

5. IMPLEMENTACIÓN

5.1. Alternativas de implementación

Se expondrán a continuación la variedad de herramientas y opciones que se encuentran en el mercado y posteriormente en el punto 5.2, se especificarán las alternativas elegidas en el desarrollo de este proyecto.

5.1.1. Suite Ofimática

Existen varias suites ofimáticas alternativas, como:

- OpenOffice.org, de software libre y código abierto desarrollada por StarDivision
- Microsoft Office, que es considerado como el estándar de facto en el software de productividad (contiene el sistema de gestión de base de datos Microsoft Access).
- WordPerfect Office de Corel.
- iWork, la única suite ofimática sólo para los Mac de Apple. Incluye Pages como procesador de textos y Keynote para presentaciones.
- KOffice, una suite ofimática de código abierto que forma parte del entorno de escritorio KDE.
- GNOME Office, un grupo de aplicaciones de código abierto como Abiword y Gnumeric, que está destinado al entorno de escritorio GNOME.
- Lotus SmartSuite, cuyo proveedor es IBM, contiene un procesador de textos llamado Word Pro, la hoja de cálculo Lotus 1-2-3, un programa para hacer presentaciones denominado Lotus Freelance Graphics y la base de datos Lotus Approach. Lotus Notes ofrece por su parte funcionalidades de correo electrónico y PIM.
- SoftMaker Office, una suite ofimática alemana que incluye un procesador de texto, una hoja de cálculo, un gestor de base de datos y una herramienta de programación.

- ThinkFree Office, una suite ofimática alternativa gratuita basada en web. Es prácticamente compatible en su totalidad con los archivos de Microsoft Office.
- AppleWorks, la suite de Apple para Mac y Windows. Incluye un procesador de textos, hoja de cálculo, software de presentación, gráficos vectoriales y raster y una base de datos.
- WPS Office, cuyo objetivo principal son los usuarios chinos. Incluye un procesador de textos, hoja de cálculo y software de presentación. La versión más reciente es WPS Office 2005. Su versión personal es descargable de forma gratuita.

5.1.2. Sistema de Gestión de base de datos (SGBD)

Los Sistemas de gestión de base de datos son 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, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System, su expresión inglesa.

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos.

Existen distintos objetivos que deben cumplir los SGBD:

- Abstracción de la información.
- Independencia.
- Redundancia mínima.
- Consistencia.
- Seguridad.
- Integridad.
- Respaldo y recuperación.
- Control de la concurrencia.
- Tiempo de respuesta.

Podemos hacer una división algunas de las SGBD más importantes en dos grupos:

1. SGBD libres

- PostgreSQL
- MySQL
- Firebird
- SQLite
- Sybase ASE (Edición gratuita para Linux)

2. SGBD comerciales

- DBase
- FileMaker
- Fox Pro
- IBM DB2
- IBM Informix
- MAGIC
- Microsoft SQL Server
- Open Access
- Oracle
- Paradox
- PervasiveSQL
- Progress (DBMS)
- Sybase ASE
- Sybase ASA
- Sybase IQ
- WindowBase

En un principio se estudió la posibilidad de usar el completo sistema de gestión de base de datos Microsoft Access (contenido en la suite Microsoft Office), pero fueron dos los puntos principales por la que se descartó la idea:

- Para bases de datos de gran calibre (en volumen y/o usuarios) es recomendable usar otros sistemas (como los que se estudian a continuación).
- Se trata de software no libre (no encaja con la política a seguir por la empresa Isotrol).

Las dos alternativas más interesantes que nos encontramos son: PostgreSQL y MySQL, a continuación se ofrece una comparativa de ambas.

MySQL

Su principal objetivo de diseño fue la VELOCIDAD. Se sacrificaron algunas características esenciales en sistemas más "serios" con este fin.

Otra característica importante es que consume MUY POCOS RECURSOS, tanto de CPU como de memoria.

Ventajas:

- Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir "selects" y demás.
- Mejores utilidades de administración (backup, recuperación de errores, etc).
- Aunque se cuelgue, no suele perder información ni corromper los datos.
- Mejor integración con PHP.
- No hay límites en el tamaño de los registros.
- Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
- MySQL se comporta mejor que Postgres a la hora de modificar o añadir campos a una tabla "en caliente".

