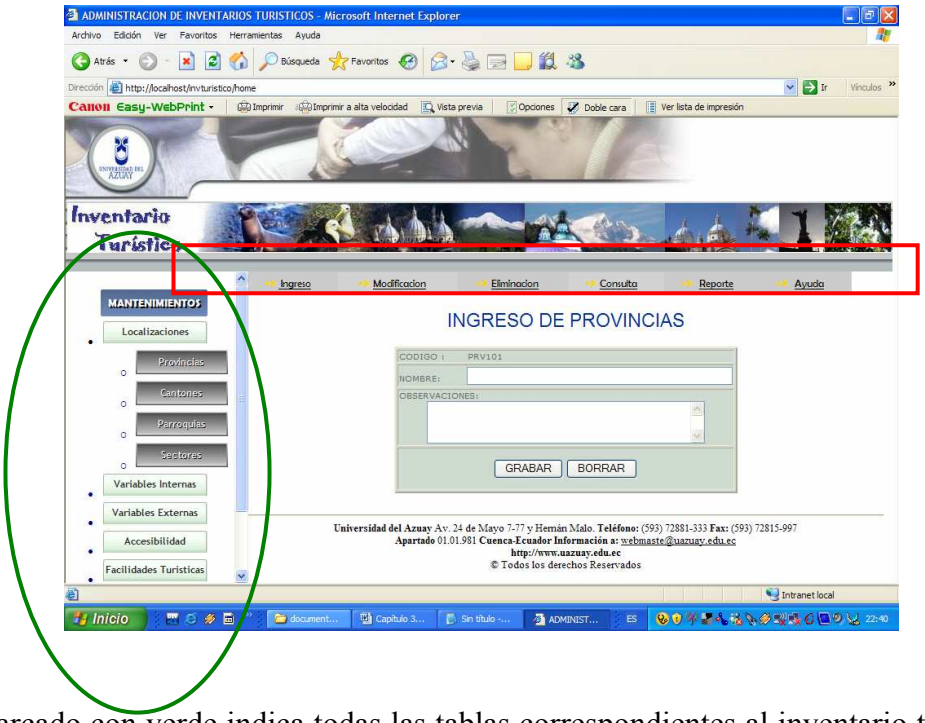
```
function ver_a()
{
global $a;
echo $a."<BR>"; // imprimirá el valor de $a

$a += 1; // sumamos 1 a $a
}
echo ver_a(); // imprimirá 1
echo ver_a(); // imprimirá 2
$a = 7;
echo ver_a(); // imprimirá 7
echo ver_a(); // imprimirá 8
?>
```

Funciones Variable: PHP soporta el concepto de funciones variables, esto es significa que si una variable tiene unos paréntesis añadidos al final, PHP buscará un función con el mismo nombre que el contenido de la variable, e intentará ejecutarla.

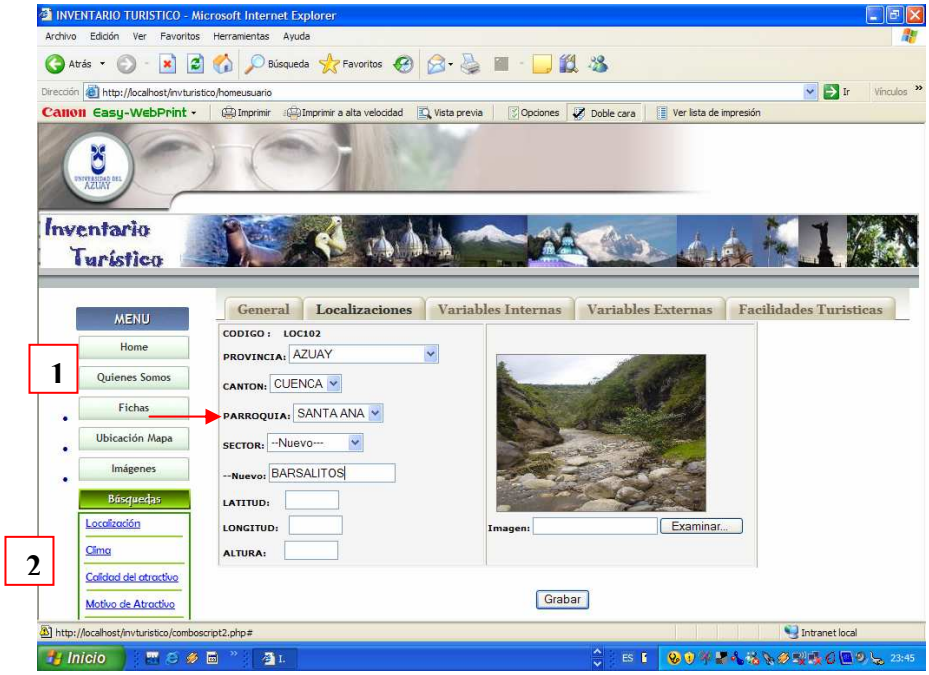
Recursión: PHP también permite la recursión, es decir, una función se puede llamar así misma.

3.5 Diseño de la aplicación Web

Se procedió a desarrollar la aplicación web basándose en el manejo de las siguientes pantallas:
La pantalla que se muestra a continuación es la correspondiente al administrador, su manejo se centra en el uso de dos menús principales:



El marcado con verde indica todas las tablas correspondientes al inventario turístico, las mismas que se clasificaron y agruparon según diferentes aspectos mencionados anteriormente, de tal manera que el administrador las maneje y encuentre con facilidad. El menú marcado con rojo se mostrará cuando se seleccione cualquier opción del menú anterior, ya que en este se encuentran todas las actividades de mantenimiento como son el ingreso, consulta, modificación, eliminación y reporte, además se contará con una opción de ayuda en la que se presentará un manual de usuario que contendrá operaciones clave para dar un buen mantenimiento de datos.



La siguiente pantalla es la que manejará el usuario (estudiantes), básicamente se enfocará hacia la ficha de ingreso de atractivos turísticos, brindando una serie de opciones relacionadas con los mismos de manera que la información sea lo más detallada y clara posible. El menú señalado con el numero uno se refiere a opciones generales y el ingreso en sí de la ficha, el numero dos presenta opciones de búsqueda y reportes que mostrarán resultados de los datos ingresados hasta el momento.

3.6 Programación

Comenzando con MySQL: Una vez instalado MySQL, vamos a crear nuestra BD. MySQL utiliza una tabla de permisos de usuarios, por defecto, en la instalación crea el usuario root sin password. Se debe crear distintos usuarios con distintos permisos. Entre ellos, el usuario administrador de MySQL, con todos los permisos, y como recomendación de seguridad, el usuario nobody sólo con el permiso de ver (SELECT). Para crear nuestra BD, debemos ser el administrador de MySQL o el root,para ello haremos lo siguiente:

mysqladmin create mybd

Conectar a MySQL desde PHP: Al tener datos en nuestra BD, con el siguiente script nos conectaremos a la BD del servidor MySQL para obtener los datos de un registro.

Conexión al MySQL

```
<html>
<body>
  <?php
$link = mysql_connect("localhost", "nobody");
mysql_select_db("mydb", $link);
$result = mysql_query("SELECT * FROM agenda", $link);
echo "Nombre: ".mysql_result($result, 0, "nombre")."<br>";
echo "Dirección: ".mysql_result($result, 0, "direccion")."<br>";
echo "Teléfono :".mysql_result($result, 0, "telefono")."<br>";
```

```
echo "E-Mail :".mysql_result($result, 0, "email")."<br>";
?>
  </body>
</html>
```

En la primera línea del script nos encontramos con la función mysql_connect(), que abre una conexión con el servidor MySQL en el Host especificado (en este caso la misma máquina en la que está alojada el servidor MySQL,localhost). También debemos especificar un usuario (nobody, root, etc.), y si fuera necesario un password para el usuario indicado (mysql_connect("localhost", "root", "clave_del_root")). El resultado de la conexión es almacenado en la variable $link.
Con mysql_select_db() PHP le dice al servidor que en la conexión $link nos queremos conectar a la base de datos mydb. Podríamos establecer distintas conexiones a la BD en diferentes servidores, pero nos conformaremos con una.
La siguiente función mysql_query(), es la que hace el trabajo duro, usando el identificador de la conexión ($link), envía una instrucción SQL al servidor MySQL para que éste la procese. El resultado de ésta operación es almacenado en la variable $result.

Finalmente, mysql_result() es usado para mostrar los valores de los campos devueltos por la consulta ($result). En este ejemplo mostramos los valores del registro 0, que es el primer registro, y mostramos el valor de los campos especificados.

Mostrar los datos de una consulta: Ahora que ya sabemos conectar con el servidor de BD, veremos como mostrar los datos por pantalla.

```
<html>
<body>
<?php
$link = mysql_connect("localhost", "nobody");
mysql_select_db("mydb", $link);
$result = mysql_query("SELECT nombre, email FROM agenda", $link);
echo "<table border = '1'> \n";
```

```
echo "<tr><td>Nombre</td><td>E-Mail</td></tr> \n";
while ($row = mysql_fetch_row($result)){
      echo ""<tr><td>$row[0]</td><td>$row[1]</td></tr> \n";
}
echo "</table> \n";
?>

</body>
</html>
```

En este script hemos introducido dos novedades, la más obvia es la sentencia de control while(), que tiene un funcionamiento similar al de otros lenguajes, ejecuta una cosa mientras la condición sea verdadera. En esta ocasión while() evalúa la función mysql_fetch_row(), que devuelve un array con el contenido del registro actual (que se almacena en $row) y avanza una posición en la lista de registros devueltos en la consulta SQL.

