
This is the **published version** of the bachelor thesis:

Pocaterra van Schermbeek, Mariangela; Rayo, Xavier. Configuración de la base de datos espacial de los servidores web del parque de Collserola, y aplicación de la funcionalidad de los clientes web caso Incidencias faunística. 2009.

This version is available at <https://ddd.uab.cat/record/50362>

under the terms of the  license

**CONFIGURACIÓN DE LA BASE DE DATOS ESPACIAL DE LOS
SERVIDORES WEB DEL PARQUE DE COLLSEROLA, Y
AMPLIACIÓN DE LA FUNCIONALIDAD DE LOS CLIENTES WEB
CASO INCIDENCIAS FAUNÍSTICAS.**

***Proyecto final de Máster en Tecnologías de la Información
Geográfica, 10ª edición.***

REALIZADO POR:

Mariangela Pocaterra Van schermbeek

TUTORES:

Miguel Ángel Vargas (UAB).

Raimon Reventos.

Xavier Rayo (UAB).

10mtig²⁰⁰⁸
Professionals per a la Societat de la Informació

UAB
Universitat Autònoma de Barcelona
Departament de Geografia

 **Parc de
Collserola**

FEBRERO 2008.

RESUMEN.

Las tecnologías de la información es un conjunto de servicios, redes, software y aparatos que tienen como fin la mejora de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario. Esta innovación servirá para romper las barreras que existen entre cada uno de ellos.

Actualmente las Tecnologías de la Información están sufriendo un desarrollo vertiginoso, esto está afectando a prácticamente todos los campos de nuestra sociedad. Esas tecnologías se presentan cada vez mas como una necesidad en el contexto de sociedad donde los rápidos cambios, el aumento de los conocimientos y las demandas se convierten en una exigencia permanente.

El presente trabajo tiene como objetivo general configuración de la base de datos espacial de los servidores web del parque de collserola, y ampliación de la funcionalidad de los clientes web.

La propuesta de realizar configuración de la base de datos espacial de los servidores web del parque de collserola, y ampliación de la funcionalidad de los clientes web caso incidencias de la fauna surge como una respuesta a la necesidad de: 1) Poder tener procedimientos establecidos para migrar Shapefile a bases de datos, 2) poder actualizar la pagina web con las bases de datos y que los usuarios puedan interactuar de manera dinámica para así poder reportar atropellos que observen dentro del área del parque.

La primera fase del proyecto esta basada en la imigración de archivos tipos shepfile contenidos en un PostGis y que se requiere convertir a bases de datos a través de la herramienta PostgreSQL.

La segunda fase está basado en un servidor de HTTP Apache y desarrollado sobre MapServer como servidor de mapas y el entorno P.mapper, para las funcionalidades del visor cliente. Además el aplicativo permite el acceso a la

tecnología Web Mapping Services (WMS) de código abierto y es compatible con los estándares del OpenGis Consortium (OGC).

Palabras clave: Parc de Collserola, atropello de fauna, Servicios, software libre, servidores de mapas, PostgreSQL, QuamtuGis, WebGIS Parc Collserola.

AGRADECIMIENTOS.

- A Dios por haber creado un mundo maravilloso y permitirnos vivir en él.
- A mi tutor, Xavier Rayos, por su continua orientación, aliento, y dedicación al asesorarme en este trabajo.
- A mis padres y hermanos por su apoyo.
- Al Geo. Raimon Reventos (Técnico SIG Consorci ParcCollserola) por el apoyo prestado siempre que lo necesite.
- A todos el equipo del Ligit
- A mi amiga Ruth María Torres y a Marcos Gallegos por apoyarme y dame el aliento necesario para continuar.
- Javier Moriana.
- A la Fundación Gran Mariscal de Ayacucho por su apoyo.

ÍNDICE.

I. INTRODUCCIÓN.....	7
1.1. Antecedentes.....	7
II. OBJETIVOS DEL PROYECTO.....	9
2.1. Objetivo General.....	9
2.2. Objetivos Específicos.....	9
III. METODOLOGIA.....	10
3.1 Fases.....	10
3.2 cronograma.....	12
IV. DISEÑO DE LA BASE DE DATOS.....	13
4.1 Las bases de datos.....	13
4.1.1 Antecedentes.....	14
4.1.2 Modelo conceptual.....	15
4.1.3 Modelo lógico.....	18
V. TECNOLOGIAS Y PROGRAMAS APLICADOS.....	21
5.1 MS4W.....	22
5.2 Servidor HTTP Apache.....	23
5.2.1 PHP (PHP Hipertext Pre-processor).....	23
5.3 P.mapper.....	27
5.4 PostgreSQL.....	28
5.5 PostGis.....	29
5.6 Quamtu Gis.....	30
VI. PROCEDIMIENTOS Y IMPLEMENTACIÓN.....	32
6.1 Imigración de bases de datos de Microsoft Access a PostgreSQL.....	34
6.2 Imigración de Shapefile a PostGis.....	30
6.3 creación y preparación de la base de datos.....	36
6.4 Interfaz del administrador de la base de dato.....	38
6.5 Funcionalidades implementadas.....	40
VII. CONCLUSIONES.....	45
VIII. BIBLIOGRAFIA.....	47
IX. ANEXOS.....	48

ÍNDICE DE FIGURAS.

Fig. 5.2.1-1 Estructura Sistema GIS web MapServer.....	26
Fig. 5.3-1 Arquitectura general de integración de p.mapper.....	28
Fig. 6.1-1 Exportación de Access 95 a 2033.....	32
Fig. 6.1-2 Exportación de la base de dato a access.....	33
Fig. 6.1-3 Conexión de la base de datos ODBC a Postgres.....	33
Fig. 6.2-1 Archivos utilizados para la migración de shp a PostGis.....	34
Fig. 6.2-2 Exportación de shap a potgis a travez de comandos.....	35
Fig. 6.2-3 SQLI de los shap exportados a postgis.....	35
Fig. 6.2-4 Envío de las tablas cradas de los shap al PostgreSQL.....	36
Fig. 6.3-1 Envío de las tablas cradas de los shap alpostgresql.....	37
Fig. 6.3-2 Ubicación de las tablas dentro del PostgreSQL.....	37
Fig. 6.3-3 Montaje de script de sql a postgres.....	38
Fig. 6.4-1 Administrador de la base de datos.....	39
Fig. 6.4-2 Estructura de la tabla obsfauna en la interfaz del administrador.....	39
Fig. 6.4-3 Edición de una fila de la tabla obsfauna en la interfaz del administrador.....	40
Fig. 6.5-1 Representación de líneas y polígonos.....	41
Fig. 6.5-2 Representación de líneas, polígonos y puntos.....	41
Fig. 6.5-3 Login del usuario.....	42
Fig. 6.5-4 Formulario de incidencias faunística.....	42
Fig. 6.5-5 Herramienta y funcionalidad del botón del formulario.....	43

ÍNDICE DE ESQUEMAS.

Esquema 3.1-1 fases del proyecto.....	11
Diagrama 4.1.3 -1 Modelo lógico de las incidencias Faunística.....	20

ÍNDICE DE CUADROS.

Cuadro 3.2-1 Cronograma de Actividades.....	12
Cuadro 4.1.1-1 Datos existentes para diseñar el modelo conceptual y lógico de la nueva base de dato de incidencias faunística.....	14
Cuadro 4.1.3 -1 Características que debe mostrar el diseño de las tablas asociadas a cada entidad.....	18

I. INTRODUCCION.

1.1 Antecedentes.

El parque natural de Collserola forma parte del área metropolitana de Barcelona, donde viven más de tres millones de personas repartidas por un conjunto de grandes municipios. Collserola es, por lo tanto, más que un espacio natural, es un oasis metropolitano, el pulmón de esta gran zona urbana, y por esta razón su valor ecológico es todavía más importante. Este parque cuenta con una extensión de cerca de 8.000 hectáreas y cada año libera 60.000 toneladas de oxígeno y elimina 80.000 de dióxido de carbono. Está poblado por unos 10 millones de árboles, 1.000 especies de plantas diferentes y cerca de 190 tipos de animales vertebrados. Tomando en cuenta que el 50% de la población de Cataluña vive a menos de diez kilómetros del parque, considerándose el parque metropolitano más grande del mundo. Estas características hacen del Parque un centro fundamental relacionado con actividades de ocio y ecología, con un gran interés potencial por su situación geográfica.

En la actualidad el parque Collserola posee una pagina Web la cual esta elaborada bojo software libre tanto en el costado cliente (MapServer) como en el costado servidor (Apache). Existe la configuración de formularios programados por asistir al usuario en la búsqueda de datos, la impresión de documentos o la selección de elementos; sin embargo, los datos geográficos que alimentan los servicios consisten, básicamente, en archivos de imágenes georeferenciadas y archivos en formato Shapefile, aparte de otros archivos vinculados a elementos en formato Word, Excel. Es por esta razón que se propone como una primera fase ampliar la funcionalidad del servidor de mapas, esta ampliación consiste en cargar a este servidor con bases de datos.

Esta base de datos tendrá que estar desarrollada dentro de un software compatible con los ya utilizados, así como deberá mantener las características de ser un software libre por esta razón se propone la utilización del **PostgreSQL**.

Con esta migración de archivos Shapefile a un entorno a base de datos se quiere promover una mayor integridad y control del conjunto de datos que participan en los servidores.

La segunda fase requiere del desarrollo de formularios de incidencias faunística donde el usuario del Parc de Collserola podrá registrar incidencias observadas, este registro estará actuando interactivamente con el visor del mapa para lograr puntualizar la ubicación de la incidencia. Toda esta información se encontrará almacenada en la base de datos anteriormente creada.

El proyecto se encuentra enmarcado en el convenio de colaboración entre la Universidad Autònoma de Barcelona (UAB) y el CPC, para el desarrollo del proyecto final de los alumnos del Master en Tecnologías de la Información Geográfica, 10ª. Edición.

II. OBJETIVOS DEL PROYECTO.

2.1. Objetivo General.

Creación de la base de datos y ampliación de la funcionalidad de los clientes web: caso de incidencias de la fauna.

2.2. Objetivos Específicos.

✚ Crear una base de datos general para los datos alfanuméricos de los atropellos de la fauna del Parque Collserola.

