

2 Introducción Teórica

A lo largo de este apartado se va a describir los diferentes programas y aplicaciones que van a ser necesarios para el desarrollo de éste. De esta forma se tendrá una concepción más global tanto del propio proyecto como de su entorno o circunstancias.

Se realizará una descripción de las bases de datos, definiendo qué son, su funcionalidad y uso, de forma que se dé a conocer de una manera más exacta el porqué de su uso en este proyecto.

La comunicación con una base de datos se realiza por medio de un lenguaje de gestión, viendo cuál será el lenguaje utilizado y el porqué de su elección.

Como se ha comentado, el proyecto hace uso de contenido web, es por tanto necesario hacer una introducción del lenguaje utilizado en dicha aplicación web, esto es, lenguaje HTML y PHP.

Relacionado con el apartado anterior se describirán los buscadores web, de esta forma se sabrá el papel que juegan dichos buscadores en la realización de este proyecto.

Se hará por tanto una introducción a todos estos elementos, usados de manera habitual tanto por la mayoría de empresas como por cualquier usuario de sistemas de información.

2.1 Bases de Datos Y SGDB

Una de las principales funciones de los sistemas informáticos es la de almacenar gran cantidad de información de manera perdurable. Pero para ello se necesitan varias cosas, en primer lugar se requiere de un soporte físico adecuado y en siguiente lugar de un orden y control. Debido a estas necesidades surge el concepto de las bases de datos. Se puede definir Base de Datos como 'Almacén de datos relacionados entre sí con diferentes modos de organización'. Una base de datos representa algunos aspectos del mundo real, aquellos que le interesan al diseñador. Se diseña y almacenan datos con un propósito específico. Con la palabra "datos" se hace referencia a hechos conocidos que pueden registrarse, como números telefónicos, direcciones, nombres... Las bases de datos almacenan datos, permitiendo manipularlos fácilmente y mostrarlos de diversas formas. El proceso de construir una base de datos es llamado diseño de base de datos.

Para entender mejor el porqué de la existencia de las Bases de Datos (teniendo en cuenta que dentro de la propia Base de Datos hay un sistema de gestión del que posteriormente se darán más detalles) se analizarán las ventajas y desventajas de las mismas:

Ventajas:

- **Control de redundancia de datos:** En las Bases de Datos a diferencia de los sistemas de ficheros, cuando un fichero es duplicado no se almacenan varias copias de los mismos datos, si no que los ficheros se integran pero sin ocupar espacio extra. Las redundancias en ocasiones son necesarias para modelar relaciones entre datos.
- **Consistencia de los datos:** Con o sin control de redundancia de datos se reduce el riesgo de inconsistencias debido a que si hay datos duplicados el sistema los actualiza a la vez. Y en caso de no haber redundancias los datos al actualizarse ya se encontrarían disponibles para todos los usuarios.
- **Integridad mejorada de datos:** Mediante reglas o restricciones que no se pueden violar, la Base de Datos expresa su integridad, asegurando validez y consistencia en los datos.
- **Mantenimiento de estándares:** La integración permite respetar más fácilmente los estándares necesarios. Pudiéndose establecer un estándar sobre el formato de los datos para facilitar su intercambio.

Estos estándares pueden ser reglas de acceso o procedimientos de actualización.

- **Compartición de datos:** Los datos dentro de la Base de Datos pueden ser compartidos por todos los usuarios autorizados, lográndose así un importante ahorro respecto al espacio físico y un mayor control de la información.
- **Mejora en accesibilidad a los datos:** Las Bases de Datos proporcionan lenguajes de consultas o generadores de informes que facilitan en gran medida el acceso de los usuarios a los datos.
- **Mejora en cuanto a seguridad:** Esto se debe a que las Bases de Datos establecen un control que permite una adecuada gestión de usuarios, grupos y permisos logrando así menor número de vulnerabilidades.
- **Mejora del mantenimiento:** Las Bases de Datos separan las descripciones de los datos de las aplicaciones, por tanto un cambio en la estructura de los datos no tiene porqué suponer cambios importantes en las aplicaciones que gestionan dichos datos, lo cual facilita las labores de mantenimiento.