La función mysql_fetch_row() tiene un pequeño problema, es que el array que devuelve sólo admite referencias numéricas a los campos obtenidos de la consulta. El primer campo referenciado es el 0, el segundo el 1 y así sucesivamente. En el siguiente script solucionaremos este pequeño inconveniente.

Consulta modificada de BD

```
<html>
<body>
<?php
$link = mysql_connect("localhost", "nobody");
mysql_select_db("mydb", $link);
$result = mysql_query("SELECT nombre, email FROM agenda", $link);
if ($row = mysql_fetch_array($result)){
  echo "<table border = '1'> \n";
  echo "<tr><td>Nombre</td><td>E-Mail</td></tr> \n";
  do {
    echo "<tr><td>".$row["nombre"]."</td><td>".$row["email"]."</td></tr> \n";
```

```
  } while ($row = mysql_fetch_array($result));
  echo "</table> \n";
} else {
echo "¡ No se ha encontrado ningún registro !";
}
?>
</body>
</html>
```

Esencialmente, este script hace lo mismo que el anterior. Almacenamos en $row el registro actual con la función mysql_fetch_array() que hace exactamente lo mismo que mysql_fetch_row(), con la excepción que podemos referenciar a los campos por su nombre ($row["email"]), en vez de por un número.
Con la sentencia if/else, asignamos a $row el primer registro de la consulta, y en caso de no haber ninguno (else) mostramos un mensaje ("No se ha encontrado..."). Mientras que con la sentencia do/while, nos aseguramos que se nos muestren todos los registros devueltos por la consulta en caso de haber más de uno.
Hay que destacar la utilización del punto (.), como operador para concatenar cadenas.

Capitulo 4
MANUAL DEL USUARIO

4.1 Introducción

Para acceder a la página principal de Inventario Turístico se ingresará mediante la página web de la Universidad del Azuay, dentro de la cual existirá un enlace a la aplicación del inventario turístico.

4.2 Desarrollo

La pantalla inicial es la presentación gráfica de nuestra aplicación en la cual se pide el registro de los usuarios, tomando como identificación su cédula y la contraseña que han sido previamente ingresados por parte del administrador el cual le ha otorgado un tipo de acceso.(Gráfico M1.1)

Gráfico M1.1 Pantalla de Acceso
homeprincipal.html

El botón **Borrar** reestablece los valores ingresados.

El botón **Ingresar** valida los datos ingresados y si son correctos accederá al enlace correspondiente según sus privilegios establecidos, es decir en caso de ser **administrador** se visualizará la pantalla principal del administrador. En esta pantalla el administrador tiene las opciones de ingreso, consulta, modificación, eliminación y reporte de las diferentes tablas que forman parte de la base de datos que se maneja en el inventario turístico.

La opción **Ingresar** pide y valida los datos que se registrarán en las diferentes tablas.(Gráfico M1.2)

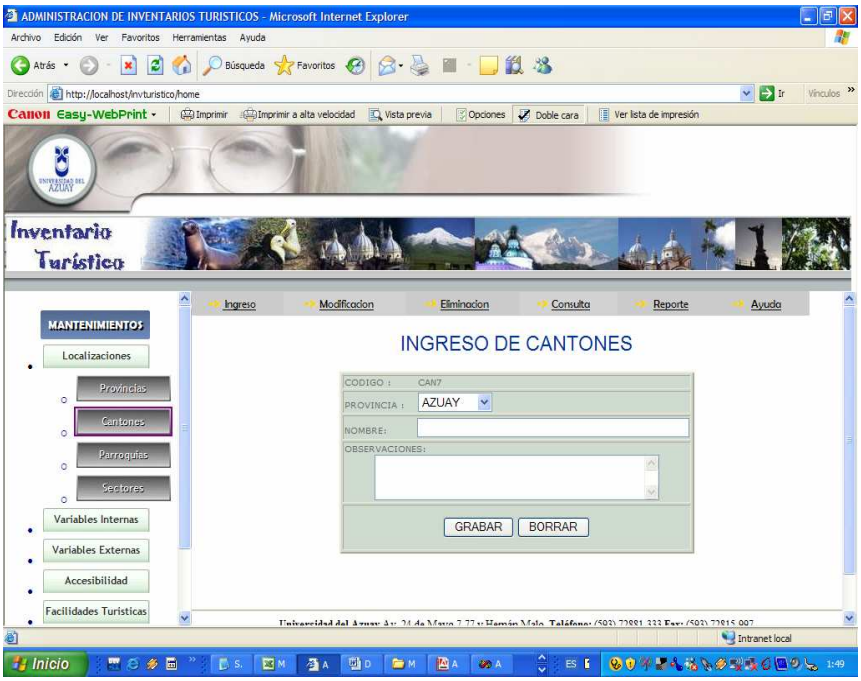


Gráfico M1.2 Pantalla principal del administrador

homeadmin.html

En el caso de escoger la opción de modificación se mostrará una lista de los datos existentes hasta el momento, escogiendo de una manera dinámica la información a ser modificada.(Gráfico M1.3)

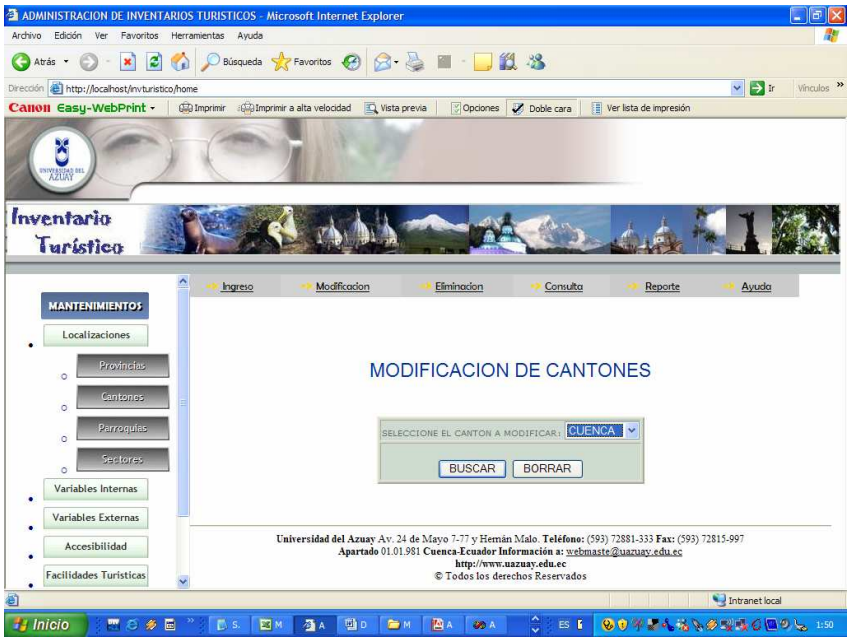


Gráfico M1.3 Pantalla de modificación

modificacantones.php

Al elegir el registro a modificar se mostrará la información actualmente guardada para que pueda ser cambiada de la mejor forma según criterio, de igual manera al momento de pulsar el botón **Modificar,** se validará los datos antes de realizar la operación a la base de datos.(Gráfico M1.4)

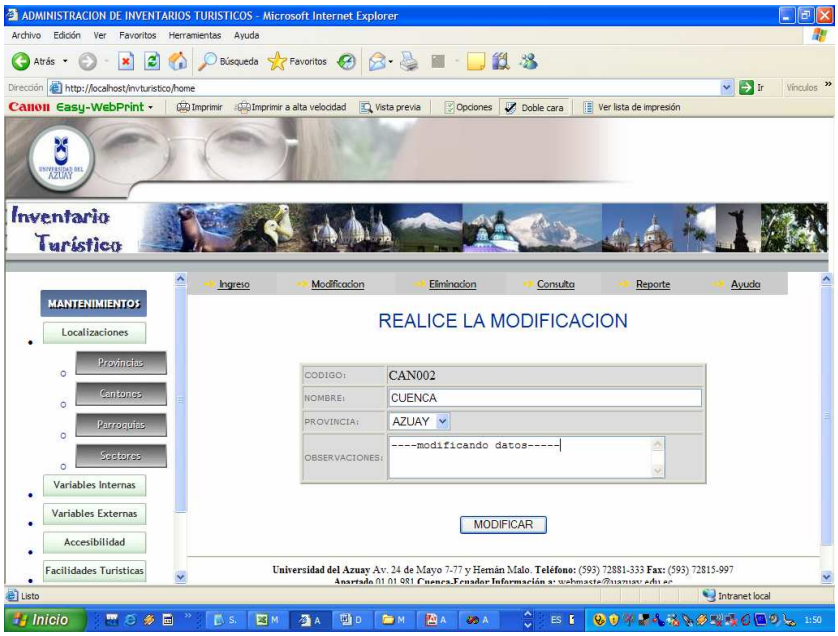


Gráfico M1.4 Pantalla de modificación

modificacanton.php

Otra de las opciones es la eliminación de datos que al igual que la modificación presenta un listado de todos los registros actuales que podrán ser eliminados.(Gráfico M1.5)