✚ Editar la base de datos desde el cliente la cual permita :

- Introducir y almacenar datos en la base de datos
- Editar datos una vez introducidos.
- Realizar consultas específicas a partir de la información alfanumérica y cartográfica como:

1. Clase de especies.
2. Localización de la especie
3. características generales.

✚ Visualizar la cartografía así como algunas de las consultas en la web.

III METODOLOGÍA.

3.1 Fases.

Exploración: Se realizó una búsqueda exhaustiva de la información bibliográfica sobre el tema en estudio, así como la revisión del tipo de dato y la cartografía con los cual se trabajo.

Definición: Se definió el sistema de información y después se estableció cual es la información necesaria a incluir en su base de datos.

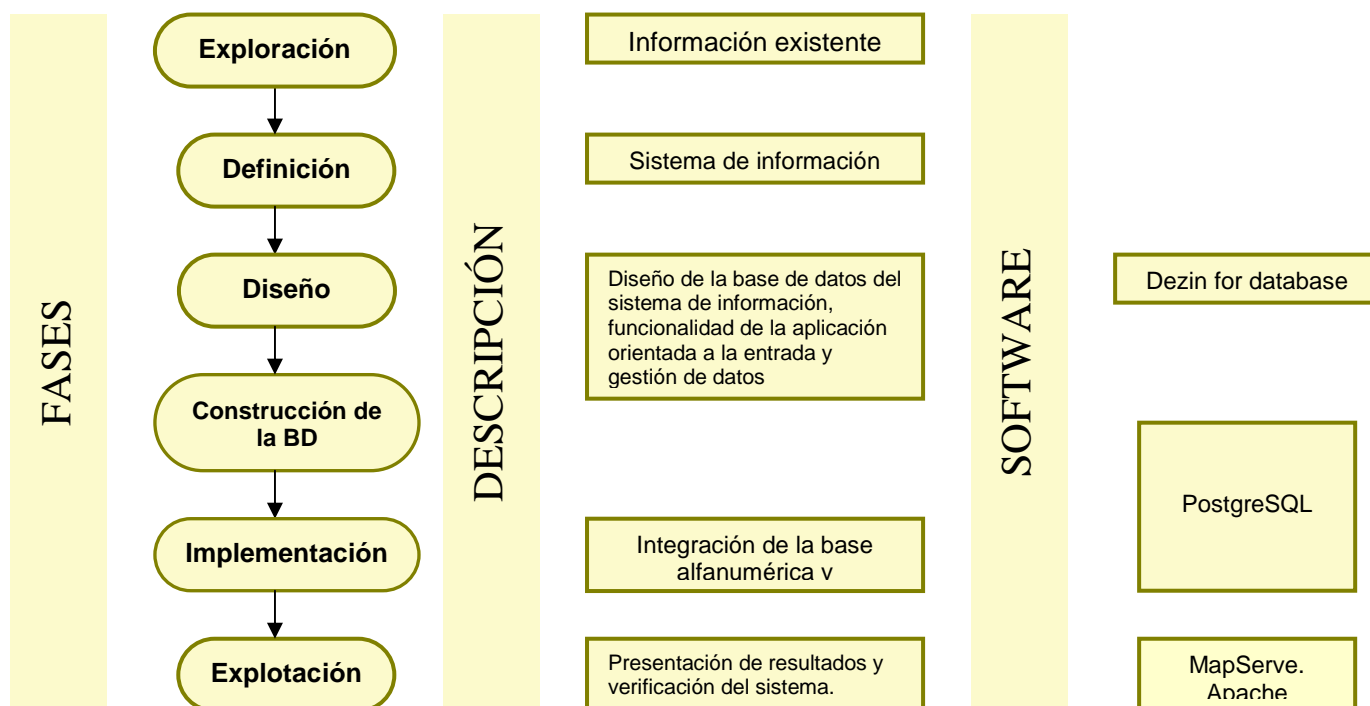
Diseño: Consistió en el diseño de la base de datos con la información suministrada por Parc de Collserola referente a las incidencias de la fauna utilizando el programa Dezipn for database y que sigue el modelo relacional.

Implementación: con la utilización del PostgreSQL se creo la estructura de la futura aplicación. Esta estructura incluye formularios para la introducción, edición y consulta de datos alfanuméricos así como la visualización de mapas donde se ha implementado alguna consulta.

Los códigos necesarios para la ejecución de la aplicación y creación de algunos formularios se desarrollaran en el PostgreSQL.

Por último se emigrara de PostgreSQL a MapServe y montado a la pagina Web para realizar las consultas establecidas.

ESQUEMA 3.1-1 FASES DEL PROYECTO



Fuente: Elaboración propia, año 2008.

Para poder llevar a cabo correctamente la recolección de la información correspondiente a las incidencias faunísticas del área en estudio, fue necesario emplear técnicas de investigación, los cuales permitieron apreciar una percepción más clara de las necesidades y falencias que cubrirá la aplicación generada. Algunas de las técnicas empleadas se describirán continuación:

Consultas documentales: Las consultas documentales secundarias hacen referencia directa a las enciclopedias, los anuarios, manuales, almanaques, las bibliografías, índices, entre otros; los datos que integran las fuentes secundarias se basan en documentos primarios. En la realización de este proyecto se utilizó este instrumento en la búsqueda de información sobre el manejo de los lenguajes de programación HTML, PHP, JavaScript, PostgreSQL entre otros.

3.2. Cronograma.

CUADRO 3.2-1.
CRONOGRAMA DE ACTIVIDADES.

Actividades	Fecha de inicio	Fecha de culminación	Septiembre 2008			Octubre 2008				Noviembre 2008				Diciembre 2008		
			38	39	40	41	42	43	44	45	46	47	48	49	50	51
Revisión de bibliografía.	15/09/08	26/09/08														
Determinación de las capas y revisión de la misma	29/09/08	3/09/08														
Diseño de la base de datos	6/10/08	17/10/08														
Manejo y carga de la base de datos	20/10/08	7/11/08														
Migración de la base de datos a Mapserve	10/11/08	21/11/08														
Creación de las herramientas de consulta y pruebas de la aplicación en la web.	24/11/08	5/12/08														
Pruebas de la aplicación en la web.	8/12/08	19/12/08														

Fuente: Elaboración propia. Año 2008

IV. DISEÑO DE LA BASE DE DATOS.

4.1. Las bases de datos.

El diseño de una base de datos es un proceso complejo que abarca decisiones a muy distintos niveles. La complejidad se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de estos subproblemas independientemente.

El diseño de una base de datos se descompone en diseño conceptual, diseño lógico y diseño físico.

El diseño conceptual parte de las especificaciones de requisitos de usuario y su resultado es el esquema conceptual de la base de datos. Un *esquema conceptual* es una descripción de alto nivel de la estructura de la base de datos, independientemente del SGBD que se vaya a utilizar para manipularla.

El diseño lógico parte del esquema conceptual y da como resultado un esquema lógico. Un *esquema lógico* es una descripción de la estructura de la base de datos en términos de las estructuras de datos que puede procesar un tipo de SGBD.

El diseño físico parte del esquema lógico y da como resultado un esquema físico. Un *esquema físico* es una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos. Por ello, el diseño físico depende del SGBD concreto y el esquema físico se expresa mediante su lenguaje de definición de datos

Las bases de datos bien diseñadas y alimentadas, no es más que una herramienta de consulta. Esta permite combinar cualquier información con todo tipo de consultas que se le formulen, disponer de una significa controlar la amplia gama de información con la que trabaja cualquier tipo de negocio.

4.1.1. Antecedentes.

Para realizar o crear la base de datos que soportara la aplicación se recogió todo un cúmulo de información existente relativa a las incidencias faunística del Parc del Collserola.

En el cuadro 4.1.1-1 se presenta los datos que se utilizaron para la creación de la nueva base de dato.

CUADRO 4.1.1-1
DATOS EXISTENTES PARA DISEÑAR EL MODELO CONCEPTUAL Y LÓGICO DE LA NUEVA
BASE DE DATO DE INCIDENCIAS FAUNÍSTICA.

Base de datos	Descripción
Zfaunapt	Esta representada en un formato de Shp de punto y contiene información de codi, X_coord, Y_coord
Obsfauba	Esta representada en un formato DBF conteniendo información alfanumérica y textual.
Fauna	Es una base de datos que contiene todas las tablas relacionadas en Excel formato MDB
Existe una información cartográfica de: areeslleure, caminsli, cobertes_creaf, comarques, districtes, f5000pl, fauna, limit_cpc, limit_pn, linelectli, habitats, linelectnd, municipis, MV-2001, pictogrames, pkvials.	
	Todas estas capas están en formato shepfile (puntos, líneas, polígonos, texto)

Fuente: consorcio de Parc de Collserola, año 2008

4.1.2. *Modelo conceptual.*

Los modelos conceptuales se utilizan para representar la realidad a un alto nivel de abstracción. Mediante los modelos conceptuales se puede construir una descripción de la realidad fácil de entender.

En el diseño de bases de datos se usan primero los modelos conceptuales para lograr una descripción de alto nivel de la realidad, y luego se transforma el esquema conceptual en un esquema lógico. El motivo de realizar estas dos etapas es la dificultad de abstraer la estructura de una base de datos que presente cierta complejidad. Un esquema es un conjunto de representaciones lingüísticas o gráficas que describen la estructura de los datos de interés.

Los modelos conceptuales deben ser buenas herramientas para representar la realidad, por lo que deben poseer las siguientes cualidades:

- ✚ **Expresividad:** deben tener suficientes conceptos para expresar perfectamente la realidad.
- ✚ **Simplicidad:** deben ser simples para que los esquemas sean fáciles de entender.
- ✚ **Minimalidad:** cada concepto debe tener un significado distinto.
- ✚ **Formalidad:** todos los conceptos deben tener una interpretación única, precisa y bien definida.

En general, un modelo no es capaz de expresar todas las propiedades de una realidad determinada, por lo que hay que añadir aserciones que complementen el esquema.