- **Mejora en la productividad:** La Base de Datos proporciona muchas de las operaciones a bajo nivel que el programador necesitaría realizar en un sistema de ficheros, por tanto el hecho de disponer de estas funciones hace que el programador pueda centrarse en otras funciones específicas requeridas por los usuarios aumentando así su productividad.
- **Aumento de la concurrencia:** Las Bases de Datos gestionan el acceso a los datos de tal manera que permiten que varios usuarios accedan a la misma información sin ocasionar fallos ni interferencias.
- **Mejora en servicios de copia de seguridad:** Las Bases de Datos se encargan de los mecanismos para llevar a cabo las copias de seguridad necesarias para minimizar el trabajo perdido en caso de fallo.

Desventajas:

- **Coste del equipamiento adicional:** Toda Base de Datos, incluyendo el sistema de gestión, definiciones de tablas, relaciones, reglas y demás elementos de control, a parte de los datos en sí, tienen una necesidad tanto de espacio como de procesado, que probablemente implique la necesidad de un mayor equipamiento. Todo esto provocará que la

implantación de un sistema de Base de Datos suponga un coste alto (aunque suficientemente rentable).

- **Aumento de la complejidad:** Las Bases de Datos pueden llegar a ser sistemas muy complejos que es necesario comprender adecuadamente para conseguir una óptima funcionalidad.
- **Vulnerabilidad ante fallos:** El hecho de que todo esté centralizado en una Base de Datos implica que el sistema sea más vulnerable ante fallos.

Las Bases de Datos pueden clasificarse de diferentes maneras, pero la más adecuada, o al menos la más habitual suele ser según el modelo de administración de datos que emplea.

Los diferentes tipos que hay son:

- **Bases de Datos Transaccionales:** Son Bases de Datos cuyo único fin es el envío y recepción de datos a grandes velocidades. Son muy poco comunes.

- **Bases de Datos Jerárquicas:** Almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un *nodo padre* de información puede tener varios *hijos*. El nodo que no tiene padres es llamado *raíz*, y a los nodos que no tienen hijos se los conoce como *hojas*.
- **Bases de Datos de Red:** Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).
- **Bases de Datos Relacionales:** Modelo usado actualmente para los problemas reales y administrar datos dinámicamente. Su idea fundamental es el uso de las relaciones. Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados “tuplas”. Se ve cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las “tuplas”, y campos (las columnas de una tabla). En este modelo, el lugar y la forma en que se almacenen los datos no tienen

relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante “consultas” que ofrecen una amplia flexibilidad y poder para administrar la información.

- **Bases de Datos Orientadas a Objetos:** Son de reciente aparición, son propias de modelos informáticos orientados a objetos. Su peculiaridad es que tratan de almacenar los objetos completos (estado y comportamiento) en la Base de Datos. Este tipo de Base de Datos incorpora todos los conceptos importantes asociados a este tipo de paradigma: Encapsulación, herencia y polimorfismo.

Las Bases de Datos tienen diferentes formas de mantenerse, una es de forma manual (mediante la cual el usuario es el encargado del control), pero la más común es mediante una o más aplicaciones que se encarguen de las gestiones apropiadas. Estas aplicaciones pueden ser aplicaciones que cualquiera diseñe y se encarguen de las tareas específicas de gestión, o pueden ser un conjunto de aplicaciones conocidas como Sistemas Gestores de Bases de Datos o más comúnmente SGDB. Estos SGDB son los

encargados de definir, construir y manipular la Base de Datos, manteniendo en todo momento la integridad y la falta de redundancia de la misma.

Se definen los SGDB (en inglés Data Base Management System) como un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y de un lenguaje de consulta.

Hasta ahora se ha estado hablando de las Bases de Datos incluyendo al Sistema Gestor como parte de un mismo todo, pero a pesar de ser dos partes de un sistema que van unidas inequívocamente, son dos elementos claramente diferenciables.

Dentro de los distintos SGDB se puede hacer una catalogación importante, aquellos que son de libre distribución, y aquellos que tienen un coste de adquisición y uso.

SGDB Libres:

- MySQL
- PostgreSQL
- Firebird
- SQLite
- DB2 Express-C

SGDB no Libres:

- Oracle

- Microsoft SQL Server
- Microsoft Access
- Nexus DB
- ...

2.2 SQL

El usuario o cualquier aplicación se van a comunicar con la base de datos por medio de un lenguaje que tiene que permitir efectuar inserciones, modificaciones o consultas de la base de datos. Para ello se crea el lenguaje estándar de comunicación de bases de datos, conocido como “Structure Query Language” o *SQL*.

El lenguaje *SQL* es un lenguaje normalizado que permite trabajar con cualquier tipo de lenguaje (*PHP*, *ASP* o *JAVA*) en combinación con cualquier tipo de base de datos (*MySQL*, *MS Access*, *SQL Server*...).