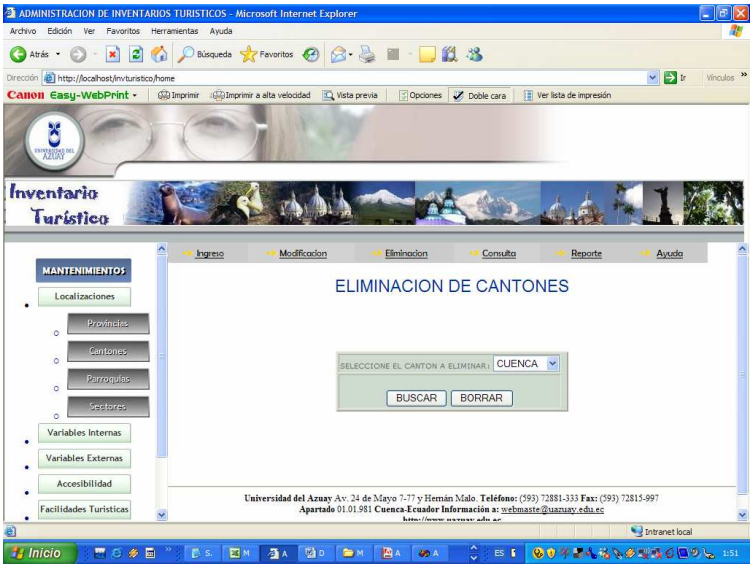


Gráfico M1.5 Pantalla de eliminación de datos

eliminacantones.php

A continuación se visualiza la información correspondiente al registro seleccionado y al pulsar el botón **Eliminar** se borrará automáticamente el registro de la base de datos.(Gráfico M1.6)

Gráfico M1.6 Pantalla de eliminación de datos

eliminacanton.php

La siguiente opción posible es la consulta de datos en la que utilizando el mismo procedimiento de las opciones anteriores permitirá escoger el registro a ser consultado.(Gráfico M1.7)

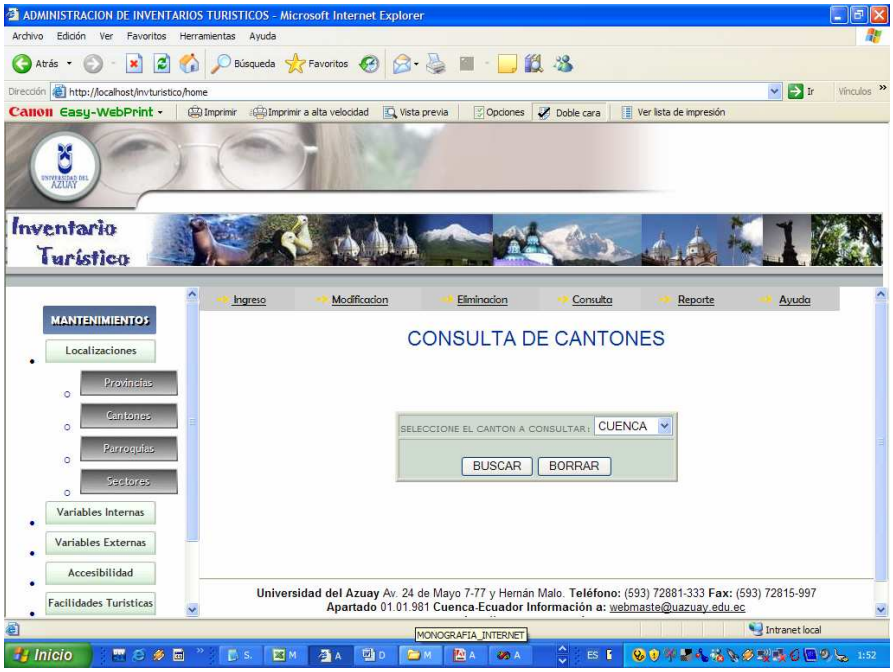


Gráfico M1.7 Pantalla de consulta de datos

consultacantones.php

Y de la misma forma se visualizará los datos correspondientes al registro escogido.(Gráfico M1.8)

Gráfico M1.8 Pantalla de consulta de datos

consultacantones.php

La última opción de manipulación de datos es el reporte, en esta pantalla se mostrará una lista de todos los registros ingresados hasta el momento junto con sus correspondientes datos, permitiendo una vista general de la tabla.(Gráfico M1.9.)

Gráfico M1.9 Pantalla de reporte de datos

reportecanton.php

Para el caso en que la clave pertenezca a un tipo **estudiante** se visualizará la pantalla principal de usuarios. En esta pantalla se brindará al estudiante la posibilidad de realizar el ingreso de la ficha correspondiente al ingreso de atractivos turísticos, además una serie de opciones relacionadas con los mismos que le permitirán conocer el volumen de información existente y realizar consultas y obtener reportes según diversos temas.(Gráfico M1.10)

Gráfico M1.10 Pantalla principal del usuario homeusuario.html

Al presionar la opción **Fichas** del menú izquierdo de la pantalla se desplegará un conjunto de pestañas que conforman una ficha de ingreso de atractivos turísticos clasificados de acuerdo a su temática.

Dentro de lo que son las **Localizaciones** el usuario registrará información referente a ubicación física y geográfica del atractivo con posibilidad de cargar una imagen que identificará al mismo.

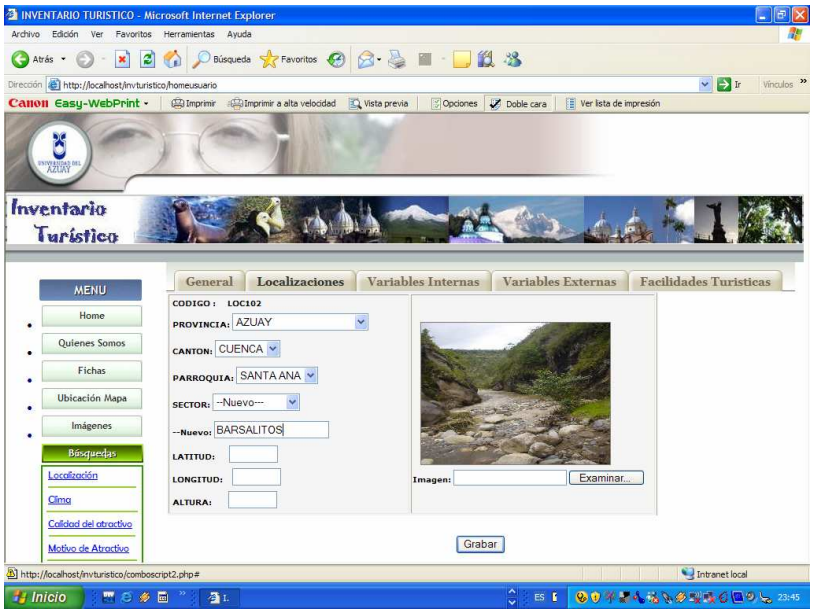


Gráfico M1.11 Ficha_Localizaciones

ficha.php

Dentro de lo que es la opción **General** se solicitará en primera instancia información que identifique al atractivo, así como la valoración correspondiente de acuerdo a valores preestablecidos, con esta información se le dará una calificación y una jerarquía al atractivo.(Gráfico M1.12)

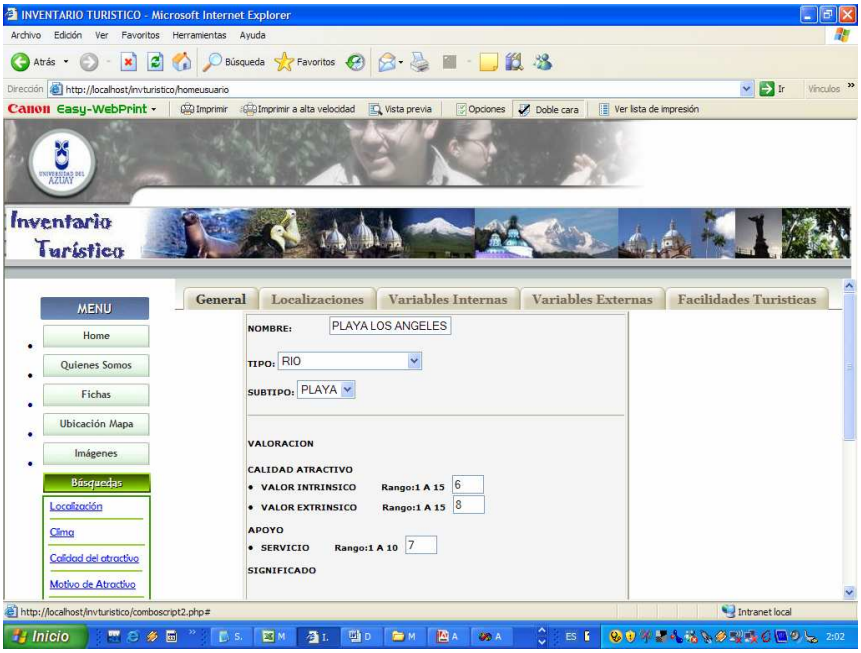


Gráfico M1.12 Ficha_General

ficha.php

La siguiente opción es el conjunto de Variable Internas en la que se solicitará el ingreso de aspectos físicos que rodean al atractivo, los cuales influirán para asignarle una calificación final. (Gráfico M1.13)