Los esquemas conceptuales correspondientes a cada vista de usuario se les denominan esquemas conceptuales locales. Cada uno de estos esquemas se compone de entidades, relaciones, atributos, dominios de atributos e identificadores. El esquema conceptual también tendrá una documentación, que se irá produciendo durante su desarrollo. Las tareas a realizar en el diseño conceptual son las siguientes:

1. Identificar las entidades.
2. Identificar las relaciones.
3. Identificar los atributos y asociarlos a entidades y relaciones.
4. Determinar los dominios de los atributos.
5. Determinar los identificadores.
6. Determinar las jerarquías de generalización (si las hay).
7. Dibujar el diagrama entidad-relación.
8. Revisar el esquema conceptual local con el usuario.

1. *Identificar las entidades.*

Se definió los principales objetos que interesan al usuario. Estos objetos son las entidades. Una forma de identificar las entidades es examinar las especificaciones de los requisitos del usuario.

2. *Identificar las relaciones.*

Una vez definidas las entidades, se definieron las relaciones existentes entre ellas. Del mismo modo que para identificar las entidades se buscaron nombres en las especificaciones de requisitos.

3. *Identificar los atributos y asociarlos a entidades y relaciones.*

Al igual que con las entidades, se buscaron los nombres en las especificaciones de requisitos. Son atributos los nombres que identifican propiedades, cualidades, identificadores o características de entidades o relaciones.

Al identificar los atributos, se tomaron en cuenta si son simples o compuestas.

4. *Determinar los dominios de los atributos.*

El dominio de un atributo es el conjunto de valores que puede tomar el atributo. Un esquema conceptual está completo si incluye los dominios de cada atributo: los valores permitidos para cada atributo, su tamaño y su formato.

5. *Determinar los identificadores.*

Cada entidad tiene al menos un identificador. En este paso, se buscaron todos los identificadores de cada una de las entidades. Los identificadores pueden ser simples o compuestos. De cada entidad se escogerá uno de los identificadores como clave primaria en la fase del diseño lógico.

6. *Determinar las jerarquías de generalización.*

En este paso se observaron las entidades que se identificaron hasta el momento. Hay que ver si es necesario reflejar las diferencias entre distintas ocurrencias de una entidad, con lo que surgirán nuevas subentidades de esta entidad genérica; o bien, si hay entidades que tienen características en común y que realmente son subentidades de una nueva entidad genérica.

7. *Dibujar el diagrama entidad-relación.*

Una vez identificados todos los conceptos, se puede dibujar el diagrama entidad-relación correspondiente a una de las vistas de los usuarios. Se obtiene así un esquema conceptual local.

8. *Revisar el esquema conceptual local con el usuario.*

Antes de dar por finalizada la fase del diseño conceptual, se debe revisar el esquema conceptual local con el usuario. Este esquema está formado por el diagrama entidad-relación y toda la documentación que describe el esquema.

4.1.3. Modelo lógico.

Una vez definido el modelo conceptual esquematizado por el modelo E-R se procede a elaborar el modelo lógico que expresa el diseño de las tablas asociadas a cada entidad. El modelo lógico típico suele describir los siguientes ítems para cada uno de los campos de atributos.

CUADRO 4.1.3-1
CARACTERÍSTICAS QUE DEBE MOSTRAR EL DISEÑO DE LAS TABLAS ASOCIADAS A CADA ENTIDAD.

NOMBRE	Nombre del campo			
CLAVE	Tipo de clave	si es la clave propia de la tabla es PRIMARIA (PK)	si es la clave que conduce a otra tabla es FORÁNEA (FK)	si no es ninguna clave va como un espacio en blanco (- -----)
TIPO	Tipo de clave	Number, String o Caracteres, Date (fechas), boolean (V o F) Campos con valores que combinan números y letras son también del tipo string		
LONGITUD MÁXIMA	numero de caracteres máximo incluyendo espacios	4,7,10, etc		
UNICIDAD	indica la exclusividad o no del valor del campo	Si otra fila puede tener ese mismo valor es NO ÚNICO	si el valor sólo puede ir en una fila es UNICO	
OBLIGATORIEDAD	Si tiene o no que ir un dato en el campo	Si el campo se puede dejar en blanco es NULO	Si el dato debe ser conocido es NO NULO	

Fuente: Elaboración propia, año 2008

La aplicación de incidencias faunística se basa una base de datos preexistente que es obsfauna.mdb, relacionadas con tablas que ofrecen a esta base de datos información sobre incidencias registradas dentro del área del Parc Collserola.

Para realizar una base de datos bien estructurada y unificada de la información del proyecto en estudio, se crearon pautas que permitieron definir y estructurar la base de dato; es decir se clasifico con la ayuda de los técnicos la información que debe presentarse en dicha base de dato.

Se toma en cuenta a la hora de diseñar y estructura la base de datos que en su mayoría los datos serán introducidos directamente por los usuarios del Parc de Collserola, a través de una interfase de la aplicación.

La información referente a las incidencias faunística se estructuro en una base de datos relacional, el diseño realizado presenta las tablas diccionarios. Todas estas tablas poseen su clave primaria determinada para implementar las relaciones de la base de dato, así como posee la clave primaria también se establecieron sus clave ajena las cuales salen de la relaciones entre si. Estas tablas diccionario están representadas en cuatro grandes grupos y si se observa el grafico 4.1.3-1 veremos que las relaciones establecidas son de una a muchas y que solo se encuentra relacionada con una sola tabla que es la encargada de almacenar toda la información.

Estas tablas están agrupadas en cuatro grades grupos y son:

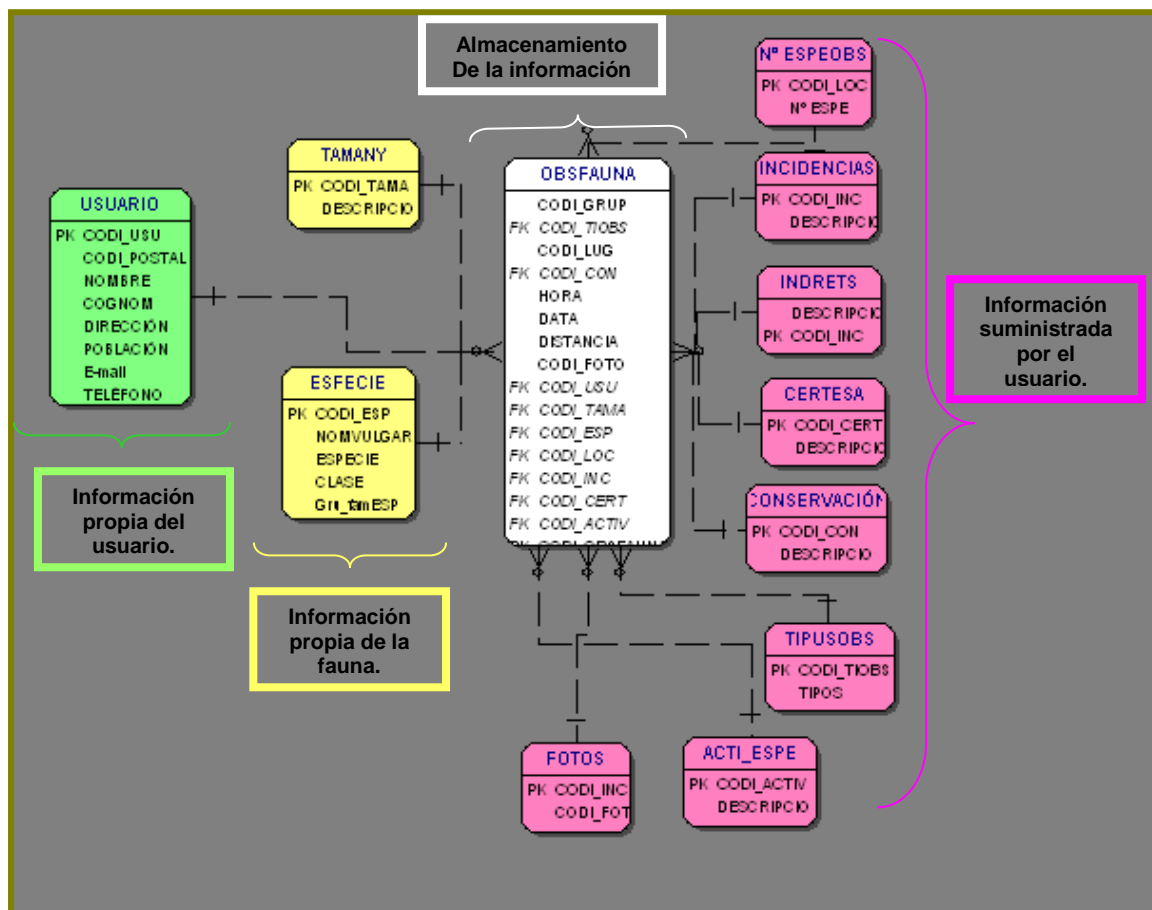
Características del usuario que esta representado por el color verde y que tiene como objetivo principal identificar al usuario que suministrar la incidencia.

Características propias de la especie representada por el color amarillo en este grupo la información que existe es una información intrinsica de la especie.

Características observada por el usuario sobre las especies representada por el color rosa, esta observación solo depende de lo que vea el usuario o como lo vea.

Almacenamiento de la información representada por el color blanco esta finalmente agrupa toda la información geográfica que se utilizada por el Servidor de Mapas para construir los mapas y es la que posee toda la información junta.

DIAGRAMA 4.1.3-1
MODELO LÓGICO DE LAS DE LAS INCIDENCIAS FAUNÍSTICA.



Fuente: Elaboración propia, año 2008

V. TECNOLOGIAS Y PROGRAMAS APLICADOS.

La Base de Datos es el punto más importante del sistema. Cumple dos funciones, almacenar toda la información del sistema y comunicar a los distintos módulos ya que algunos dejan información disponible para que otros la utilicen. Algunos de los requerimientos que se le deben exigir a la Base de Datos para que sea adecuada son:

- ✚ Soporte para información geográfica. Generalmente, los productos de Base de Datos no soportan nativamente el manejo de este tipo de información, pero es posible extenderlos con componentes especiales.

La información geográfica es fundamental en el desarrollo del sistema ya que mucha de la información utilizada será presentada en mapas. Información como calles, manzanas, plazas, ubicación de hospitales o cuarteles de bomberos, es el tipo de información que debe tener representación geográfica.