Las principales características de este lenguaje van a ser son la potencia de uso y la versatilidad del lenguaje, que se ven contrastadas por su facilidad de aprendizaje, permitiendo una gran variedad de operaciones sobre las bases de datos relacionales.

En este proyecto se va a hacer uso del lenguaje *MySQL*, debido a las ventajas que se enumeran a continuación:

- Acceso a la base de datos de forma simultánea por varios usuarios y/o aplicaciones.
- Seguridad, en forma de permisos y privilegios, determinados usuarios tendrán permiso para o consulta o modificación de determinadas tablas. Esto permite compartir datos sin que peligre la integridad de la base de datos o protegiendo determinados contenidos.
- Potencia: *MySQL* es un lenguaje muy potente para consulta de bases de datos, usar un motor nos ahorra una enorme cantidad de trabajo.
- Portabilidad: *MySQL* es también un lenguaje estandarizado, de modo que las consultas hechas usando *MySQL* son fácilmente portables a otros sistemas y plataformas.
- Escalabilidad: es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros, y hasta 32 índices por tabla.
- *MySQL* está escrito en C y C++ y probado con multitud de compiladores y dispone de APIs (Interfaz

de programación de aplicaciones) para muchas plataformas diferentes.

- Conectividad: es decir, permite conexiones entre diferentes máquinas con distintos sistemas operativos. Es corriente que servidores *Linux* o *Unix*, usando *MySQL*, sirvan datos para ordenadores con *Windows*, *Linux*, *Solaris*, etc. Para ello se usa *TCP/IP*, tuberías o sockets Unix.
- Es multihilo, con lo que puede beneficiarse de sistemas multiprocesador.
- Permite manejar multitud de tipos para columnas.
- Permite manejar registros de longitud fija o variable.

2.3 HTML

HTML – *HyperText Markup Language* (Lenguaje de Marcas de Hipertexto) – es el *lenguaje de marcado* que se utiliza para crear páginas web.

Un *lenguaje de marcado* o *lenguaje de marcas* es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación. En ocasiones, se confunden los lenguajes de marcado con lenguajes de

programación, siendo esto un error ya que los lenguajes de marcado carecen de funciones aritméticas o variables.

En el caso de *HTML*, este es un estándar reconocido en todo el mundo y cuyas normas define un organismo llamado *World Wide Web Consortium*. Al ser un estándar reconocido, una misma página se visualiza de igual forma en los distintos navegadores de cualquier sistema operativo.

HTML se va a usar para describir la estructura y el contenido de la página a modo de texto, pudiendo añadirse otros elementos como imágenes,...

El principal cometido de los documentos *HTML* es albergar texto con formato. Para esto el archivo electrónico almacena tanto los contenidos como la información sobre el formato de esos contenidos.

El proceso de indicar las diferentes partes que componen la información se denomina marcar (*markup* en inglés). Cada una de las palabras empleadas para marcar el inicio y el final de una sección se denominan *etiquetas*.

Aunque existen algunas excepciones, en general las etiquetas se indican por pares y se forman de la siguiente manera:

- Etiqueta de apertura: carácter <, seguido del nombre de la etiqueta (sin espacios en blanco) y terminados con el carácter >

- Etiqueta de cierre: carácter <, seguido del carácter / seguido del nombre de la etiqueta (sin espacios en blanco) y terminado con el carácter >

Las páginas *HTML* se dividen en dos partes: la cabecera y el cuerpo. La cabecera incluye la información sobre la propia página, como por ejemplo su título y su idioma. El cuerpo de la página incluye todos sus contenidos, como párrafos de texto e imágenes.

El cuerpo (llamado *body* en inglés) contiene todo lo que el usuario ve en su pantalla y la cabecera (*head* en inglés) contiene todo lo que no se ve (con la única excepción del título de la página, que los navegadores muestran como título de sus ventanas).

Ejemplo de estructura básica:

```
<html>
  <head>
    <title>Título del documento HTML</title>
  </head>
  <body>
    <b>
      Esto es un párrafo en negrita
    </b>
  </body>
</html>
```

2.4 PHP

Internet, sirve de soporte a una gran cantidad de sistemas de información y comunicaciones que engloban áreas tan importantes como la investigación, el comercio electrónico, la visualización de información, el correo electrónico..

Las páginas web son textos *ASCII* escritos en *HTML* (*Hypertext Markup Language*) como se comentó anteriormente, que se transfieren entre los servidores de *WWW* y los navegadores mediante el protocolo *HTTP* (*Hypertext Transfer Protocol*).