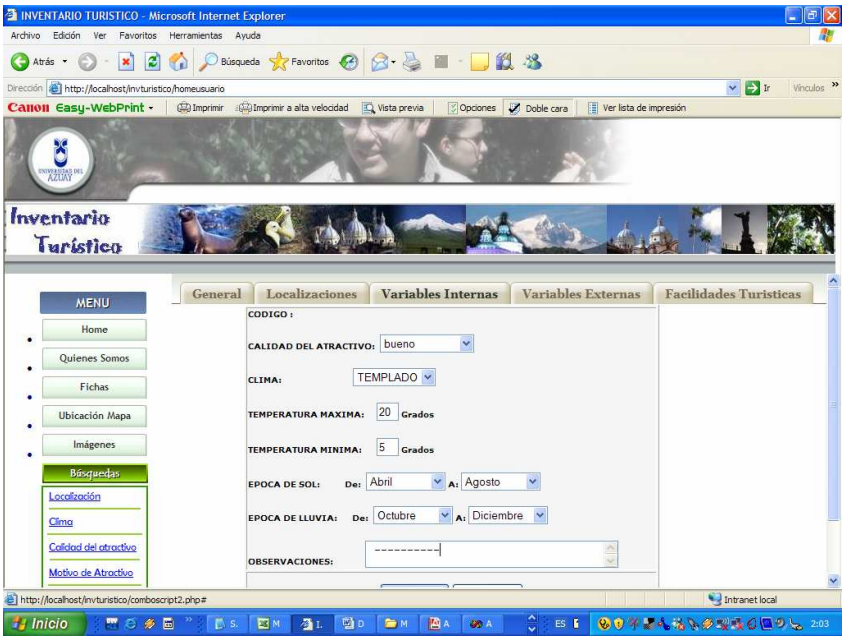


Gráfico M1.12 Ficha_VariablesInternas ficha.php

Dentro de las Variables Externas estará información relativa al espacio físico en el que se encuentra el sito como estado, infraestructura, comunicación, etc.(Gráfico M1.13)

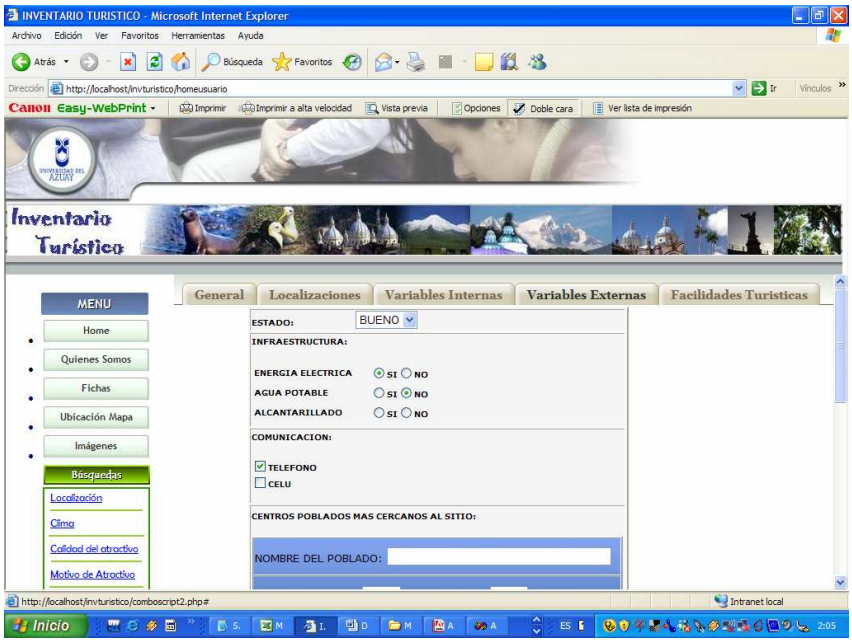


Gráfico M1.13 Ficha_VariablesExternas ficha.php

La última opción es la de Facilidades Turísticas en la que claramente se puede ver el tipo de información que se solicita ya que constituye uno de los principales motivos por los que un lugar se considera atractivo turístico, tal información tiene que ver directamente con la alimentación y alojamiento de los visitantes.(Gráfico M1.14)

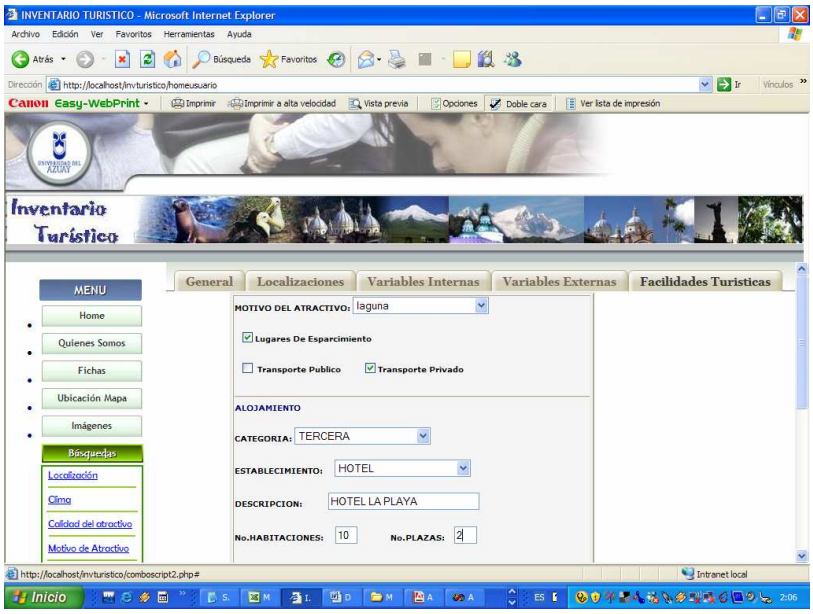


Gráfico M1.14 Ficha_FacilidadesTurísticas

ficha.php

Como se puede ver es muy poca la información que el usuario ingresa, se brinda la facilidad de que simplemente seleccione entre algunas opciones cada dato requerido.

Finalmente se pueden obtener una variedad de reportes clasificados según temas específicos de toda la información ingresada por los estudiantes. Esta información ayudará a elaborar proyectos futuros partiendo de información ya procesada y seleccionada en tiempo real.(Gráfico M1.15)

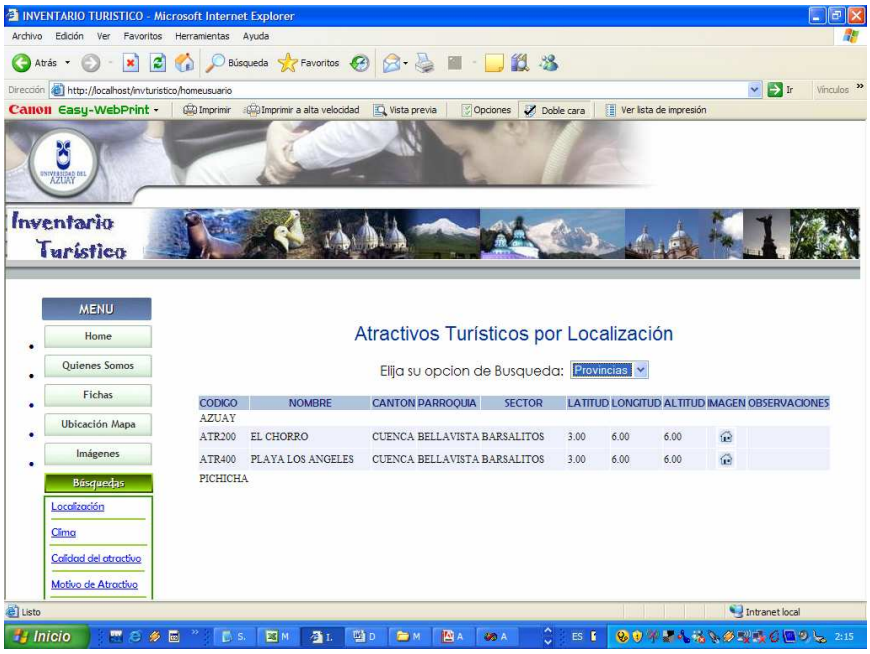


Gráfico M1.15 Reportes

reporte_localización.php

4.3 Implementación del manual en la aplicación Web

El presente manual estará al alcance de los usuarios ya que se accederá a el a través de un enlace situado en el menú principal de la página.

Capítulo 5
Conclusiones y Bibliografía

5.1 Conclusiones

Al concluir este proyecto podemos destacar que el registro de un Inventario Turístico mediante la aplicación Web en tiempo real cuenta con mayor facilidad de uso, mayor objetividad, rapidez y seguridad, para todos los usuarios que hacen uso de esta información.

Se garantiza la integridad de los datos ingresados, ya que en esta aplicación se cuenta con la debida validación de los mismos, evitando de esta manera ingresar información inconsistente. Obteniendo resultados aptos para poder fomentar y realizar Inventarios Turísticos que sirvan de base para la realización de productos turísticos que puedan difundirse con mayor veracidad.

Las herramientas utilizadas en esta aplicación nos permiten tener la posibilidad de expansión según las necesidades que se requieran involucradas al tema.

5.2 Recomendaciones

Para el correcto funcionamiento de la Aplicación Web se debe hacer referencia a la base de datos en MySql con el nombre: “bdituristico”, y tener en cuenta que la creación de todas las tablas pertenecientes a la misma deben ser en minúscula, ya que si existe alguna diferencia de sintaxis en el nombre de las tablas estas no se reconocerán adecuadamente.

Además como primer paso para la utilización de la aplicación Web se debe establecer los parámetros a ser manejados para la generación de códigos, contadores, etc.