- ✚ El Gestor de Base de Datos debe soportar manejo de transacciones, acceso concurrente de usuario, soporte eficiente para gran cantidad de registros, herramientas de administración, entre otras propiedades deseables para el tipo de aplicaciones que es un y un sitio web. Principalmente deben exigirse las propiedades de atomicidad, consistencia, aislamiento y durabilidad que el Gestor debe tener para lograr un correcto manejo de transacciones .

En la etapa inicial del proyecto se seleccionó para manejar la información geográfica archivos Shapefile y como gestor de Base de Datos PostgreSQL que es un gestor de código abierto (open source), que cumple con las características exigidas anteriormente [3]. Para manejar información geográfica se utilizó una extensión llamada PostGIS que permite almacenar objetos geográficos, crear índices espaciales sobre los datos geográficos y realizar operaciones básicas sobre los objetos geográficos.

Para el desarrollo del sistema de información se tuvo en cuenta inicialmente la utilización de herramientas propietarias como lo es ArcView 3.2 de ESRI, la cual permitiría desarrollos mas rápidos y menos dificultosos por ser una herramienta que permite muchas funcionalidades SIG y por lo tanto no se tendría la necesidad de grandes bloques de códigos desarrollados. Sin embargo, por ser software propietario, la institución debería incurrir en la compra de licencias, las cuales en la actualidad son de alto costo y la institución, por las restricciones económicas, que presenta no podría asumirlo.

Por ello se consideró como alternativa el desarrollo con herramientas de libre uso y distribución, lo cual suponía un mayor riesgo ya que no se contaba con una estadística comprobada sobre el índice de productividad y efectividad de aplicaciones desarrolladas con estos lenguajes; por lo tanto se inició el proceso de estudio e investigación de los adelantos en este tema pudiendo establecer finalmente el manejo de Mapserver como servidor de mapas, PostGis como extensión de PostgreSQL para el manejo de la información espacial, servidor Web Apache y plataforma pmapper demo para el manejo de algunas funcionalidades SIG. Para el entrenamiento en estas herramientas se optó por utilizar las fuentes de información existentes en Internet y foros al respecto, con el fin de poder indagar sobre el desarrollo y manejo de estas herramientas; se contó también con capacitaciones ofrecidas por personal experto en el tema de otras ciudades e incluso de otros países como Chile (vía chat, video y e-mail).

5.1 MS4W.

MS4W, es un instalador rápido y fácil de MapServer para Windows, así como para sus aplicaciones añadidas que nos proporcionan diversos entornos de desarrollo (p.e. Chameleon, MapLab, Pmapper, etc). El propósito de este paquete es permitir a usuarios iniciales y avanzados instalar rápidamente un ambiente de desarrollo de MapServer.




El paquete MS4W está diseñado para llevar a cabo unas instalaciones completas de Apache, PHP, MapServer CGI, MapScript (CSharp, Java, PHP, Python), completamente configuradas y listas para usar. El diseño de la estructura de

directorios que se crea es compacto permitiendo que la aplicación sea totalmente transferible de manera completa, con solo copiar el directorio de trabajo. Además el diseño del directorio de la aplicación permite que se puedan efectuar actualizaciones parciales de las aplicaciones o incluso la desinstalación de las mismas sin que el resultado tenga que afectar al funcionamiento básico del resto de aplicaciones instaladas.

5.2 Servidor HTTP Apache.

El **servidor HTTP Apache** es un programa de código abierto para plataformas Unix, Windows, Macintosh y otras. El servidor Apache se desarrolla dentro del proyecto HTTP Server de la Apache Software Foundation.

El servidor Apache es un software que esta estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

-  **Módulos Base:** Módulo con las funciones básicas del Apache.
-  **Módulos Multiproceso:** son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
-  **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

5.2.1 PHP (PHP Hypertext Pre-processor).

PHP (PHP Hypertext Pre-processor)

Creado por Rasmus Lerdorf en 1995 como un conjunto simple de scripts PERL para controlar accesos a sus trabajos online llamándolo “Personal Home Page Tools”, posteriormente decidió liberar el código para continuar su desarrollo hasta ser uno de los mas importantes lenguajes de programación en entorno WEB.




Este lenguaje de programación presenta características de tipo clásico al utilizar variables, funciones, condicionantes, etc. a diferencia de http, XML o WML con la característica de ejecutarse en el equipo servidor, no en el cliente como JAVA, admitiendo el acceso a recursos que se encuentran en el servidor como pueden ser bases de datos.

Su interpretación y ejecución se da en el servidor web, en el cual está almacenado el script o archivo de procesamiento por lotes, y el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado que generará el contenido de manera dinámica, pudiendo modificar el contenido a enviar, y devuelve el resultado al servidor, el cual se encarga de devolverlo al cliente. Además es posible utilizar PHP para generar archivos PDF, Flash, así como imágenes en diferentes formatos, permite la conexión a diferentes servidores de bases de datos (MySQL,Postgres,Oracle,ODBC) lo cual permite la creación de aplicaciones web muy robustas. Indicar finalmente que PHP tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos y que puede interactuar con los servidores web más populares.

MapServer es un servidor de mapas en ambiente WEB desarrollado por la University of Minnesota (UMN) Al ser un software de código libre permite ser completamente personalizado por un usuario con conocimientos medios de programación, además de poder ejecutarse tanto bajo plataformas LINUX como Windows 95 o superior. Existe una permanente actualización tanto del “código fuente” del programa como de nuevas versiones creadas por los muchos grupos de desarrolladores que existen alrededor de todo el mundo. Soportado WMS, (Web Map Server) permite la producción de mapas georeferenciados en formatos PNG, GIF o JPEG, mejorando la entrega de resultados al ser requeridos por un cliente que accede por medio de un explorador WEB (Internet Explorer, Firefox, Opera, etc.). Estos archivos de Figura son generados luego de realizada una consulta al servidor, el cual accede a las bases de datos de cualquiera de los múltiples formatos soportados por MAPSERVER, sin necesidad de otros software.

La interfaz gráfica entre servidor y cliente puede ser creada utilizando lenguaje HTML y combinarlo con otros tales como JavaScript, PHP, SQL, etc. MAPSERVER utiliza su propio lenguaje para el despliegue de capas de información geográfica, leyendas, escalas, simbología y la configuración de estas tanto en ancho líneas, color, etiquetado, etc., las cuales se definen en el archivo de configuración o MAPFILE.

Principales Formatos Soportados por MAPSERVER

-  ESRI Shapefiles (SHP).
-  PostGIS / PostgreSQL Database
-  MapInfo Files (TAB/MID/MIF).
-  Oracle Spatial Database.
-  Web Feature Service (WFS).
-  Geography Markup Language (GML).
-  Virtual Spatial Data (ODBC/OVF)
-  ESRI Binary Coverages (ADF).
-  ESRI ArcSDE (SDE).
-  Microstation Design Files (DGN).
-  Spatial Data Transfer Standard (SDTS).
-  Inline MapServer Features.
-  National Transfer Format (NTF).

Las aplicaciones CGI con MapServer utilizan los siguientes recursos:







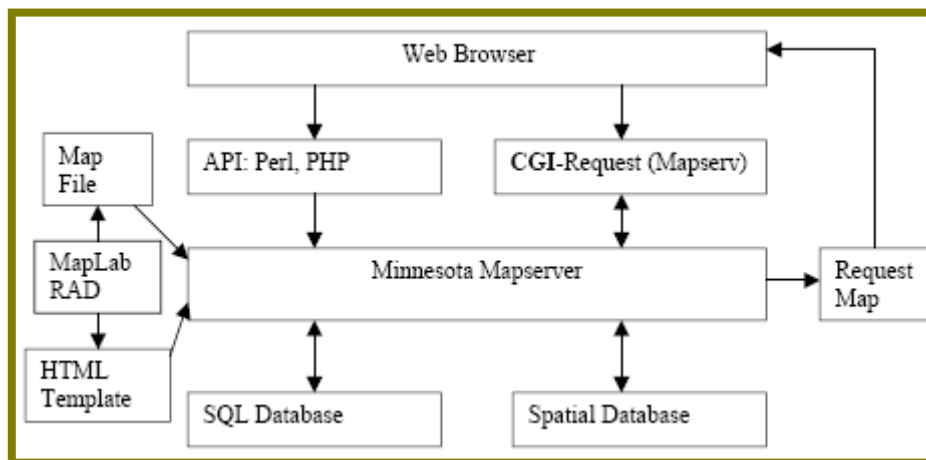
-  Un servidor http como Apache o Internet Information Server.
-  El Programa MapServer.
-  Un archivo de inicialización que lance la primera vista de una aplicación con MapServer (opcional).
-  Un Mapfile que controla el manejo que MapServer realizará con los datos.
-  Un archivo plantilla HTML que controle la interfaz de usuario de la aplicación con MapServer en la ventana del explorador de Internet.
-  Un conjunto de datos SIG.

FIGURA 5.2.1-1
ESTRUCTURA SISTEMA GIS WEB MAPSERVER



Algunas características del Mapserver son la que se exponen a continuación:

- ✚ Soporte para acceder a información geográfica en múltiples formatos vectoriales, raster y CAD.
- ✚ Soporte de los protocolos del OGC.
- ✚ Soporte Multiplataforma: Linux, Windows, UNIX y MacOS.
- ✚ Características avanzadas para la representación adecuada de la información: etiquetado, temáticos, simbología compleja.
- ✚ Un motor de renderización de las imágenes raster muy rápido.
- ✚ Posibilidad de programación en la parte del servidor de distintos lenguajes (PHP, Java, C#, Ruby, Perl.).

Dentro de las extensiones OGR que Mapserve posee, para este proyecto se debió utilizar virtual datos espaciales OVF el cual le permite conectarse a bases de datos que no tienen explícitamente los datos espaciales, así como texto plano archivos. Sus datos debe tener una X y una columna de Y, y los datos pueden acceder a través de una conexión ODBC o directamente a un puntero de archivo de texto.

Es importante destacar que el tipo de conexión utilizada en este trabajo es de tipo ODBC entendiéndose que es una abreviatura para la Conectividad abierta de bases de datos, y es una interfaz para acceder a bases de datos a través de


consultas SQL. El ODBC se puede utilizar como una herramienta de acceso a diversas bases de datos tales como MS-Access, dBase, DB2, Excel y Texto. A través de estas Nivel Interfaz de llamadas (CLI) de las especificaciones de SQL Acceso Group, el ODBC permite una forma neutral de acceso a los datos almacenados en los ordenadores personales y diversas bases de datos.