Cuando comenzó la *World Wide Web*, los sitios web ofrecían páginas estáticas, es decir, que a efectos del usuario, el único proceso realizado era el de visualización de sus contenidos por parte del navegador del cliente.

A medida que pasaron los años, aumentó la afluencia de público y surge la necesidad de reunir y procesar las peticiones del cliente con la finalidad de ofrecerle informaciones mejor dirigidas, escogidas y elaboradas.

Es entonces, cuando surgen dos alternativas de diseño web:

Las llamadas páginas en el lado del cliente, programadas en *HTML* dinámico (*DHTML*), que se ejecutan en el navegador y son capaces de cambiar su apariencia. Utilizando *DHTML* se puede conseguir que los objetos aparezcan y desaparezcan, que varíen su

tamaño, que se les asocie posiciones absolutas en pantalla... En definitiva, *DHTML* aumenta las posibilidades de visualización de las instrucciones *HTML*.

Las páginas del lado del servidor, permiten modificar los contenidos antes de enviarlos al cliente. Las primeras fueron *CGI* (*Common Gateway Interface*) que implementan páginas web activas en el servidor, capaces de leer lo que le enviaba un cliente al servidor a través de un formulario, o escribir en la página del cliente.

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

PHP es capaz de realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas programados en un lenguaje distinto al *HTML*. Esto se debe a que *PHP* ofrece un extenso conjunto de funciones para la explotación de bases de datos sin complicaciones.

Es por esto, que levanta un mayor interés con respecto a los lenguajes pensados para los *CGI*.

PHP se puede desplegar en la mayoría de los servidores web, y en casi todos los sistemas operativos sin ningún coste. En este proyecto, va a ser utilizado junto al servidor de bases de datos MySQL, pero va a permitir la conexión con otros tipos de servidores como pueden ser Postgres, Oracle, Microsoft SQL Server, Firebird etc.

En este proyecto se hará uso de la versión 5.3.0, que es la que ofrece el servidor independiente *Wamp* junto la base de datos MySQL y el servidor web Apache. Esto será comentado en posteriores puntos de esta memoria.

La versión PHP 5 va a ofrecer las siguientes ventajas:

- Mejor soporte para la Programación Orientada a Objetos, que en versiones anteriores era extremadamente rudimentario.
- Mejoras de rendimiento
- Mejor soporte para MySQL con extensión completamente reescrita
- Mejor soporte a XML
- Soporte nativo para SQLite
- Soporte integrado para SOAP
- Iterados de datos
- Manejo de excepciones

2.5 CSS

CSS son las siglas de Cascading Style Sheets - Hojas de Estilo en Cascada - que es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla, por impresora, por voz (cuando la información es pronunciada a través de un dispositivo de lectura) o en dispositivos táctiles basados en Braille.

CSS es una especificación desarrollada por el W3C (World Wide Web Consortium) para permitir la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL de la presentación del documento con las hojas de estilo, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra..., modificando así la apariencia de una página web de una forma más sencilla, permitiendo a los desarrolladores controlar el estilo y formato de sus documentos.

El lenguaje CSS se basa en una serie de reglas que rigen el estilo de los elementos en los documentos estructurados, y que forman la sintaxis de las hojas de estilo.

Cada regla consiste en un selector y una declaración, esta última va entre corchetes y consiste en una propiedad o atributo, y un valor separados por dos puntos.

Un ejemplo de CSS:

```
h2 {color: green;}
```

h2 ---> es el selector

{color: green;} ---> es la declaración

- color ---> es la propiedad o atributo
- green ---> es el valor

El *selector* especifica qué elementos van a estar afectados por esa declaración, de manera que hace de enlace entre la estructura del documento y la regla estilística en la hoja de estilo.

La declaración que va entre corchetes es la información de estilo que indica cómo se va a ver el selector. En caso de que haya más de una declaración se usa punto y coma para separarlas.

Dentro de la declaración, la *Propiedad* o *Atributo* define la interpretación del elemento asignándosele un cierto *Valor*, que puede ser color, alineación, tipo de fuente, tamaño..., es decir, especifican qué aspecto del selector se va a cambiar.