También se debe tener en cuenta la precaución al crear nuevos usuarios, ya que el administrador será quien asigne los tipos de usuarios, siendo estos “Administrador”; quien puede modificar, eliminar, consultar , cualquier tipo de datos , y “Alumno”, quien será el que ingrese la información de las fichas, y acceda a cualquier tipo de reporte con la información existente sobre el Inventario Turístico.

5.3 Bibliografía

FRITZ SCHNEIDER Thomas Powel "JAVASCRIPT" Manual de Referencia
Mc Graw Hill
Osborne Media
McGRaw-Hill Interamericana de España, S.A.U

INTERNET

1) es.wikipedia.org/wiki/Aplicacion_web

Autor: Wikipedia
Fecha de Ingreso: 05/01/2007

2) www.WebEstilo.com/mysql/

Autor : Joaquin Gracia Murugarren

Fecha de Ingreso: 15/01/2007

3) www.esepestudio.com/tutoriales

Autor: José Manuel Pérez

Fecha de Ingreso: 15/01/2007

4) http://es.wikipedia.org/wiki/MySQL

Categoría: Sistemas de gestión de bases de datos libres

Autor: Wikipedia

Fecha de Ingreso: 05/01/2007

5) www.fabforce.net

Fabulous Force Database Tools

Autor : Fabforce

Fecha de Ingreso: 05/01/2007

6) http://www.maestrosdelweb.com/editorial/phpmysqlap/

Autor: Fernando Atanasio Negrete

Fecha de Ingreso: 20/01/2007

7) http://www.webestilo.com/php/php08a.phtml

Autor: Carlos Gallús Lahoz.
C.E.S. Fundación San Valero.

Fecha de Ingreso: 15/01/2007

8) http://www.desarrolloweb.com/manuales/

Autor: desarrolloweb.net

Fecha de Ingreso: 25/01/2007

9) http://www.emagister.com/php-mysql-ts.htm

Fecha de Ingreso: 25/01/2007

10) www.elmercurio.com.ec

Autor : Diario El Mercurio

Fecha de Ingreso: 02/02/2007

11) www.mcd.gob.gt

Autor : Lizbeth Barrientos

Fecha de Ingreso: 02/02/2007

12) http://www.uaim.edu.mx/web-carreras/carreras/turismo/empresarial/Cuarto%20Trimestre/TURISMO.pdf.

Autor: uaim.edu.mx

Fecha de Ingreso: 02/02/2007

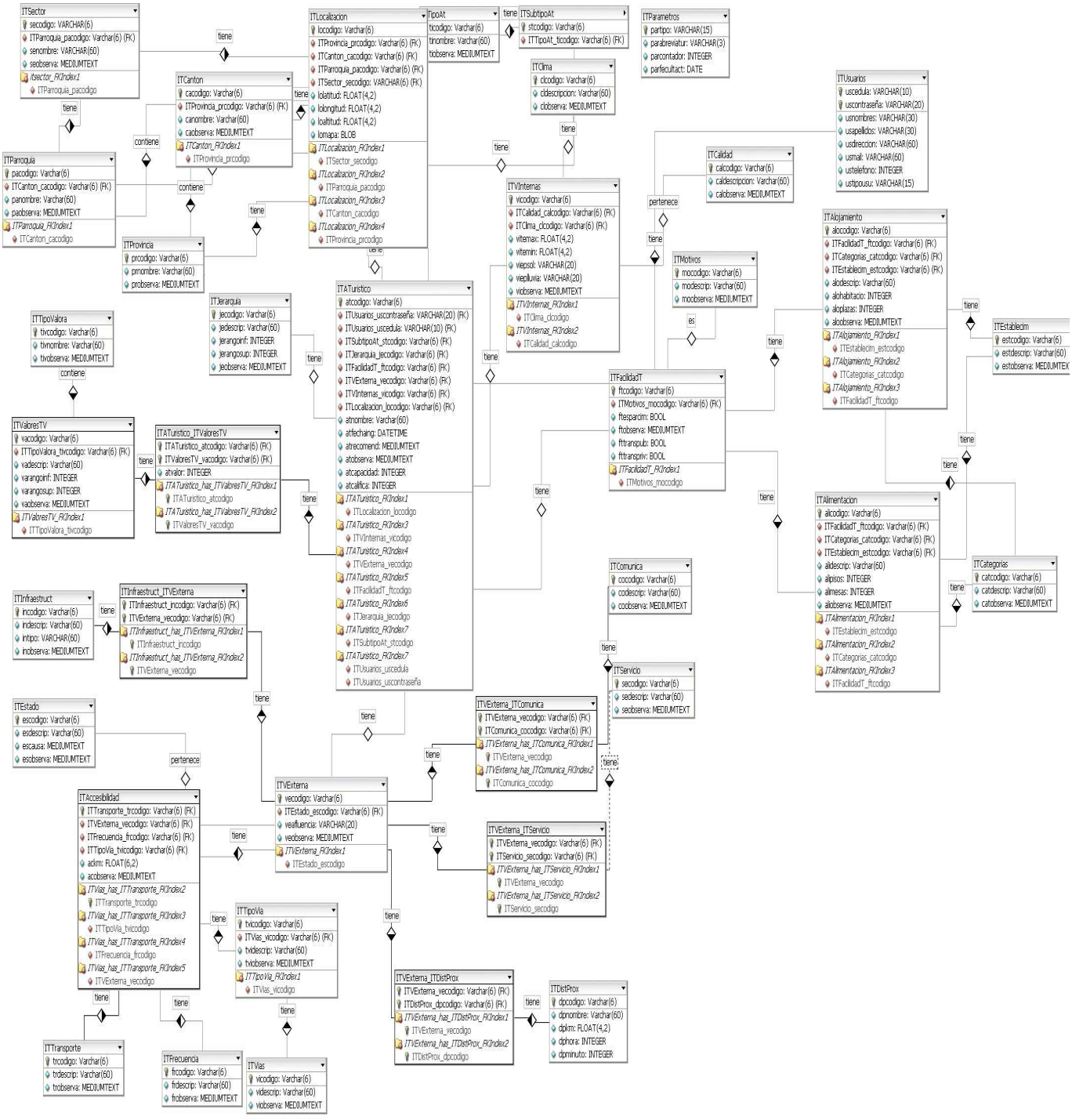
5.4 ANEXOS

ANEXO 1

FICHAS DE INVENTARIOS TURISTICO

ANEXO 2

MODELO ENTIDAD-RELACION



ANEXO 3

DICCIONARIO DE DATOS

ITAccesibilidad

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
ITTransporte_trcodigo	**Varchar(6)**	PK	NN				

ITVExterna_vecodigo	Varchar(6)	NN	
ITFrecuencia_frcodigo	Varchar(6)	NN	
ITTipoVia_tvicodigo	Varchar(6)	NN	
ackm	FLOAT(6,2)		
acobserva	MEDIUMTEXT		

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ITTransporte_trcodigo
ITVias_has_ITTransporte_FKIndex2	Index	ITTransporte_trcodigo
ITVias_has_ITTransporte_FKIndex3	Index	ITTipoVia_tvicodigo
ITVias_has_ITTransporte_FKIndex4	Index	ITFrecuencia_frcodigo
ITVias_has_ITTransporte_FKIndex5	Index	ITVExterna_vecodigo

ITAlimentacion

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
alicodigo	**Varchar(6)**	PK	NN			Codigo del lugar de alimentacion	
ITFacilidadT_ftcodigo	Varchar(6)		NN				
ITCategorias_catcodigo	Varchar(6)		NN				
ITEstablecim_estcodigo	Varchar(6)		NN				
alidescrip	Varchar(60)					Descripcion del Lugar	
alipisos	INTEGER			UNSIGNED		Numero de pisos existentes	
alimesas	INTEGER			UNSIGNED		Numero de mesas existentes	
aliobserva	MEDIUMTEXT					Observaciones o comentarios de la alimentacion en el atractivo	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	alicodigo
ITAlimentacion_FKIndex1	Index	ITEstablecim_estcodigo
ITAlimentacion_FKIndex2	Index	ITCategorias_catcodigo
ITAlimentacion_FKIndex3	Index	ITFacilidadT_ftcodigo

ITAlojamiento

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
alocodigo	**Varchar(6)**	PK	NN			Codigo de alojamiento	
ITFacilidadT_ftcodigo	Varchar(6)		NN				
ITCategorias_catcodigo	Varchar(6)		NN				
ITEstablecim_estcodigo	Varchar(6)		NN				
alodescrip	Varchar(60)					Descripcion del alojamiento	
alohabitacio	INTEGER			UNSIGNED		Numero de habitaciones	
aloplazas	INTEGER			UNSIGNED		numero de plazas	
aloobserva	MEDIUMTEXT					Observaciones o comentarios del alojamiento	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	alocodigo
ITAlojamiento_FKIndex1	Index	ITEstablecim_estcodigo
ITAlojamiento_FKIndex2	Index	ITCategorias_catcodigo
ITAlojamiento_FKIndex3	Index	ITFacilidadT_ftcodigo

ITATuristico

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
atcodigo	**Varchar(6)**	PK	NN			Codigo del atractivo turistico	
ITUsuarios_uscontraseña	VARCHAR(20)		NN				
ITUsuarios_uscedula	VARCHAR(10)		NN				
ITSubtipoAt_stcodigo	Varchar(6)		NN				
ITJerarquia_jecodigo	Varchar(6)		NN				
ITFacilidadT_ftcodigo	Varchar(6)		NN				
ITVExterna_vecodigo	Varchar(6)		NN				
ITVInternas_vicodigo	Varchar(6)		NN				
ITLocalizacion_locodigo	Varchar(6)		NN				
atnombre	Varchar(60)		NN			Nombre del Atractivo Turistico	
atfechaing	DATETIME		NN			Fecha de Ingreso de la informacion del atractivo	
atrecomend	MEDIUMTEXT					Recomendaciones acerca del atractivo	
atobserva	MEDIUMTEXT					Observaciones o comentarios del atractivo visitado	
atcapacidad	INTEGER			UNSIGNED			
atcalifica	INTEGER			UNSIGNED			