La fuerza de ODBC es que al ofrecer una interfaz de acceso universal de datos, permite a las compañías de software independientes y los partidos que no tienen que aprender múltiples interfaces de programación de aplicaciones. Para simplemente, con ODBC, las aplicaciones pueden acceder simultáneamente, ver y modificar la **base de datos** de numerosas y muy diversas bases de datos. Esto se debe a que el ODBC "volver a los códigos de" la consultas SQL de forma que sería leído por las distintas bases de datos.

Sin embargo, el ODBC también tiene sus inconvenientes. Como la gestión de un gran número de clientes ODBC puede significar una inmensa cantidad de controladores y archivos DLL que se ejecuta, puede dar lugar a un **sistema** de administración de gastos generales. No obstante, esta cuestión menor condujo a los usos adicionales de la tecnología de servidores de ODBC (o el "Multi-Nivel controladores ODBC") para aliviar la carga. Además, como los conductores son un factor clave en ODBCs, algunos han planteado la cuestión de los nuevos conductores de la estabilidad, como a menudo muchos de ellos han demostrado tener errores en ellas

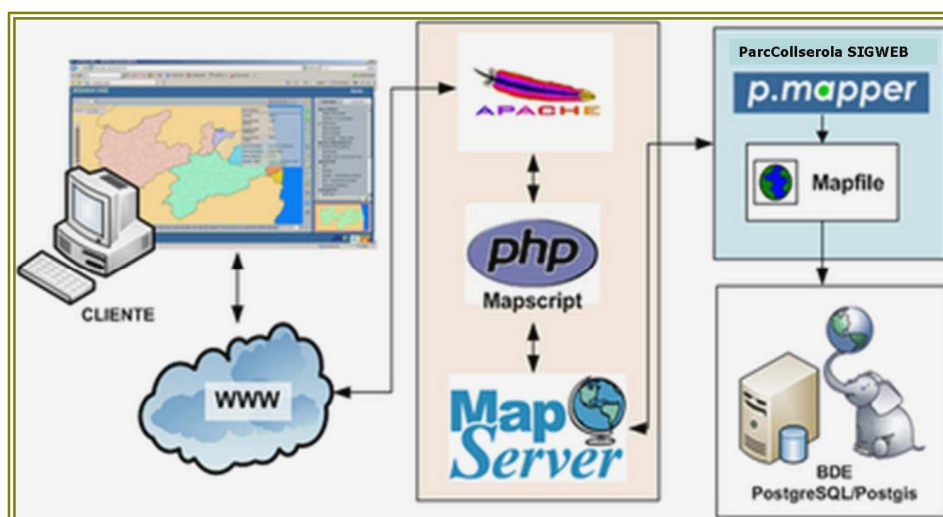
5.3 Entorno de desarrollo P.mapper.

P.mapper es un cliente ligero (se denomina cliente ligeros las aplicaciones que puedan acceder a servidores OGC con la utilización de un simple Browser o navegador por lo que es posible acceder a los servicios geomáticos desde cualquier ubicación con conexión a Internet), esta desarrollado con MapScrip _Php mas Mapserver que dispone de las siguientes características:

-  DHTML zoom/pan .Visores soportados: Mozilla/Firefox, IE, Opera, Konqueror.

- ✚ Pan/zoom con el ratón, movimientos con el teclado del cursor, mapa de referencia.
- ✚ Funciones de interrogación (identificación, selección, búsqueda).
- ✚ Muestra resultados de búsqueda de manera gráfica con enlaces a las base de datos relacionas e hyperlinks.
- ✚ Funciones de impresión: HTML y PDF.
- ✚ Configuración de múltiples funciones y vistas a través de un archivo INI.
- ✚ Creación directa de leyendas en formato HTML.
- ✚ Varios estilos de muestra de leyendas y tabla de contenidos.
- ✚ Posibilidad de multilinguaje.

FIGURA 5.3-1
ARQUITECTURA GENERAL DE INTEGRACIÓN DE P.MAPPER



FUENTE: Benigno M. 2007, Utilización de Mapserver en la gestión de los recursos hídricos del Estado de Paraíba, Brasil.





5.4 PostgreSQL.

Postgres fue desarrollado originalmente en el “Departamento de Ciencias de la Computación de la Universidad de California” en “Berkeley”, pionero en muchos de los conceptos de bases de datos relacionales orientadas a objetos que comienzan a estar disponibles en administradores de bases de datos comerciales. Ofrece soporte al lenguaje SQL92/SQL3, integridad de transacciones, y





extensibilidad de tipos de datos. PostgreSQL es un descendiente de dominio público y código abierto del código original de Berkeley.

Los sistemas de mantenimiento de “Bases de Datos relacionales tradicionales” (DBMS,s) soportan un modelo de datos que consisten en una colección de relaciones con nombre, que contienen atributos de un tipo específico. En los sistemas actuales, los tipos posibles incluyen numéricos de punto flotante, enteros, cadenas de caracteres, cantidades monetarias y fechas.

Está proyectado que este modelo será inadecuado para aplicaciones futuras de procesado de datos. El modelo relacional sustituyó modelos previos en parte por su simplicidad la cual también hace difícil la implementación de ciertas aplicaciones. Postgres ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema.

-  Clases.
-  Herencia.
-  Tipos.
-  Funciones.

Otras características aportan potencia y flexibilidad adicional:

-  Restricciones (Constraints).
-  Disparadores (triggers).
-  Reglas (rules).
-  Integridad transaccional.

5.5 PostGis.

PostGIS añade el soporte para objetos geográficos a la base de datos objeto-relacional PostgreSQL. PostGIS proporciona la capacidad espacial a un servidor PostgreSQL, permitiendo ser utilizado como un cliente GIS de la base de datos. Se puede afirmar que PostGIS más PostgreSQL forman una base de datos espacial donde se almacena y gestiona tanto la geometría de los elementos

geográficos como los atributos temáticos de los mismos. PostGIS permite al SGBD orientado a objetos PostgreSQL la gestión de objetos geográficos.

PostGIS consigue que el servidor de bases de datos PostgreSQL pueda manejar objetos geográficos capacitándolo para funcionar como soporte de datos espaciales en un Sistema de Información Geográfica. En este sentido, PostGIS se encontraría al nivel de la base de datos SDE de ESRI o al de la versión de Oracle con su correspondiente extensión espacial.

PostGIS sigue la norma OpenGIS referente a bases de datos SQL conocida como “Simple Features Specifications for SQL”. PostGIS ha sido desarrollado por Refrations Research Inc. como un proyecto de investigación en tecnologías de bases de datos en código abierto.

La lista de proyectos futuros incluye el desarrollo de tecnologías para carga y descarga de datos, la creación de herramientas para el acceso y manipulación directa de las bases de datos, y el soporte para topologías avanzadas, como pueden ser coberturas, redes y superficies, en entornos de servidor.

5.6 Quantum Gis.

Gran abanico de herramientas que pueden aportar gran ayuda al proyecto y utilización de la aplicación es predecible y amigable para poder realizar diversas tareas. La implementación de las capas es bastante completa, proporciona configuración, edición y visualización. También es una herramienta que proporciona ayuda a la hora de querer transformar un shapefile a un postGIS.

El Quantum GIS posee un módulo para generar mapas temáticos. Además utiliza las paletas del sistema operativo por lo que no tendréis problema a la hora de usar los colores que hayáis seleccionado. Sin embargo, al no generar rangos de colores de forma automática, tendréis que hacerlo manualmente y desarrollar de paso, vuestras capacidades artísticas. Algunas características de estas herramientas son:

- ✚ La aplicación permite manejar formatos **raster** (GRASS GIS, GeoTIFF, TIFF, JPG, etc.) y vectoriales (Shapefile, Mapinfo, GRASS GIS, etc.).

Además permite la creación de polígonos, líneas y puntos georeferenciados.

✚ El programa está en **C++** lo que le reporta una asombrosa velocidad según comentan los que lo han utilizado.

✚ Incluye es la exportación de los mapas que habéis creado para ser automáticamente servidos desde el servidor OGC WMS MapServer. Este servidor es un entorno de desarrollo para construir el espacio habilitado para las aplicaciones de Internet. MapServer sobresale principalmente por su prestación de datos espaciales (mapas, imágenes y datos vectoriales) para la web.

✚ Manejo y lectura de GPS, así como para los sistemas de localización por satélite, este dispone de funciones orientadas a este sistema. Es posible bajar puntos de referencia desde un receptor GPS, importar sus datos y crear distintas capas.

Esta herramienta también presenta Algunas limitaciones a considerar como son:

✚ La opción de georeferenciar no es eficaz. Esta limitado a solo una manera de rectificar las fotografías y es sabiendo previamente las coordenadas de mínimo dos punto de la imagen cargada. No estaría de más que se pudiera georeferenciar sabiendo la escala o puntos de referencia de las esquinas de la imagen, en cualquier caso la herramienta es eficiente.

✚ A la hora de importar un shepfile a PostGis esta herramienta presenta un inconvenientes y es que el OIDS (identificador de objeto) no esta activado por lo cual debe ser activado, lo cual se deberá hacer mediante programación de código.

VI. PROCEDIMIENTOS E IMPLEMENTACION.

En este capítulo se presentarán los procedimientos y las aplicaciones que se utilizaron para poder lograr con los objetivos del proyecto que se plantearon al inicio del presente trabajo.

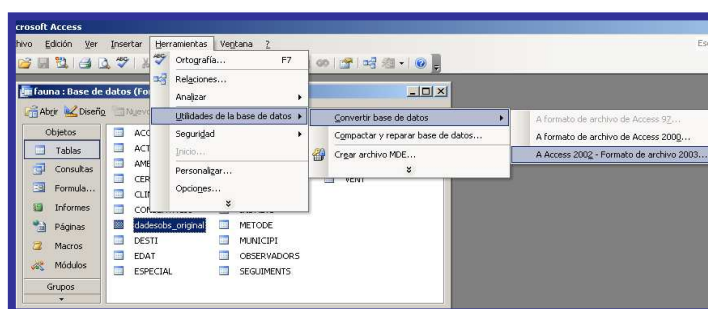
Se describirán brevemente los procedimientos que fueron empleados así como su implementación en la web con el objetivo de que sirva como referencia a los investigadores y usuarios que se encuentran inmersos en este tipo de trabajos relacionados con los sistemas de información geográfica.