2.5.1 Tres tipos de estilos

La información CSS se puede proporcionar por varias fuentes, ya sea adjunto como un documento por separado o incorporado en el documento HTML, y dentro de estas posibilidades destacan tres formas de dar estilo a un documento web:

2.5.1.1 Hoja de Estilo Externa

La *Hoja de Estilo Externa* se almacena en un archivo diferente al del archivo con el código HTML al cual estar vinculado a través del elemento *link*, que debe ir situado en la sección *head*. Es la manera de programar más eficiente, ya que separa completamente las reglas de formato para la página HTML de la estructura básica de la página.

2.5.1.2 Hoja de Estilo Interna

La *Hoja de Estilo Interna* está incorporada a un documento HTML, a través del elemento *style* dentro de la sección *head*, consiguiendo de esta manera separar la información del estilo del código HTML.

2.5.1.3 Estilo en Línea

El *Estilo en Línea* sirve para insertar el lenguaje de estilo directamente dentro de la sección *body* con el elemento *style*. Sin embargo, este tipo de estilo no se recomienda pues se debe intentar siempre separar el contenido de la presentación.

2.6 Aplicaciones de gestión de Bases de Datos

Una base de datos por sí sola no resulta de mucha utilidad si va acompañada de una adecuada aplicación de gestión. Dichas aplicaciones suelen diseñarse a medida de cada base de datos y ajustándose a las necesidades del cliente o empresa. Pero también hay numerosas herramientas de gestión estándar con las que poder explotar las bases de datos adecuadamente.

Se pueden diferenciar entre aplicaciones Web y aplicaciones de escritorio:

- **Aplicaciones Web**

Las aplicaciones Web son aquellas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una Intranet mediante un navegador. Es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

- **phpMyAdmin**

Herramienta elaborada para el manejo de la administración de MySQL, pensada para una interfaz Web, fue desarrollada con código php.

Esta herramienta permite la creación y eliminación de bases de datos, crear, alterar y eliminar tablas, editar, borrar y añadir campos, ejecutar cualquier tipo de sentencia SQL, administrar privilegios, administrar claves en campos y exportar datos en varios formatos. Se encuentra disponible bajo la licencia GPL.

Este proyecto se encuentra vigente desde el año 1998 siendo uno de los mejor valorados por diferentes asociaciones especializadas. Como esta herramienta corre en máquinas con Servidores Web y soporte de PHP y MySQL, la tecnología utilizada ha ido variando durante su desarrollo.

Otras herramientas similares se han creado para dar la misma utilidad pero en diferentes bases de datos, por ejemplo phpMSAdmin para Microsoft SQL Server, o phpPgAdmin para PostgreSQL.