IndexName	IndexType	Columns
PRIMARY	PRIMARY	atcodigo
ITATuristico_FKIndex1	Index	ITLocalizacion_locodigo
ITATuristico_FKIndex3	Index	ITVInternas_vicodigo
ITATuristico_FKIndex4	Index	ITVExterna_vecodigo
ITATuristico_FKIndex5	Index	ITFacilidadT_ftcodigo
ITATuristico_FKIndex6	Index	ITJerarquia_jecodigo
ITATuristico_FKIndex7	Index	ITSubtipoAt_stcodigo
ITATuristico_FKIndex7	Index	ITUsuarios_uscedula ITUsuarios_uscontraseña

ITATuristico_ITValoresTV

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
ITATuristico_atcodigo	**Varchar(6)**	PK	NN				
ITValoresTV_vacodigo	**Varchar(6)**	PK	NN				
atvalor	INTEGER			UNSIGNED			

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ITATuristico_atcodigo ITValoresTV_vacodigo
ITATuristico_has_ITValoresTV_FKIndex1	Index	ITATuristico_atcodigo
ITATuristico_has_ITValoresTV_FKIndex2	Index	ITValoresTV_vacodigo

ITCalidad

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
calcodigo	**Varchar(6)**	PK	NN			Codigo de tipo de calidad de un atractivo	
caldescripcion	Varchar(60)		NN			Descripcion del tipo de calidad del atractivo	
calobserva	MEDIUMTEXT					Observaciones o comentarios del tipo de	

	calidad

IndexName	IndexType	Columns
PRIMARY	PRIMARY	calcodigo

ITCanton

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
cacodigo	**Varchar(6)**	PK	NN			Codigo de Canton	
ITProvincia_prcodigo	Varchar(6)		NN				
canombre	Varchar(60)		NN			Nombre del Canton	
caobserva	MEDIUMTEXT					Observaciones, Comentarios del Canton	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	cacodigo
ITCanton_FKIndex1	Index	ITProvincia_prcodigo

ITCategorias

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
catcodigo	**Varchar(6)**	PK	NN			Codigo de la categoria de alimentacion	
catdescrip	Varchar(60)					Descripcion de la categoria	
catobserva	MEDIUMTEXT					Observaciones o comentarios de la categoria	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	catcodigo

ITClima

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
clcodigo	**Varchar(6)**	PK	NN			Codigo del Clima	
cldescripcion	Varchar(60)		NN			Descripcion del clima	
clobserva	MEDIUMTEXT					Observaciones o comentarios del clima	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	clcodigo

ITComunica

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
cocodigo	**Varchar(6)**	PK	NN			Codigo de comunicacion existente en el atractivo	
codescrip	Varchar(60)					Descripcion del tipo de comunicacion	
coobserva	MEDIUMTEXT					Observaciones o comentarios del tipo de comunicacion	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	cocodigo

ITDistProx

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
dpcodigo	**Varchar(6)**	PK	NN				
dpnombre	Varchar(60)						

dpkm	FLOAT(4,2)		
dphora	INTEGER	UNSIGNED	
dpminuto	INTEGER	UNSIGNED	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	dpcodigo

ITEstablecim

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
estcodigo	**Varchar(6)**	PK	NN			Codigo de la categoria de alojamiento	
estdescrip	Varchar(60)					Descripcion de la categoria de alojamiento	
estobserva	MEDIUMTEXT					Observaciones o comentarios de la categoria de alojamiento	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	estcodigo

ITEstado

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
escodigo	**Varchar(6)**	PK	NN			Codigo del Estado en el que se encuentra el atractivo turistico	
esdescrip	Varchar(60)		NN			Descripcion del Estado	
escausa	MEDIUMTEXT					Causa por la que se encuentra en dicho estado	
esobserva	MEDIUMTEXT					Observaciones o comentarios del estado en el que se encuentra el atractivo turistico	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	escodigo

ITFacilidadT

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
ftcodigo	**Varchar(6)**	PK	NN			Codigo del conjunto de facilidades turisticas	
ITMotivos_mocodigo	Varchar(6)		NN				
ftesparcim	BOOL		NN			Existe areas de esparcimiento	
ftobserva	MEDIUMTEXT					Observaciones o comentarios de las facilidades turisticas	
fttranspub	BOOL		NN			Existe transporte publico	
fttranspriv	BOOL		NN			Existe transporte privado	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ftcodigo
ITFacilidadT_FKIndex1	Index	ITMotivos_mocodigo

ITFrecuencia

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc

frcodigo	**Varchar(6)**	PK	NN	Codigo de la frecuencia con al que se accede al atractivo
frdescrip	Varchar(60)		NN	Descripcion de la frecuencia (anual,mensual,semanal,diaria,etc)
frobserva	MEDIUMTEXT			Observaciones o comentarios de la frecuencia con la que se accede al atractivo

IndexName	IndexType	Columns
PRIMARY	PRIMARY	frcodigo

ITInfraestruct

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
incodigo	**Varchar(6)**	PK	NN			Codigo de Infraestructura	
indescrip	Varchar(60)		NN			Descripcionde la Infraestructura	
intipo	VARCHAR(60)		NN			Tipo de Infraestructura	
inobserva	MEDIUMTEXT					Observaciones o comentarios de la infraestructura	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	incodigo

ITInfraestruct_ITVExterna

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
ITInfraestruct_incodigo	**Varchar(6)**	PK	NN				
ITVExterna_vecodigo	**Varchar(6)**	PK	NN				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ITInfraestruct_incodigo ITVExterna_vecodigo
ITInfraestruct_has_ITVExterna_FKIndex1	Index	ITInfraestruct_incodigo
ITInfraestruct_has_ITVExterna_FKIndex2	Index	ITVExterna_vecodigo

ITJerarquia

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
jecodigo	**Varchar(6)**	PK	NN			Codigo de Nivel de Jerarquia del Atractivo	
jedescrip	Varchar(60)					Descripcion del nivel	
jerangoinf	INTEGER			UNSIGNED		Rango Inferior	
jerangosup	INTEGER			UNSIGNED		Rango Superior	
jeobserva	MEDIUMTEXT						

IndexName	IndexType	Columns
PRIMARY	PRIMARY	jecodigo

ITLocalizacion

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
locodigo	**Varchar(6)**	PK	NN			Codigo de la Localizacion	
ITProvincia_prcodigo	Varchar(6)		NN				
ITCanton_cacodigo	Varchar(6)		NN				
ITParroquia_pacodigo	Varchar(6)		NN				
ITSector_secodigo	VARCHAR(6)		NN				
lolatitud	FLOAT(4,2)		NN			Coordenadas de Latitud	

lolongitud	FLOAT(4,2)	NN	Coordenadas de Longitud
loaltitud	FLOAT(4,2)	NN	Coordenadas de altura
lomapa	BLOB		Nombre de la imagen del mapa a ser cargado

IndexName	IndexType	Columns
PRIMARY	PRIMARY	locodigo
ITLocalizacion_FKIndex1	Index	ITSector_secodigo
ITLocalizacion_FKIndex2	Index	ITParroquia_pacodigo
ITLocalizacion_FKIndex3	Index	ITCanton_cacodigo
ITLocalizacion_FKIndex4	Index	ITProvincia_prcodigo

ITMotivos

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
mocodigo	**Varchar(6)**	PK	NN			Codigo del motivo del atractivo	
modescrip	Varchar(60)		NN			Descripcion del motivo del atractivo	
moobserva	MEDIUMTEXT					Observaciones o comentarios del motivo del atractivo	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	mocodigo

ITParametros

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
partipo	**VARCHAR(15)**	PK	NN				
parabreviatur	VARCHAR(3)						
parcontador	INTEGER			UNSIGNED			
parfecultact	DATE						

IndexName	IndexType	Columns
PRIMARY	PRIMARY	partipo

ITParroquia

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
pacodigo	**Varchar(6)**	PK	NN			Codigo de Parroquia	
ITCanton_cacodigo	Varchar(6)		NN				
panombre	Varchar(60)		NN			Nombre de la Parroquia	
paobserva	MEDIUMTEXT					Observaciones, Comentarios de la parroquia	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	pacodigo
ITParroquia_FKIndex1	Index	ITCanton_cacodigo

ITProvincia

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
prcodigo	**Varchar(6)**	PK	NN			Codigo de Provincia	
prnombre	Varchar(60)		NN			Nombre de Provincia	
probserva	MEDIUMTEXT					Observaciones, Comentarios de Provincia	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	prcodigo

ITSector

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
secodigo	**VARCHAR(6)**	PK	NN				
ITParroquia_pacodigo	Varchar(6)		NN				
senombre	VARCHAR(60)		NN				
seobserva	MEDIUMTEXT						

IndexName	IndexType	Columns
PRIMARY	PRIMARY	secodigo
itsector_FKIndex1	Index	ITParroquia_pacodigo