Inconvenientes:

- No soporta transacciones, "roll-backs" ni subselects.
- **No considera las claves ajenas.** Ignora la integridad referencial, dejándola en manos del programador de la aplicación.

PostgreSQL

Postgres intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase.

Ventajas:

- Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.
- Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
- Tiene mejor soporte para triggers y procedimientos en el servidor.
- **Soporta un subconjunto de SQL mayor** que el que soporta MySQL. Además, tiene **ciertas características orientadas a objetos.**

Inconvenientes:

- Consume BASTANTES más recursos y carga más el sistema.
- Es de 2 a 3 veces más lenta que MySQL.
- Menos funciones en PHP.

Como conclusión a la comparación entre MySQL y Postgres, parece aceptado que MySQL junto con Apache y PHP forman un buen equipo para servir páginas web con contenido dinámico, discusiones, noticias, etc. En general, sistemas en los que la velocidad y el número de accesos concurrentes sea algo primordial, y la seguridad no sea muy importante (pueda bastar con hacer backups periódicos que se restaurarán tras una caída del servidor). En cambio, **para sistemas más serios en las que la consistencia de la BD sea fundamental** (BD con información realmente importante, bancos, etc.) **PostgreSQL es una mejor opción** pese a su mayor lentitud.

Cualquiera de las dos es pues un sustituto adecuado para Access, pero dadas las características de nuestra base de datos, la mejor opción es PostgreSQL.

5.1.3. Interfaz de usuario. Gestión de la base de datos.

Una vez que nos hemos decidido por la suite ofimática OpenOffice.org (las razones se explican en el siguiente punto), necesitamos crear documentos que contengan elementos fijos (como texto), pero que a la vez permitan a los usuarios agregar información adicional y realizar acciones sobre dicha información. Con OpenOffice tenemos dos opciones para montar la interfaz de usuario y poder así gestionar la base de datos:

Formularios

Ventajas:

- Se crean con el piloto automático y son sencillos de implementar.
- Los elementos de control de un formulario pueden ser vinculados con una tabla de una base de datos externa.

Inconvenientes:

- Cada formulario va enlazado a una única tabla de la base de datos, nunca se puede enlazar a más de una tabla.
- Otro de los inconvenientes de los formularios radica en el hecho de que un usuario cuando visualiza un formulario fácilmente puede

cambiar al modo diseño y modificar lo que desee. Esto puede acarrear graves problemas si es usado por manos inexpertas, o si se produce una modificación de forma no intencionada. El nivel de seguridad es mínimo.

- No permite montar varias páginas en el mismo formulario
- Visualmente son poco atractivos, ya que son desplegados directamente en el documento, como elementos de dibujo.
- Las funciones de los formularios están solamente disponibles en los documentos de texto y en las hojas de cálculo.

Diálogos

Ventajas:

- Mayor nivel de seguridad.
- Permite montar varias páginas en el mismo diálogo y avanzar fácilmente de unas a otras. Así se pueden crear autopilotos que guían al usuario en la tarea de introducción de datos.
- Un mismo diálogo permite enlazar múltiples tablas de la base de datos.
- Ofrece una mejor presentación de cara al usuario.
- Los diálogos aparecen en el formulario en una ventana de diálogo sencilla, la cual es desplegada sobre un documento que no permite ninguna otra acción más que el proceso del diálogo, hasta que éste es finalizado.
- Las funciones de los diálogos están disponibles en todos los documentos OpenOffice.org.

Inconvenientes:

- Montaje más complejo.
- Los elementos de control de un diálogo no puede ser vinculado con una tabla de una base de datos externa.

5.2. Solución Adoptada

De todas las opciones que nos encontramos, la elección se hace principalmente por varios motivos:

- Requerimientos específicos de la propia empresa Isotrol
- Mejor adaptabilidad a nuestras necesidades
- Software libre

Así pues, se ha hecho una elección de las siguientes herramientas:

Suite Ofimática	OpenOffice.org
Sistema de Gestión de Base de datos	PostgreSQL
Aplicación gráfica para gestionar PostgreSQL	PgAdmin III
Interfaz de usuario	Diálogos
Lenguaje de consulta a base de datos	SQL
Lenguaje de programación de macros	StarOffice Basic

5.2.1. Acceso a bases de datos con OpenOffice.org

Para desarrollar aplicaciones sobre bases de datos se puede utilizar lenguajes de programación como C o C++, python, pascal, . . . La programación con estos lenguajes permite una gran flexibilidad y rendimiento. Sin embargo, es bastante fácil cometer errores o no programar eficientemente. Ésta es una de las razones por las que se han desarrollado generadores de aplicaciones, de formularios y de informes, que hacen que el desarrollo de aplicaciones sobre bases de datos sea mucho más fácil, fiable y rápido.