6.1 Migración de bases de datos de Microsoft Access a PostgreSQL.

Para realizar la migración de las tablas de Microsoft Access a PostgreSQL se tuvieron que realizar unos pasos antes de llegar allí, estos pasos fueron los siguientes:

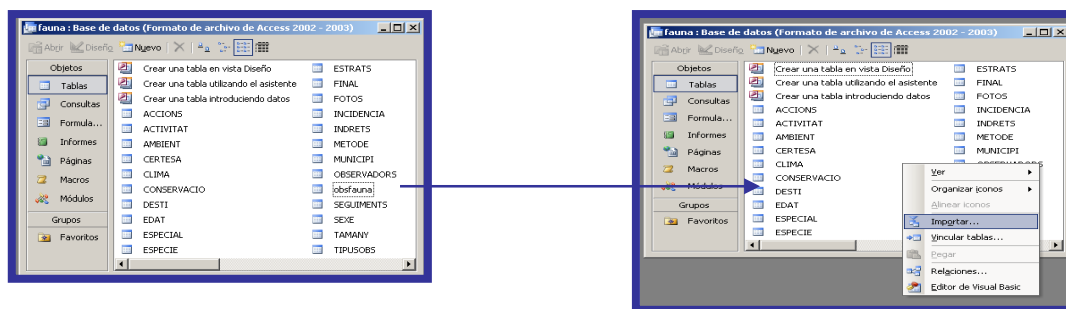
- ✚ La base original de los registros de fauna se encontraban en Microsoft Access 95, por lo cual se debió exportar a Access 2003.

FIGURA 6.1-1
EXPORTACIÓN DE ACCESS 95 A 2003



- ✚ La base de datos que contenía toda la información recolectada (osbe fauna) se debió pasar a un archivo de Access.

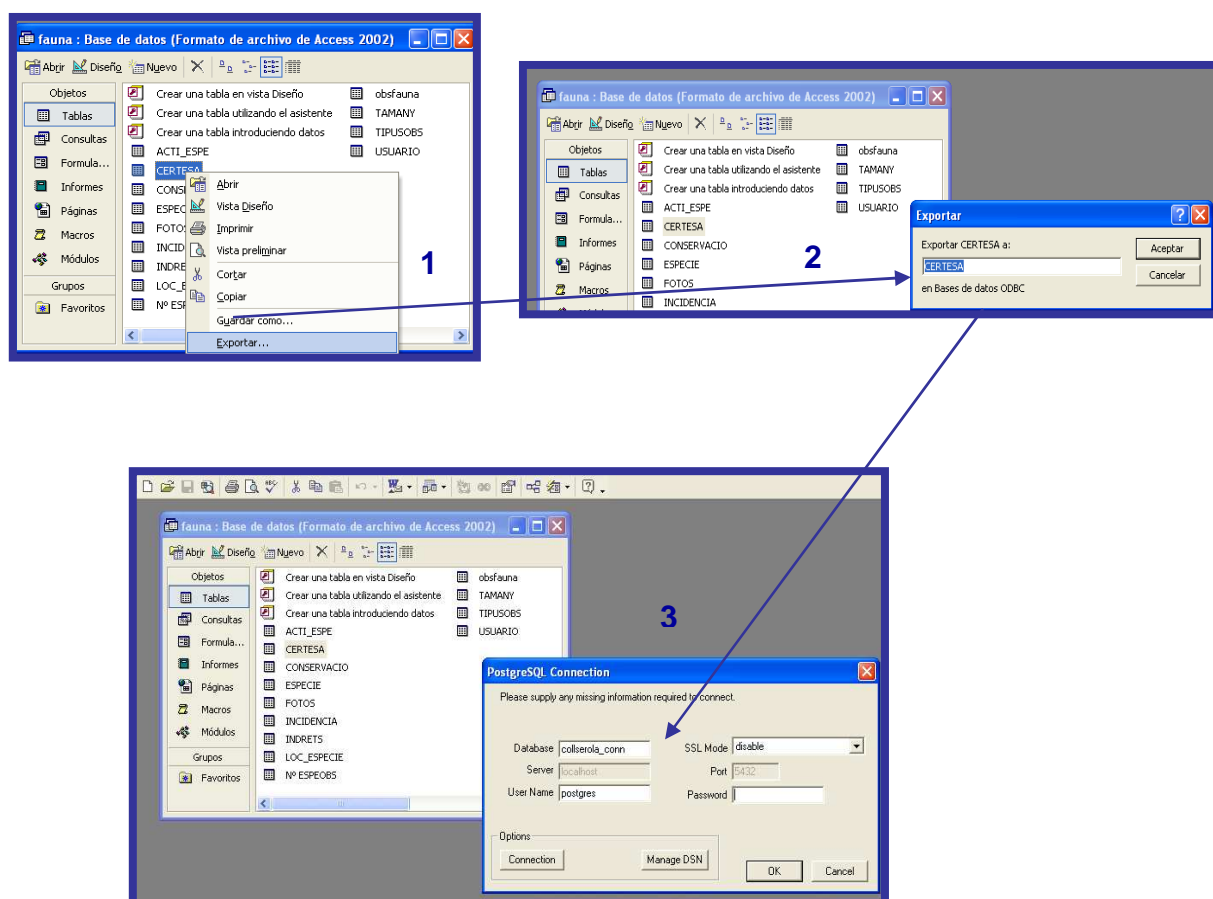
FIGURA 6.1-2
EXPORTACIÓN DE LA BASE DE DATO
A ACCESS.



➤ Luego de ser exportadas las tablas se tomo la decisiones de cuales deberían ser utilizadas para el formulario y cuales de ellas irían con información y cuales no.

➤ Por ultimo se exportaron una a una las tablas y se guardaron como una base de datos ODBC para luego hacer la conexión con el PostgreSQL.

FIGURA 6.1-3
CONEXIÓN DE LA BASE DE DATOS
ODBC A POSTGRES

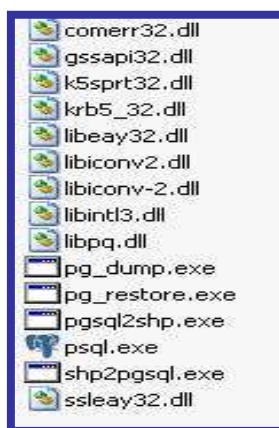


6.2 migración de Shapefile a PostGis.

Para la imigración de los archivos *Shapefile* a PostGis como se menciona anteriormente en el punto donde se habla sobre el quantumGis se trato de hacer por esta herramienta pero no se pudo, por lo que se procedió a realizarse de la siguiente manera:

- ✚ Se utilizo la consola de comandos de Windows la cual se localiza en:
Inicio → Programas → PostgreSQL 8.3 → Command Prompt.
- ✚ Se ejecutaron una par de programas que se encuentran en la carpeta de instalación del PostgreSQL ubicada en: C/Archivos de programas/postgresql/8.3/bin. Estos programas son los que se m muestran a continuación:

FIGURA 6.2-1
ARCHIVOS UTILIZADOS PARA LA MIGACIÓN DE SHAPEFILE
A POSTGIS.



- ✚ De estos archivos.exe solo dos de ellos son los que se ejecutarán los cuales son: shp2pgsql y el otro es *psql.exe*.
- ✚ se ejecuta el archivo shp2pgsql como primer paso.

- ✚ Para pasar el sql creado anteriormente se debe ejecutar *psql.exe* para que sean enviadas las tablas creadas al PostgreSQL.

FIGURA 6.2-4
ENVIO DE LAS TABLAS CRADAS DE LOS SHAPEFILE AL
POSTGRESQL

```
psql: FATAL: la autenticación password falló para el usuario «postgres»
C:\ParcCollserora\capascollserola>c:/Mutili/psql -d MARY -h localhost -U postgres
Password for user postgres:
psql: FATAL: la autenticación password falló para el usuario «postgres»
C:\ParcCollserora\capascollserola>c:/Mutili/psql -d MARY -h localhost -U postgres
BEGIN
psql:comarques.sql:10: NOTICE: CREATE TABLE crear  una secuencia impl cita «comarques»
psql:comarques.sql:10: NOTICE: CREATE TABLE / PRIMARY KEY crear  el  ndice impl cito
CREATE TABLE
        addgeometrycolumn
        public.comarques.the_geom SRID=23031 TYPE=MULTIPOLYGON DIMS=2
<1 row>
INSERT 50230 1
INSERT 50232 1
```

Donde:

- **d:** es el nombre de la base de dato.
- **h:** es el nombre del servidor.
- **U:** el nombre de la herramienta a utilizar.

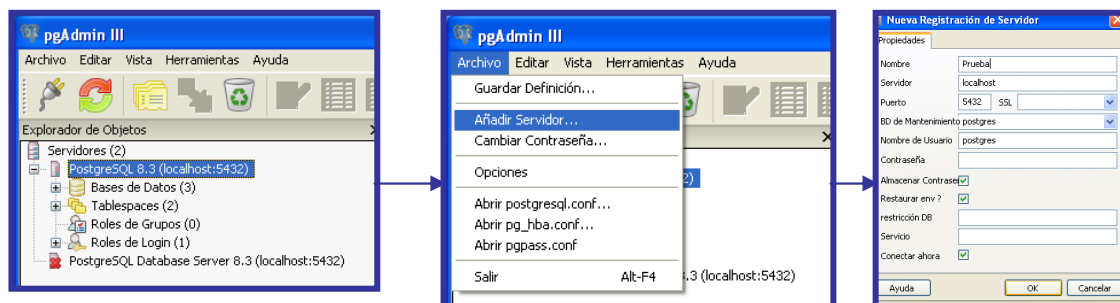
6.3 Creaci n y preparaci n de la base de datos.

Para la creaci n de la base de datos se utilizo el gestor de pgAdmin del PostgreSQL.