ITServicio

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
secodigo	**Varchar(6)**	PK	NN				
sedescrip	Varchar(60)						
seobserva	MEDIUMTEXT						

IndexName	IndexType	Columns
PRIMARY	PRIMARY	secodigo

ITSubtipoAt

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
stcodigo	**Varchar(6)**	PK	NN			Codigo del Subtipo de atarctivo	
ITTipoAt_ticodigo	Varchar(6)		NN				
stnombre	Varchar(60)		NN			Descripcion del subtipo de atractivo	
stobserva	MEDIUMTEXT					Observaciones o comentarios del atractivo	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	stcodigo
ITSubtipoAt_FKIndex1	Index	ITTipoAt_ticodigo

ITTipoAt

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
ticodigo	**Varchar(6)**	PK	NN			Codigo del tipo de atractivo	
tinombre	Varchar(60)		NN			Descripcion del tipo de atractivo	
tiobserva	MEDIUMTEXT					Observaciones o comentarios del tipo de atractivo	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ticodigo

ITTipoValora

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
tivcodigo	**Varchar(6)**	PK	NN			Codigo de tipo de valoracion de calidad del atractivo	
tivnombre	Varchar(60)					Descripcion del tipo de valoracion de la calidad del atrctivo	
tivobserva	MEDIUMTEXT					Observaciones o comentarios del tipo de valoracion de la calidad del atractivo	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	tivcodigo

ITTipoVia

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
tvicodigo	**Varchar(6)**	PK	NN			Codigo del tipo de via	
ITVias_vicodigo	Varchar(6)		NN				
tvidescrip	Varchar(60)		NN			Descripcion del tipo de via	
tviobserva	MEDIUMTEXT					Observaciones o comentarios del tipo de via	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	tvicodigo
ITTipoVia_FKIndex1	Index	ITVias_vicodigo

ITTransporte

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
trcodigo	**Varchar(6)**	PK	NN			Codigo del tipo de transporte	
trdescrip	Varchar(60)		NN			Descripcion del tipo de transporte	
trobserva	MEDIUMTEXT					Observaciones o comentarios del tipo de transporte	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	trcodigo

ITUsuarios

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
uscedula	**VARCHAR(10)**	PK	NN				
uscontraseña	**VARCHAR(20)**	PK	NN				
usnombres	VARCHAR(30)		NN				
usapellidos	VARCHAR(30)						
usdireccion	VARCHAR(60)						
usmail	VARCHAR(60)						
ustelefono	INTEGER			UNSIGNED			
ustipousu	VARCHAR(15)						

IndexName	IndexType	Columns
PRIMARY	PRIMARY	uscedula uscontraseña

ITValoresTV

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
vacodigo	**Varchar(6)**	PK	NN			Codigo de Valor del tipo	
ITTipoValora_tivcodigo	Varchar(6)		NN				
vadescrip	Varchar(60)					Descripcion del Valor del tipo de valoracion de calidad del atractivo	
varangoinf	INTEGER			UNSIGNED		Rango Inferior de la Valoracion del tipo de valoracion de la calidad del atractivo	

varangosup	INTEGER		UNSIGNED	Rango Superior de la Valoracion del tipo de valoracion de la calidad del atractivo
vaobserva	MEDIUMTEXT			Observaciones o Comentarios del valor del tipo de vaoracion de la calidad del atractivo

IndexName	IndexType	Columns
PRIMARY	PRIMARY	vacodigo
ITValoresTV_FKIndex1	Index	ITTipoValora_tivcodigo

ITVExterna

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
vecodigo	**Varchar(6)**	PK	NN			Codigo del conjunto de variables externas	
ITEstado_escodigo	Varchar(6)		NN				
veafluencia	VARCHAR(20)					Frecuencia de Afluencia Turistica	
veobserva	MEDIUMTEXT					Observaciones o comentarios de las varibles externas del atractivo turistico	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	vecodigo
ITVExterna_FKIndex1	Index	ITEstado_escodigo

ITVExterna_ITComunica

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
ITVExterna_vecodigo	**Varchar(6)**	PK	NN				
ITComunica_cocodigo	**Varchar(6)**	PK	NN				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ITVExterna_vecodigo ITComunica_cocodigo
ITVExterna_has_ITComunica_FKIndex1	Index	ITVExterna_vecodigo
ITVExterna_has_ITComunica_FKIndex2	Index	ITComunica_cocodigo

ITVExterna_ITDistProx

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
ITVExterna_vecodigo	**Varchar(6)**	PK	NN				
ITDistProx_dpcodigo	**Varchar(6)**	PK	NN				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ITVExterna_vecodigo ITDistProx_dpcodigo
ITVExterna_has_ITDistProx_FKIndex1	Index	ITVExterna_vecodigo
ITVExterna_has_ITDistProx_FKIndex2	Index	ITDistProx_dpcodigo

ITVExterna_ITServicio

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
ITVExterna_vecodigo	**Varchar(6)**	PK	NN				
ITServicio_secodigo	**Varchar(6)**	PK	NN				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	ITVExterna_vecodigo ITServicio_secodigo
ITVExterna_has_ITServicio_FKIndex1	Index	ITVExterna_vecodigo
ITVExterna_has_ITServicio_FKIndex2	Index	ITServicio_secodigo

ITVias

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
vicodigo	**Varchar(6)**	PK	NN			Codigo de la via	
videscrip	Varchar(60)		NN			Descripcion de la via	
viobserva	MEDIUMTEXT					Observaciones o comentarios de la via	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	vicodigo

ITVInternas

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
vicodigo	**Varchar(6)**	PK	NN			codigo del conjunto de variables internas del atractivo	
ITCalidad_calcodigo	Varchar(6)		NN				
ITClima_clcodigo	Varchar(6)		NN				
vitemax	FLOAT(4,2)		NN			Temperatura maxima del atractivo	
vitemin	FLOAT(4,2)		NN			Temperatura minima del atractivo	
viepsol	VARCHAR(20)					Meses en los que se tiene epoca de sol	
vieplluvia	VARCHAR(20)					Meses en los que se tiene epoca de lluvia	
viobserva	MEDIUMTEXT					Observaciones o comentarios del atractivo	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	vicodigo
ITVInternas_FKIndex1	Index	ITClima_clcodigo
ITVInternas_FKIndex2	Index	ITCalidad_calcodigo