OpenOffice.org, en las versiones actuales, no posee un gestor de base de datos, pero dispone de un componente muy potente para el acceso a diversas bases de datos externas que puede así integrarse dentro de la interfaz o los documentos de OpenOffice.org. El modelo de integración de datos de OpenOffice.org es muy peculiar y permite crear sencillas aplicaciones de bases de datos que, por ser OpenOffice.org multiplataforma, pueden utilizarse en distintos sistemas operativos (linux, MS Windows, Solaris, . . .).

OpenOffice.org, como ya se ha dicho, no es una base de datos ni un SGBD. Muchos gestores de bases de datos no ofrecen una interfaz gráfica al usuario para acceder a los datos, suelen trabajar en background como procesos servidores que responden a peticiones de otros programas para almacenar o recuperar datos. Estos programas, llamados clientes, pueden tener una interfaz gráfica que facilita al usuario el acceso a los datos.

OpenOffice.org es un cliente y como tal puede entablar una relación cliente/servidor con diferentes tipos de servidores de bases de datos. La conexión entre un cliente y un servidor se establece normalmente con la ayuda de programas denominados connectors (conectores). Cada servidor de bases de datos tiene sus propios conectores.

Para poder utilizar un determinado servidor con OpenOffice.org es necesario configurar un conector para él. Los SGBDR son las fuentes de datos más potentes que puede utilizar OpenOffice.org, pero puede utilizar otros tipos más simples.

Los tipos de fuentes de datos que, por el momento, puede utilizar son:

- Servidores que soportan Java Database Connectivity, *JDBC*.
- **Servidores que soportan Open Database Connectivity, ODBC** (Oracle, **postgresql**, mysql, access, sqlServer, . . .).
- Servidores MySQL.
- Ficheros en formato de dbase.
- Ficheros en formato de Adabas.
- Ficheros de texto.
- Hojas de cálculo.
- Libretas de direcciones.

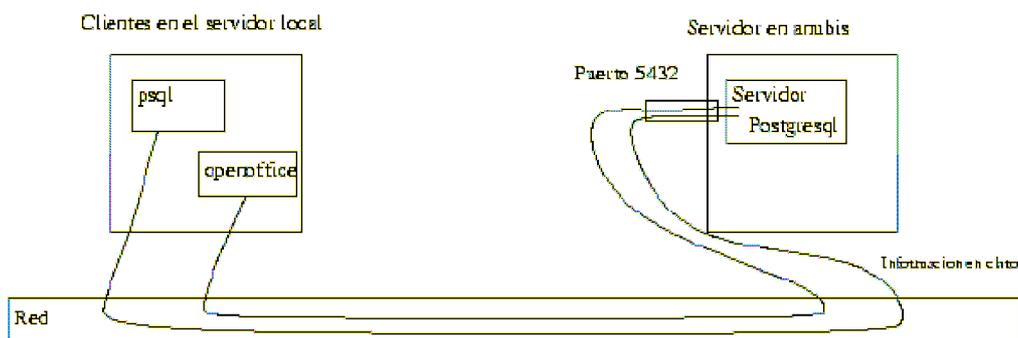


Ilustración 5.1

OpenOffice.org ofrece las siguientes funcionalidades:

- *Registro de bases de datos*, lo cual permite a OpenOffice.org conectarse a las fuentes de datos.
- *Creación de tablas* utilizando una sencilla interfaz gráfica.
- *Diseño de consultas* que permite extraer información de la base de datos en base a condiciones especificadas por el usuario. OpenOffice.org posee un diseñador gráfico de consultas que simplifica la creación de las mismas. También es posible definir consultas utilizando directamente SQL.
- *Diseño de formularios*, los formularios se utilizan para mostrar e introducir datos de un modo amigable. Se pueden utilizar para asegurar que el usuario introduce los datos apropiados (validación) y ocultan la complejidad de la aplicación subyacente (es conveniente que los usuarios finales no trabajen directamente con las tablas en las que pueden causar daños irreparables). OpenOffice.org posee una herramienta para crear formularios. También se pueden crear de forma manual.
- *Diseño de informes* que permiten dar formato a los datos extraídos de la base de datos para crear documentos impresos. Los informes se crean mediante una herramienta específica.
- *Diseño de macros* que permiten automatizar parte de la aplicación. Las macros se pueden crear grabando una serie de pasos (acciones) realizados o utilizando el editor de macros. Las macros se pueden ejecutar como consecuencia de la ocurrencia de eventos en la interfaz.
- *Creación de mensajes de correo o cartas personalizadas* a partir de los datos en la base de datos.

5.2.2. Conexión ODBC (Open DataBase Connectivity)

ODBC, conectividad abierta de bases de datos, es una interfaz de programación desarrollada por Microsoft. El objetivo de *ODBC* es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (*DBMS* por sus siglas en inglés, *SGBD* en español) almacene los datos. *ODBC* logra esto al insertar una capa intermedia llamada manejador de Bases de Datos, entre la aplicación y el *SGBD*, el propósito de esta capa es traducir las consultas de datos de la aplicación en comandos que el *SGBD* entienda. Para que esto funcione tanto la aplicación como el *SGBD* deben ser compatibles con *ODBC*, esto es que la aplicación debe ser capaz de producir comandos *ODBC* y el *SGBD* debe ser capaz de responder a ellos. Ésta conexión es transparente al usuario y se consigue a través del driver adecuado a la base de datos accedida.

Esta es la idea: por un lado el ODBC provee de unas características siempre homogéneas, y por el otro permite distintos controladores que aseguran la conectividad de la aplicación con diferentes bases de datos

Para conectarse a la Base de Datos se crea una DSN dentro del **ODBC** que define los parámetros, ruta y características de la conexión según los datos que solicite el fabricante.

Instalación del soporte ODBC para PostgreSQL

Para crear la conexión tenemos que seguir los siguientes pasos:

1. Tenemos que acceder dentro de Windows a:
Inicio > configuración > panel de control > Herramientas administrativas
2. Buscamos el icono llamado "Orígenes de Datos (ODBC)" y hacemos doble clic sobre él.

3. Se nos abrirá un panel desde el cual podemos ver todos los drivers que tiene nuestro sistema para hacer conexiones sobre las bases de datos vía ODBC. Aquí podremos localizar los drivers de Microsoft Access, MySQL, DB2, Oracle,... siempre y cuando los hayamos instalado en nuestro sistema.
4. En este panel pinchamos sobre la pestaña de DSN del Sistema y pulsamos sobre el botón Agregar. Esto hará que se lance el wizard de creación de una nueva conexión.
5. Seleccionamos un controlador para el que deseamos establecer un origen de datos. En nuestro caso elegimos "PostgreSQL Unicode".
6. El campo 'Nombre del origen de datos' será el nombre que le demos a la conexión y el que utilizemos posteriormente en nuestro código.