Se mostrara a continuaci n los pasos que se deben realizar para su creaci n desde este tipo de gestor:

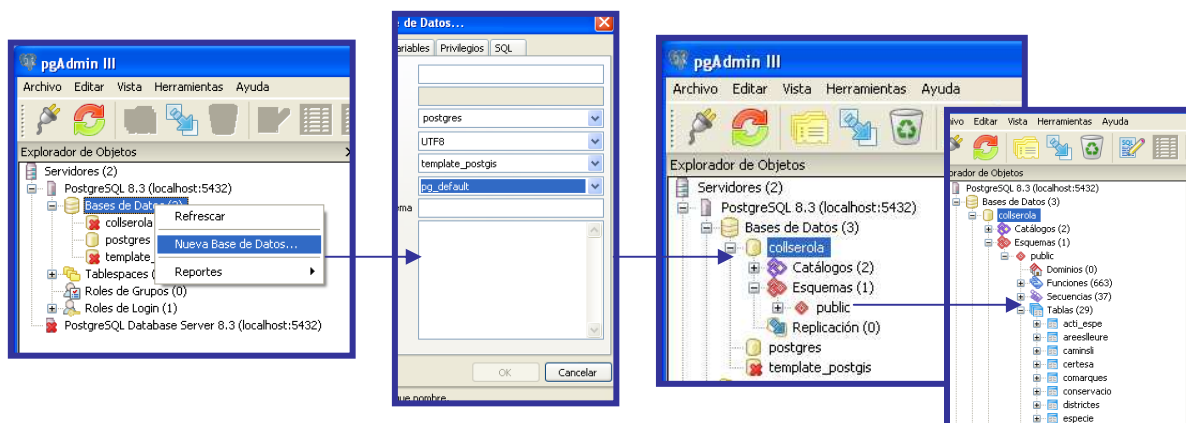
- ✚ Se crea un servido la cual estar  integrado por una base de datos, tablespaces, roles de Grupos, roles de login.

FIGURA 6.3-1
ENVIO DE LAS TABLAS CREADAS DE LOS SHAPEFILE AL
POSTGRESQL



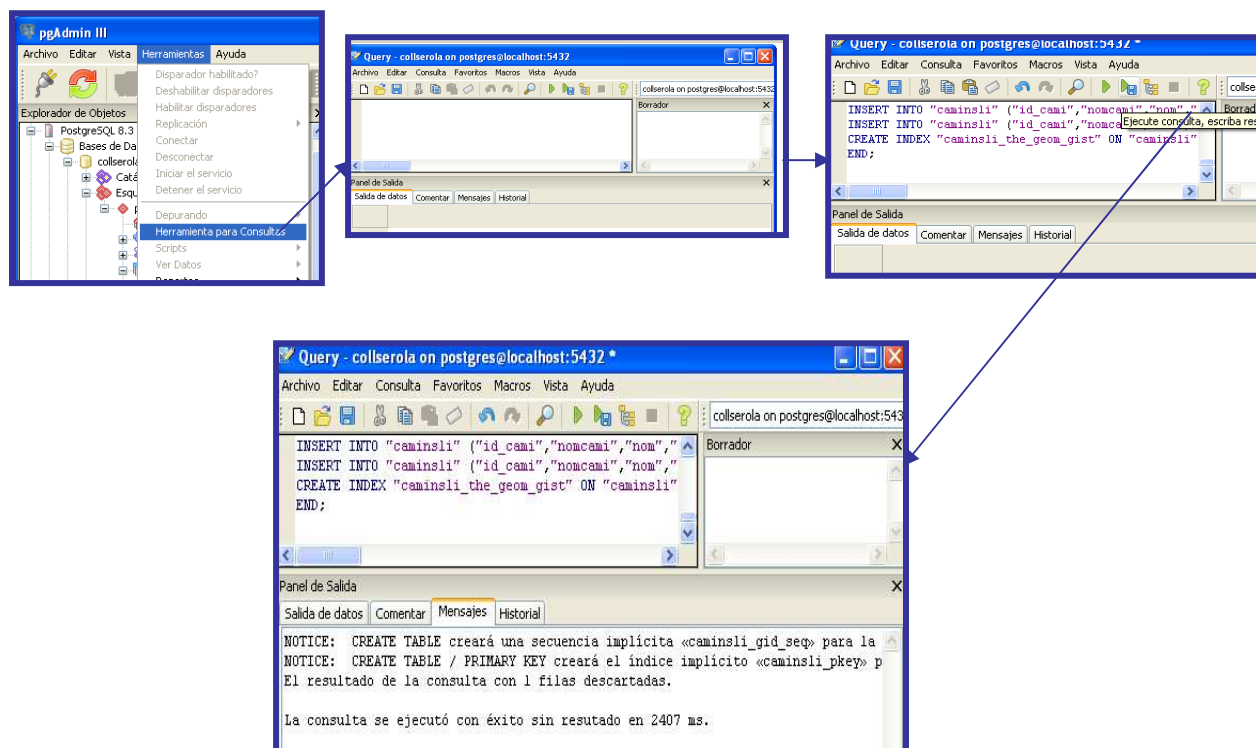
- Dentro de la base de datos anteriormente creada se debió incorporar una nueva base de datos que es la que contiene las tablas que se utilizaran. Esta base de dato nueva posee inmersamente los siguientes comandos catálogos, esquemas y replicación, dentro de esquema se encuentra una carpeta llamada public y dentro de esa carpeta se ubican la de las tablas.

FIGURA 6.3-2
UBICACIÓN DE LAS TABLAS DENTRO
DEL POSTGRESSQL



- Para ubicar los shapefile que se han exportado al PostGis a pgAdmin se debe copiar el archivo. sql esto se deberá hacer si la exportación fue hecha antes de crear la base de datos en PostgreSQL.

FIGURA 6.3-3
MONTAJE DE SCRIPT DE SQL A POSTGRES



6.4. Interfaz del administrador de la base de datos.

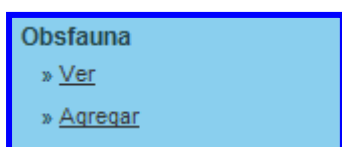
Para realizar el administrador de la base de datos se deben tomar muchos parámetros previos que solamente puede ofrecernos los técnicos que manejan dicho administrador. Esta herramienta se crea con los objetivos de poder editar, eliminar y agregar datos siempre que se necesite y a su vez este administrador solo podrá ser tocado por la persona en cargada de la administración de la base de datos del parc de Collserola.

En esta figura 6.4-1 podemos observar a mano izquierda un menú con las tablas que conforman la base de datos. Para cada una estas tablas el administrador nos permitirá visualizar, editar o bien eliminar su contenido.

FIGURA 6.4-1
ADMINISTRADOR DE LA BASE DE DATOS



Como se observa en la figura, cada una de las tablas contiene las siguientes funcionalidades:



- “Ver”: Permite visualizar cada uno de los campos y filas de la tabla seleccionada. Además podremos editar o eliminar cualquiera de las filas mediante las opciones de Acción (“Editar”/”Eliminar”).
- “Agregar”: Nos permitirá añadir una nueva fila en la tabla seleccionada.

FIGURA 6.4-2
ESTRUCTURA DE LA TABLA OBSFAUNA EN LA INTERFAZ
DEL ADMINISTRADOR

Obsfauna				
				Agregar nuevo
Nombre especie	Distancia	Fecha	Hora	Acción
daadaadadadad	3 mt	27/01/2009	15:49	Editar Eliminar
hbxxgxfxfg	7	7	15:54	Editar Eliminar
nuria	2mt	22/07/08	12:37	Editar Eliminar
loro	2 mt	6/02/2009	14:47	Editar Eliminar

FIGURA 6.4-3
EDICION DE UNA FIILA DE LA TABLA OBSFAUNA EN LA
INTERFAZ DEL ADMINISTRADOR

Agregar Obsfauna

Nombre de la especie	<input type="text"/>	(Obligatorio, max 50 caracteres)
Distancia	<input type="text"/>	(Obligatorio, max 50 caracteres)
Fecha	<input type="text"/>	(Obligatorio, max 50 caracteres)
Hora	<input type="text"/>	(Obligatorio, max 50 caracteres)
Usuario	cccc cccc <input type="button" value="v"/>	
Tamany	Grande <input type="button" value="v"/>	
Lloc	Acera <input type="button" value="v"/>	
Especie	ww <input type="button" value="v"/>	
Acti Espe	Alimentandose <input type="button" value="v"/>	
Incidencia	Escapado / Introducido <input type="button" value="v"/>	
Conservacio	Aplastado <input type="button" value="v"/>	
Certesa	Posible <input type="button" value="v"/>	
tipusobs	Cae / Nido <input type="button" value="v"/>	
<input type="button" value="Cancelar"/> <input type="button" value="Guardar"/>		

6.5 Funcionalidades implementadas.

El diseño de la aplicación busca que el usuario que acceda a ella no requiera de ser un especialista en SIG para poder consulta e informar una incidencia faunística dentro de las áreas el Parc de Collserola. Para ello tanto la manera de acceder como la utilización de las herramientas que aparecen en la aplicación será sencilla.

El acceso al WebSIG ParcCollserola se realizará a través de la página principal del Consorci, (<http://www.parccollserola.net>) mediante un acceso HTTP. Una vez seleccionado el enlace al WebGIS se abre la vista principal de la aplicación. En esta página se puede elegir el idioma de la aplicación.

Para la publicación web del proyecto es necesario inicialmente instalar un Sistema de Gestión de Bases de Datos que en nuestro caso serán el PostgreSQL y

PostGIS, debido a que se trabajo exportando algunos datos geográficos que se encontraban en formato shapefile a PostGIS. La gestión de la información alfanumérica necesaria para el visualizador se deberá realizar a con la base de dato asociada.

Es importante destacar que se contaba con un cúmulo de información en formato shapefile.

Para realizar la exportación de shapefile a PostGIS se tomaron ejemplos que anteriormente estaban hechos en otros visores de mapas como itinerarios de bicicleta, para ello se realizaron las llamadas desde la base de datos creada y se fueron montando diferentes tipos de geometría como líneas, polígonos y puntos como se puede observa en cada ejemplo.