ANEXO 4

SCRIPT CREACION DE TABLAS DE BASE DE DATOS

```
create table itaccesibilidad (
  ittransporte_trcodigo varchar(6) not null,
  itvexterna_vecodigo varchar(6) not null,
  itfrecuencia_frcodigo varchar(6) not null,
  ittipovia_tvicodigo varchar(6) not null,
  ackm float(6,2) null,
  acobserva mediumtext null,
  primary key(ittransporte_trcodigo),
  index itvias_has_ittransporte_fkindex2(ittransporte_trcodigo),
  index itvias_has_ittransporte_fkindex3(ittipovia_tvicodigo),
  index itvias_has_ittransporte_fkindex4(itfrecuencia_frcodigo),
  index itvias_has_ittransporte_fkindex5(itvexterna_vecodigo)
);

create table italimentacion (
  alicodigo varchar(6) not null,
  itfacilidadt_ftcodigo varchar(6) not null,
  itcategorias_catcodigo varchar(6) not null,
  itestablecim_estcodigo varchar(6) not null,
  alidescrip varchar(60) null,
  alipisos integer unsigned null,
  alimesas integer unsigned null,
  aliobserva mediumtext null,
  primary key(alicodigo),
  index italimentacion_fkindex1(itestablecim_estcodigo),
  index italimentacion_fkindex2(itcategorias_catcodigo),
  index italimentacion_fkindex3(itfacilidadt_ftcodigo)
);

create table italojamiento (
  alocodigo varchar(6) not null,
  itfacilidadt_ftcodigo varchar(6) not null,
  itcategorias_catcodigo varchar(6) not null,
  itestablecim_estcodigo varchar(6) not null,
  alodescrip varchar(60) null,
  alohabitacio integer unsigned null,
  aloplazas integer unsigned null,
  aloobserva mediumtext null,
  primary key(alocodigo),
  index italojamiento_fkindex1(itestablecim_estcodigo),
  index italojamiento_fkindex2(itcategorias_catcodigo),
  index italojamiento_fkindex3(itfacilidadt_ftcodigo)
);

create table itaturistico (
  atcodigo varchar(6) not null,
  itusuarios_uscontraseña varchar(20) not null,
  itusuarios_uscedula varchar(10) not null,
  itsubtipoat_stcodigo varchar(6) not null,
  itjerarquia_jecodigo varchar(6) not null,
```

```
  itfacilidadt_ftcodigo varchar(6) not null,
  itvexterna_vecodigo varchar(6) not null,
  itvinternas_vicodigo varchar(6) not null,
  itlocalizacion_locodigo varchar(6) not null,
  atnombre varchar(60) not null,
  atfechaing datetime not null,
  atrecomend mediumtext null,
  atobserva mediumtext null,
  atcapacidad integer unsigned null,
  atcalifica integer unsigned null,
  primary key(atcodigo),
  index itaturistico_fkindex1(itlocalizacion_locodigo),
  index itaturistico_fkindex3(itvinternas_vicodigo),
  index itaturistico_fkindex4(itvexterna_vecodigo),
  index itaturistico_fkindex5(itfacilidadt_ftcodigo),
  index itaturistico_fkindex6(itjerarquia_jecodigo),
  index itaturistico_fkindex7(itsubtipoat_stcodigo),
  index itaturistico_fkindex7(itusuarios_uscedula, itusuarios_uscontraseña)
);

create table itaturistico_itvalorestv (
  itaturistico_atcodigo varchar(6) not null,
  itvalorestv_vacodigo varchar(6) not null,
  atvalor integer unsigned null,
  primary key(itaturistico_atcodigo, itvalorestv_vacodigo),
  index itaturistico_has_itvalorestv_fkindex1(itaturistico_atcodigo),
  index itaturistico_has_itvalorestv_fkindex2(itvalorestv_vacodigo)
);

create table itcalidad (
  calcodigo varchar(6) not null,
  caldescripcion varchar(60) not null,
  calobserva mediumtext null,
  primary key(calcodigo)
);

create table itcanton (
  cacodigo varchar(6) not null,
  itprovincia_prcodigo varchar(6) not null,
  canombre varchar(60) not null,
  caobserva mediumtext null,
  primary key(cacodigo),
  index itcanton_fkindex1(itprovincia_prcodigo)
);

create table itcategorias (
  catcodigo varchar(6) not null,
  catdescrip varchar(60) null,
  catobserva mediumtext null,
  primary key(catcodigo)
```

```
);

create table itclima (
  clcodigo varchar(6) not null,
  cldescripcion varchar(60) not null,
  clobserva mediumtext null,
  primary key(clcodigo)
);

create table itcomunica (
  cocodigo varchar(6) not null,
  codescrip varchar(60) null,
  coobserva mediumtext null,
  primary key(cocodigo)
);

create table itdistprox (
  dpcodigo varchar(6) not null,
  dpnombre varchar(60) null,
  dpkm float(4,2) null,
  dphora integer unsigned null,
  dpminuto integer unsigned null,
  primary key(dpcodigo)
);

create table itestablecim (
  estcodigo varchar(6) not null,
  estdescrip varchar(60) null,
  estobserva mediumtext null,
  primary key(estcodigo)
);

create table itestado (
  escodigo varchar(6) not null,
  esdescrip varchar(60) not null,
  escausa mediumtext null,
  esobserva mediumtext null,
  primary key(escodigo)
);

create table itfacilidadt (
  ftcodigo varchar(6) not null,
  itmotivos_mocodigo varchar(6) not null,
  ftesparcim bool not null,
  ftobserva mediumtext null,
  fttranspub bool not null,
  fttranspriv bool not null,
  primary key(ftcodigo),
  index itfacilidadt_fkindex1(itmotivos_mocodigo)
);
```

```
create table itfrecuencia (
  frcodigo varchar(6) not null,
  frdescrip varchar(60) not null,
  frobserva mediumtext null,
  primary key(frcodigo)
);

create table itinfraestruct (
  incodigo varchar(6) not null,
  indescrip varchar(60) not null,
  intipo varchar(60) not null,
  inobserva mediumtext null,
  primary key(incodigo)
);

create table itinfraestruct_itvexterna (
  itinfraestruct_incodigo varchar(6) not null,
  itvexterna_vecodigo varchar(6) not null,
  primary key(itinfraestruct_incodigo, itvexterna_vecodigo),
  index itinfraestruct_has_itvexterna_fkindex1(itinfraestruct_incodigo),
  index itinfraestruct_has_itvexterna_fkindex2(itvexterna_vecodigo)
);

create table itjerarquia (
  jecodigo varchar(6) not null,
  jedescrip varchar(60) null,
  jerangoinf integer unsigned null,
  jerangosup integer unsigned null,
  jeobserva mediumtext null,
  primary key(jecodigo)
);

create table itlocalizacion (
  locodigo varchar(6) not null,
  itprovincia_prcodigo varchar(6) not null,
  itcanton_cacodigo varchar(6) not null,
  itparroquia_pacodigo varchar(6) not null,
  itsector_secodigo varchar(6) not null,
  lolatitud float(4,2) not null,
  lolongitud float(4,2) not null,
  loaltitud float(4,2) not null,
  lomapa blob null,
  primary key(locodigo),
  index itlocalizacion_fkindex1(itsector_secodigo),
  index itlocalizacion_fkindex2(itparroquia_pacodigo),
  index itlocalizacion_fkindex3(itcanton_cacodigo),
  index itlocalizacion_fkindex4(itprovincia_prcodigo)
);
```

```
create table itmotivos (
  mocodigo varchar(6) not null,
  modescrip varchar(60) not null,
  moobserva mediumtext null,
  primary key(mocodigo)
);

create table itparametros (
  partipo varchar(15) not null,
  parabreviatur varchar(3) null,
  parcontador integer unsigned null,
  parfecultact date null,
  primary key(partipo)
);

create table itparroquia (
  pacodigo varchar(6) not null,
  itcanton_cacodigo varchar(6) not null,
  panombre varchar(60) not null,
  paobserva mediumtext null,
  primary key(pacodigo),
  index itparroquia_fkindex1(itcanton_cacodigo)
);

create table itprovincia (
  prcodigo varchar(6) not null,
  prnombre varchar(60) not null,
  probserva mediumtext null,
  primary key(prcodigo)
);

create table itsector (
  secodigo varchar(6) not null,
  itparroquia_pacodigo varchar(6) not null,
  senombre varchar(60) not null,
  seobserva mediumtext null,
  primary key(secodigo),
  index itsector_fkindex1(itparroquia_pacodigo)
);

create table itservicio (
  secodigo varchar(6) not null,
  sedescrip varchar(60) null,
  seobserva mediumtext null,
  primary key(secodigo)
);

create table itsubtipoat (
  stcodigo varchar(6) not null,
  ittipoat_ticodigo varchar(6) not null,
```

```
  stnombre varchar(60) not null,
  stobserva mediumtext null,
  primary key(stcodigo),
  index itsubtipoat_fkindex1(ittipoat_ticodigo)
);

create table ittipoat (
  ticodigo varchar(6) not null,
  tinombre varchar(60) not null,
  tiobserva mediumtext null,
  primary key(ticodigo)
);

create table ittipovalora (
  tivcodigo varchar(6) not null,
  tivnombre varchar(60) null,
  tivobserva mediumtext null,
  primary key(tivcodigo)
);

create table ittipovia (
  tvicodigo varchar(6) not null,
  itvias_vicodigo varchar(6) not null,
  tvidescrip varchar(60) not null,
  tviobserva mediumtext null,
  primary key(tvicodigo),
  index ittipovia_fkindex1(itvias_vicodigo)
);

create table ittransporte (
  trcodigo varchar(6) not null,
  trdescrip varchar(60) not null,
  trobserva mediumtext null,
  primary key(trcodigo)
);

create table itusuarios (
  uscedula varchar(10) not null,
  uscontraseña varchar(20) not null,
  usnombres varchar(30) not null,
  usapellidos varchar(30) null,
  usdireccion varchar(60) null,
  usmail varchar(60) null,
  ustelefono integer unsigned null,
  ustipousu varchar(15) null,
  primary key(uscedula, uscontraseña)
);

create table itvalorestv (
  vacodigo varchar(6) not null,
```

```
  ittipovalora_tivcodigo varchar(6) not null,
  vadescrip varchar(60) null,
  varangoinf integer unsigned null,
  varangosup integer unsigned null,
  vaobserva mediumtext null,
  primary key(vacodigo),
  index itvalorestv_fkindex1(ittipovalora_tivcodigo)
);

create table itvexterna (
  vecodigo varchar(6) not null,
  itestado_escodigo varchar(6) not null,
  veafluencia varchar(20) null,
  veobserva mediumtext null,
  primary key(vecodigo),
  index itvexterna_fkindex1(itestado_escodigo)
);

create table itvexterna_itcomunica (
  itvexterna_vecodigo varchar(6) not null,
  itcomunica_cocodigo varchar(6) not null,
  primary key(itvexterna_vecodigo, itcomunica_cocodigo),
  index itvexterna_has_itcomunica_fkindex1(itvexterna_vecodigo),
  index itvexterna_has_itcomunica_fkindex2(itcomunica_cocodigo)
);

create table itvexterna_itdistprox (
  itvexterna_vecodigo varchar(6) not null,
  itdistprox_dpcodigo varchar(6) not null,
  primary key(itvexterna_vecodigo, itdistprox_dpcodigo),
  index itvexterna_has_itdistprox_fkindex1(itvexterna_vecodigo),
  index itvexterna_has_itdistprox_fkindex2(itdistprox_dpcodigo)
);

create table itvexterna_itservicio (
  itvexterna_vecodigo varchar(6) not null,
  itservicio_secodigo varchar(6) not null,
  primary key(itvexterna_vecodigo, itservicio_secodigo),
  index itvexterna_has_itservicio_fkindex1(itvexterna_vecodigo),
  index itvexterna_has_itservicio_fkindex2(itservicio_secodigo)
);

create table itvias (
  vicodigo varchar(6) not null,
  videscrip varchar(60) not null,
  viobserva mediumtext null,
  primary key(vicodigo)
);

create table itvinternas (
```

```
  vicodigo varchar(6) not null,
  itcalidad_calcodigo varchar(6) not null,
  itclima_clcodigo varchar(6) not null,
  vitemax float(4,2) not null,
  vitemin float(4,2) not null,
  viepsol varchar(20) null,
  viepluvia varchar(20) null,
  viobserva mediumtext null,
  primary key(vicodigo),
  index itvinternas_fkindex1(itclima_clcodigo),
  index itvinternas_fkindex2(itcalidad_calcodigo)
);
```