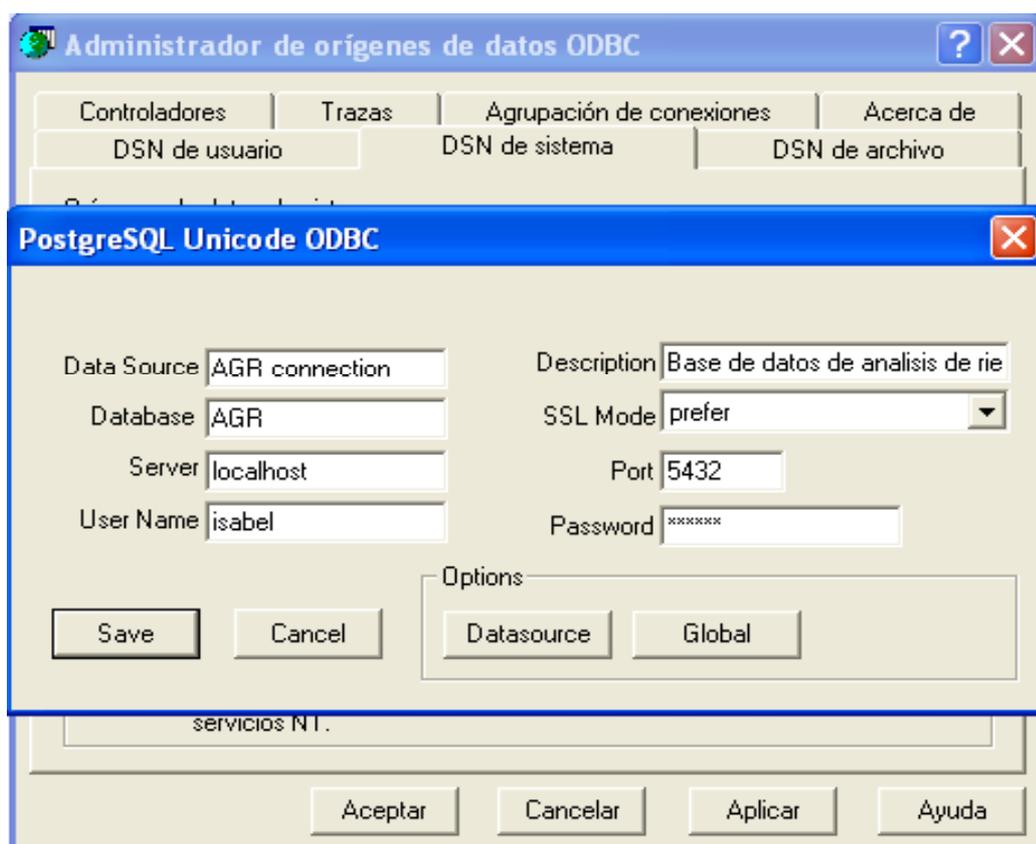


Ilustración 5.2

7. Y solo nos quedará pulsar sobre Aceptar para tener nuestra conexión creada.

5.2.3. Acerca de StarOffice Basic

Como su propio nombre indica, StarOffice Basic (también conocido como StarBasic y OOBASIC) es un lenguaje de programación de la familia Basic.

El lenguaje de programación StarOffice Basic se divide en cuatro componentes:

- **El lenguaje de StarOffice Basic:** define las construcciones elementales del lenguaje; por ejemplo, declaraciones de variables, bucles y funciones.
- **La biblioteca de ejecución:** proporciona funciones estándar sin referencia directa a StarOffice; por ejemplo, declaraciones de variables, bucles y funciones.
- **La API (interfaz de programación de aplicaciones) de StarOffice:** permite acceder a los documentos de StarOffice, crearlos, guardarlos, modificarlos e imprimirlos.
- **El editor de diálogos:** crea cuadros de diálogo personalizados y ofrece un entorno para agregar elementos de control y gestores de eventos.

StarOffice Basic permite automatizar las tareas rutinarias, crear vínculos con otros programas (por ejemplo, un servidor de bases de datos) y utilizar scripts predefinidos para llevar a cabo actividades complejas con una simple pulsación de botón.

StarOffice Basic es un lenguaje interpretado. Un programa de StarOffice Basic se puede ejecutar pulsando un botón. El código se verifica primero para localizar los errores más obvios y a continuación se ejecuta línea por línea.

StarOffice ofrece centenares de servicios que, para mostrar una versión simplificada, se han combinado en módulos. Dichos módulos carecen de ninguna otra importancia funcional para los programadores de StarOffice Basic. Únicamente al especificar un nombre de servicio tiene importancia el nombre del módulo. El nombre completo de un servicio incluye la expresión `com.sun.star`, que especifica que se trata de un servicio de StarOffice, seguida por los nombres del módulo y del servicio en sí. Aparte de los términos de módulo y servicio, UNO introduce el de *interfaz*. Una interfaz combina diversos métodos.

5.2.4. Introducción a la API de StarOffice

La API (Application Programming Interface) de StarOffice es una interfaz de programación universal para acceder a OpenOffice.org que se puede utilizar para crear, abrir, modificar e imprimir documentos de OpenOffice.

Ofrece la opción de ampliar el ámbito funcional de StarOffice mediante macros personales, así como escribir diálogos personalizados.

La API de StarOffice no solo se puede usar con StarOffice Basic, sino también con otros lenguajes de programación como Java y C++. Esto es posible gracias a una técnica denominada UNO (*Universal Network Objects*, Objetos de red universales), que ofrece una interfaz orientada a objetos para diversos lenguajes de programación.