FIGURA 6.5-1
REPRESENTACION DE LINEAS Y POLIGONOS

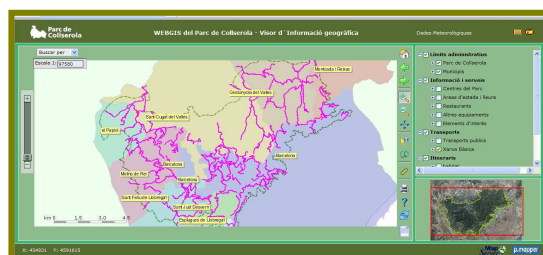
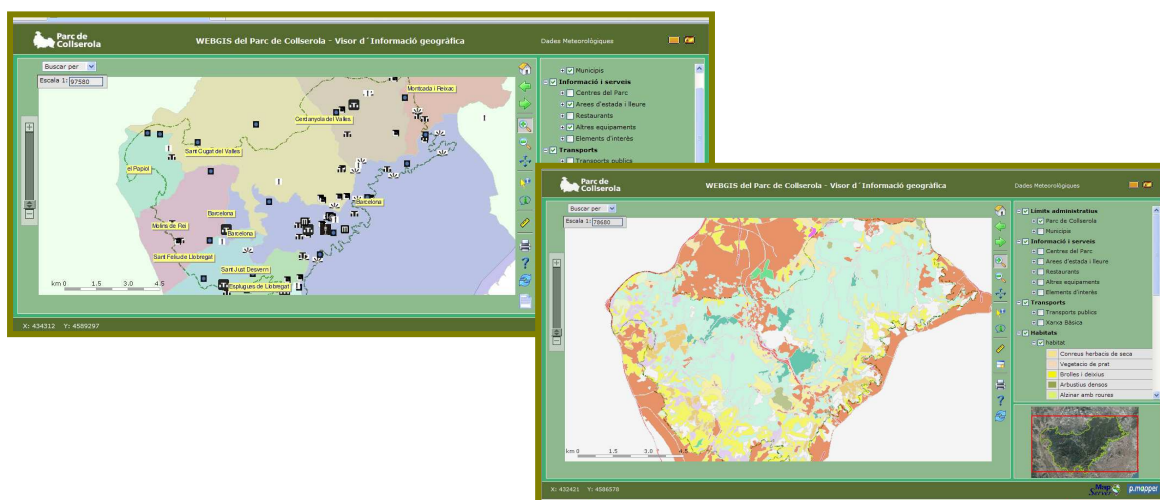


FIGURA 6.5-2
REPRESENTACION DE LINEAS, POLIGONOS Y PUNTOS.



El objetivo de la aplicación es que el usuario pueda registrar una incidencia faunística y poder utilizar el mapa para georeferenciar la localización del acontecimiento así como visualizar toda la información del Parc de Collserola.

Para ello lo primero que se realizó, fue crear unos formularios previamente en HTLM luego pasados a PHP, el primero de ello es para que el usuario se registre con el objetivo que este quede guardado en la base de dato y no se repita el nombre del usuario cada vez que entre a reportar la incidencia.

Este formulario de registro esta compuesto por 5 campos que son obligatorios y 4 que no lo son. Los primeros los constituyen, el nombre, apellido, E_mail, contraseña y la confirmación de la contra seña y los segundo por domicilio, código postal localización y teléfono.

FIGURA 6.5-3
REGISTRO DEL USUARIO



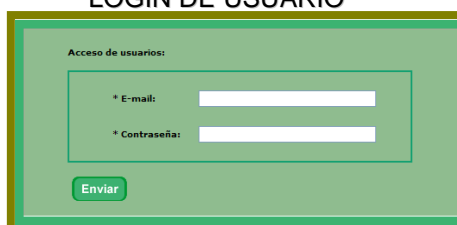
Formulario de registro de usuario con los siguientes campos:

- * Nombre:
- * Apellido:
- * E-mail:
- * Contraseña:
- * Confirmar Contraseña:
- Domicilio:
- Código Postal:
- Localidad:
- Teléfono:

Botón:

En segundo lugar una vez registrado se abre una pagina que pide que te logues, este una vez logado podrás acceder al formulario de incidencias faunística y este login sirve para llevar controles en la base de datos y también le permite al usuario entra sin tener que nuevamente registrarse.

FIGURA 6.5-4
LOGIN DE USUARIO



Formulario de login de usuario con los siguientes campos:

- * E-mail:
- * Contraseña:

Botón:

Como último se creo el formulario de incidencias, este se encuentra estructurado con información suministrada por el consorcio de parque de Collserola, la información fue la elegida por los técnicos tomando en cuenta lo que se quería obtener de la información de incidencias faunística, el usuario podrá seleccionar solamente y en otra debe escribir su observación, en este formulario también existe un campo que es el de localización de la especie este campo se rellenara con la interacción del mapa y el usuario.

FIGURA 6.5-5
FORMULARIO DE INCIDENCIAS FAUNÍSTICAS

Este formulario tiene el objetivo de proporcionar al usuarios de parque, un instru incidencias faunísticas.

INFORMACIÓN GENERAL DE LA OBSERVACIÓN

Grandària:	Grande
Tipo de la observación:	Cae / Nido
Localización:	Acera
Nom de la especie:	ww
Actividad de la especie:	Alimentandose
Incidencias:	Escapado / Introducido
Conservacion:	Aplastado
Certesa:	No información
tipusobs:	Cae / Nido

El mapa posee sus propias herramientas de navegación que son para aumentar el zoom, disminuirlo, moverse hacia los lados, o centrar el mapa en la posición inicial. También podremos acceder a la información alfanumérica asociada a los objetos a través de los elementos visuales (pintados en el mapa), o a través de unas listas cargadas con todos los elementos a seleccionar. Cuando se accede a la información a través de un elemento visual, o a través de una lista, el elemento gráfico deberá cambiar de color para representar que está siendo seleccionado. Todas estas herramientas existen con anterioridad.

La incorporación del punto de coordenadas como una herramienta nueva de la aplicación tendrá como objetivo que el usuario haga clic sobre el visor del mapa para ubicar previamente el sitio donde observe la incidencia, esta información se cargará directamente en el formulario y luego el usuario procederá a rellenar toda la información que se encuentra en dicho formulario.

De esta manera la información será guardada directamente en la base de datos del PostgreSQL.

Para que esta herramienta fuese efectiva y cumpliera con su objetivo se implemento una herramienta que posee el mapserve que se denomina OFV, el código de esta herramienta fue ejecuta en el archivo map del Collserola, para que pudiera dibujar el punto de la incidencia.

Luego en el archivo de javascript /pm_js se indicaran las rutas en donde debería visualizarse la coordenadas localizadas, debido a que existen varias formas de poder entrar al formulario de incidencias se tuvieron que incluir en los formularios de incidencia, login y de registro una sentencias que permitieran que una vez que el usuario pinchase en el mapa y este cargara las coordenadas no se perdiera y el usuario pudiera si no estaba registrado registrarse y poder continuar suministrando la información de la incidencia observada.

VII. CONCLUSIONES

Para lograr resultados óptimos, actualmente se cuenta con múltiples herramientas tecnológicas, destacando sobre todas, los Sistemas de Información Geográfica cada vez más populares, ya no sólo entre los usuarios y desarrolladores del área de la Geomática.

Esta popularidad se debe, principalmente, a la masificación de la Internet y al surgimiento de herramientas “open source” que al ser de libre disposición, acercan al usuario normal a estas tecnologías y permiten a las instituciones reducir costos o destinarlos a otras áreas.

El manejo de la información en formato digital admite el cruce de distintos tipos de datos, facilitando análisis y proyecciones, como su disponibilidad a un número mayor de usuarios, complementando diferentes áreas que utilizan información en común.

Si esta información es administrada por un GIS, el cual permite al acceso a múltiples usuarios de distintas áreas y niveles, se facilitará a éstos sus tareas, ya sea, de análisis, desarrollo o simplemente consultas de información.

La aplicación obtenida, al ser de libre disposición vía WEB, facilita el acceso a este proyecto a una gran cantidad de personas.

Las herramientas “open source” como MapServer y sus complementos mencionados, permiten crear aplicaciones como la tratada en este trabajo, la cual puede y debe continuar su desarrollo.

Los objetivos planteados fueron alcanzados prácticamente a cabalidad.

En cuanto a la aplicación y funcionalidad de ésta, cumple plenamente con las expectativas de diseño y, a las solicitudes y sugerencias hechas por parte del consorcio parc de Collserola.

Así mismo, queda a disposición de la comunidad una aplicación informativa interactiva, disponible desde cualquier computador con acceso a Internet, en cualquier lugar del mundo.

VIII. BIBLIOGRAFIA.

- Cuadro tomado de :Fuster M. et al, “Divulgación de información turística a través de Internet “El proyecto SITTUR”.
- ESRI. “ESRI Shapefile Technical Description”. Reporte técnico disponible en Internet: <http://www.esri.com>, 2008.
- Korth, H. y A. Silberschatz. “Fundamentos de bases de datos.” 1993. Mc Graw Hill. ISBN 84-481-0079-4.
- Sitio web de PostgreSQL: <http://www.PostgreSQL.org>, 2008.
- <http://mapserver.gis.umn.edu/>.
- Imagen tomada de <http://learners.in.th/file/rpggunner/view/14277/mapserver.jpg>.
- Sitio web de PostGIS: <http://postgis.refrations.net/>. 2008.
- Sitio web de Open GIS Consortium: <http://www.opengis.org>. Ultima visita:30/7/2008.
- Giosa, D., L. Tansini. “Interoperabilidad en Sistemas de Información. Geográfica”. No publicado. Instituto de Computación. Universidad de la Republica Oriental del Uruguay. 1999.
- Open GIS Consortium. “Simple Features for Specification for SQL”. Estándar publicado por Open GIS Consortium. Disponible en Internet: <http://www.opengis.org/techno/specs/99-049.pdf>, 2008.
- Genexus. Sitio web de ARTech: <http://www.artech.com.uy>. Ultima visita 2008.
- <http://www.arquitectuba.com.ar/software-gratis/quantum-gis/>,2008.
- http://es.wikipedia.org/wiki/Apache_http_server,2008.
- P.MAPPER, <http://www.pmapper.net/>, 2008.

IX. ANEXOS