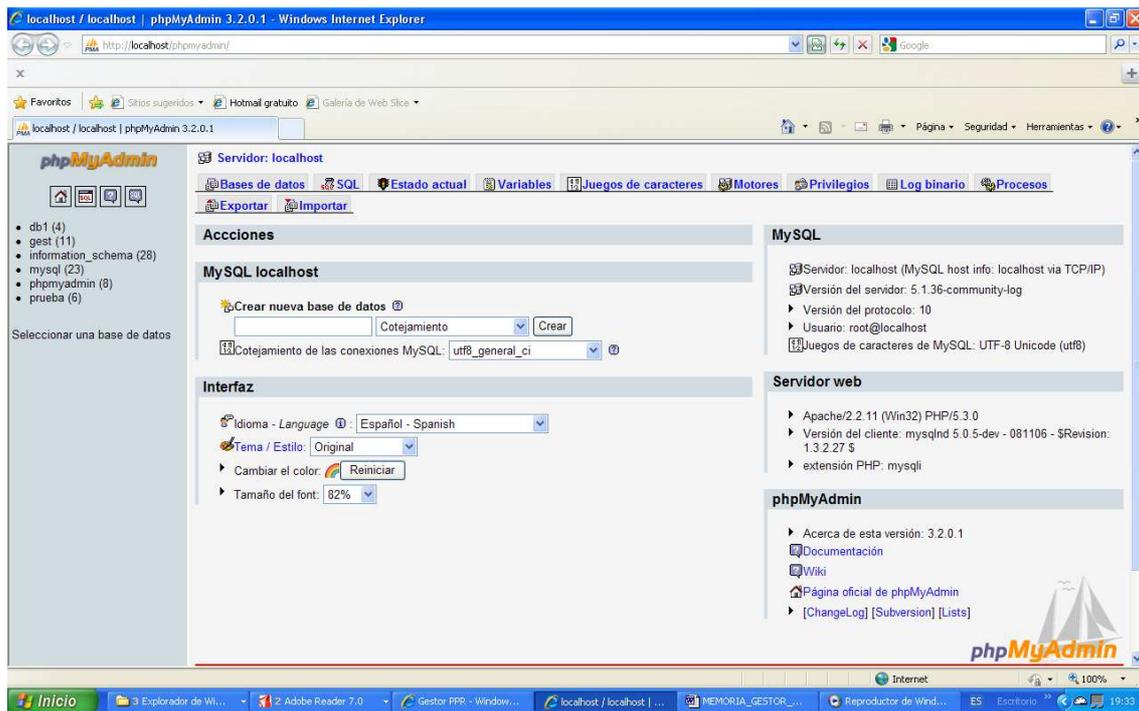


Figura 2.6.1 1

Ejemplo de la aplicación web phpMyAdmin

• **Aplicaciones de escritorio**

Este tipo de aplicaciones tienen en común el hecho de que son ejecutadas directamente por el sistema operativo, ya sea Microsoft Windows, Mac OS X o Linux, y su rendimiento depende de diversas configuraciones de hardware como memoria RAM, disco duro, etc.

○ **MySQLAdministrator:**

Software de administración de servidores de bases de datos de MySQL que ha creado

MySQL AB. Se trata de un software multiplataforma, que se encuentra disponible para Linux y Microsoft Windows y que cuenta con un entorno gráfico de usuario muy intuitivo. Suple las carencias que tiene MySQL Control Center en el área de Administración de servidores. MySQL Control Center ha quedado obsoleto y no sigue desarrollándose. Se ha sustituido por el conjunto de programas MySQL Administrator y MySQL Query Browser. MySQL Administrator es una herramienta que permite realizar tareas administrativas sobre servidores de MySQL.

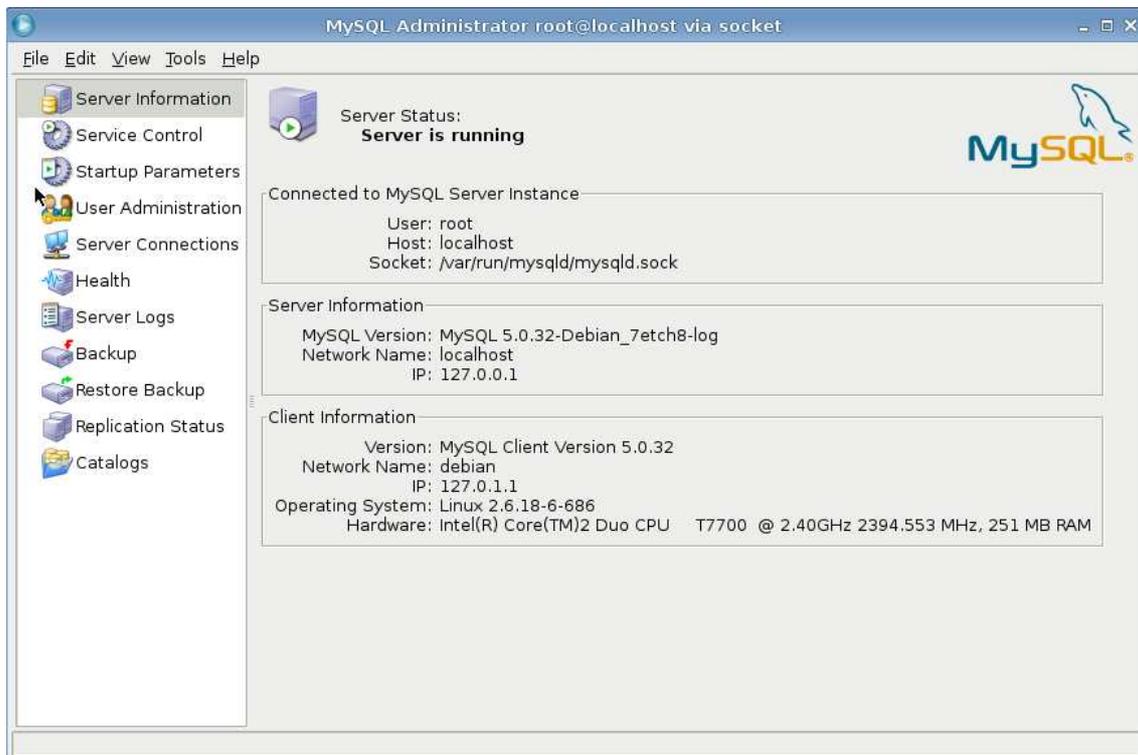


Figura 2.6.2 1

Ejemplo aplicación de escritorio