

**DISEÑO E IMPLEMENTACIÓN PARA UN PROTOTIPO DE SISTEMA DE
INFORMACIÓN GEOGRÁFICO BAJO AMBIENTE WEB, PARA LOS
RECURSOS FÍSICOS Y TÉCNICOS DE LA SECCIONAL SOCORRO DE
LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.**

**RENZO ANDRES CAMARGO TABARES
JOSE EVERARDO RODRIGUEZ ARDILA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO – MECANICAS
ESCUELA DE INGENIERIA CIVIL
BUCARAMANGA
2004**

**DISEÑO E IMPLEMENTACIÓN PARA UN PROTOTIPO DE SISTEMA DE
INFORMACIÓN GEOGRÁFICO BAJO AMBIENTE WEB, PARA LOS
RECURSOS FÍSICOS Y TÉCNICOS DE LA SECCIONAL SOCORRO DE
LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.**

RENZO ANDRES CAMARGO TABARES
JOSE EVERARDO RODRIGUEZ ARDILA

Proyecto de Grado para optar el título de Ingeniero Civil

Director

Ingeniero Carlos Alonso Camargo Mantilla

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO – MECANICAS
ESCUELA DE INGENIERIA CIVIL
BUCARAMANGA
2004**

Dedico este libro a mis padres, porque el presente manuscrito, el cual es fruto de mi esfuerzo personal, también lo es de ellos.

A mi hermana, que con su ejemplo de constancia, dedicación y esfuerzo, ha logrado inspirarme para alcanzar el anhelado sueño de convertirme en Ingeniero.

Renzo

Este es el primer gran logro que obtengo en mi vida profesional, y por eso quisiera dedicárselo muy especialmente a mi viejo y a mi vieja, que han sido dos padres excelentes y que han tenido la paciencia suficiente para esperar y soportar en este largo camino. También quisiera dedicárselo a mis hermanos por su apoyo incondicional, el cual muchas veces me llenó de energías para seguir trabajando.

José

AGRADECIMIENTOS

Agradezco muy especialmente a mis padres y hermana, pues sin sus enseñanzas nunca habría de lograr todo lo que he alcanzado hasta el día de hoy. También agradezco a la Ingeniera Luz Marina Duarte G. por su paciencia y colaboración; pues sin su apoyo y guía permanente, el camino habría sido mucho más arduo.

De manera muy especial quisiera agradecer al Ingeniero Erwin Meza por su brillante y oportuna intervención, la cual nos guió de forma expedita hasta el final del recorrido.

De igual manera agradezco enormemente a mis amigos y amigas, quienes siempre supieron tenderme una mano en los momentos de dificultad y me acompañaron en las duras y las maduras.

Por último, al grupo 5 F.C.UIS y especialmente a su jefatura, por las enseñanzas que me han brindado y que me ayudaron enormemente a llegar hasta donde me encuentro... gracias a todos.

Renzo

AGRADECIMIENTOS

Gracias, a todos aquellos que confiaron en mí. Gracias por esperar tan anhelado logro. Gracias por aguantar tantas cosas durante el desarrollo de este proyecto, pero especialmente gracias a mis padres que siempre depositaron en mí, la confianza necesaria para que las cosas me salieran bien.

Por último me gustaría darles las gracias a todos mis amigos, por su apoyo incondicional, especialmente a Albita y a Oscar.

José

TABLA DE CONTENIDO

	Pág.
INTRODUCCIÓN.....	1
1. SISTEMAS DE INFORMACIÓN GEOGRÁFICA	3
1.1 DEFINICIÓN DE SIG	3
1.2 COMPONENTES DE UN SIG.....	4
1.2.1 Equipos (Hardware)	4
1.2.2 Programas (Software)	4
1.2.3 Datos.....	5
1.2.4 Recurso Humano.....	5
1.3 MODELOS DE DATOS	5
1.3.1 Modelo Vectorial	5
1.3.2 Modelo Raster.	6
1.3.3 Modelo orientado a objetos.	6
1.4 BASE DE DATOS GEOGRÁFICA.....	6
1.5 FUNCIONES BÁSICAS DE UN SIG	7
1.6 APLICACIONES DE UN SIG	8
2 BASES DE DATOS	10
2.1 DEFINICIÓN DE BASE DE DATOS.....	10
2.2 MODELO CONCEPTUAL.....	10

2.3	MODELO LÓGICO.....	11
2.4	MODELO ENTIDAD RELACIÓN	12
2.5	CLASIFICACIÓN DE LAS RELACIONES	13
2.5.1	Relaciones Uno a Uno (1:1)	13
2.5.2	Relaciones Uno a Muchos (1: M)	13
2.5.3	Relaciones Mucho a Muchos (M: M)	13
3	DESCRIPCIÓN DEL SOFTWARE EMPLEADO	14
3.1	HERRAMIENTAS DE SOFTWARE LIBRE DISPONIBLE	14
3.1.1	Servidores Web	14
3.1.2	Servidores de Bases de Datos.....	17
3.1.3	Servidores de mapas	21
3.1.4	Lenguajes de programación para páginas Web	24
3.2	LINUX RED HAT 9.0	29
3.3	APACHE VERSIÓN 2.0	31
3.4	PHP VERSIÓN 4.2.2	32
3.5	POSTGRESQL VERSIÓN 7.3.2	33
3.6	POSTGIS VERSIÓN 0.7.5.....	35
3.7	MAPSERVER VERSIÓN 3.6.6.....	35
3.8	PHP/MAPSCRIPT VERSIÓN MAPSERVER 3.6.6.....	36
4	METODOLOGÍA	38
4.1	RECOLECCIÓN Y PROCESAMIENTO DE LA INFORMACIÓN ..	38

4.2	DISEÑO DE LA BASE DE DATOS	39
4.2.1	Modelo Conceptual	40
4.2.2	Modelo Lógico	40
4.2.3	Diccionario de Datos	41
4.2.4	Modelo Físico	41
4.3	DEFINICIÓN DE TOPOLOGÍAS	41
4.3.1	Tipos de Topologías	42
4.3.2	Creación de Topologías	45
4.3.3	Exportar como archivos Shape	48
4.4	METODOLOGÍA DE PROTOTIPOS	51
4.4.1	Determinación de Requerimientos	52
4.4.2	Desarrollo del Prototipo	52
4.4.3	Revisión del Prototipo	53
4.5	METODOLOGÍA DE INGENIERÍA WEB	54
4.6	DESCRIPCIÓN SITIO WEB	56
5	DISEÑO Y DESARROLLO DEL SISTEMA	59
5.1	Descripción del Sistema	59
5.2	Requisitos del Sistema	62
6	CONCLUSIONES	64

LISTA DE FIGURAS

Figura 1. Creación de Topología: Menú Map	47
Figura 2. Creación de topología: Caja de diálogo	48
Figura 3. Exportar como Shape: Menú Map	49
Figura 4. Exportar como Shape: Caja de diálogo	50
Figura 5. Directorio que contiene los archivos del sitio	57
Figura 6. Directorio que contiene los shapes	58

LISTA DE ANEXOS

ANEXO A.....	67
ANEXO B.....	83
ANEXO C.....	136
ANEXO D.....	140
ANEXO E.....	159
ANEXO F.....	176

RESUMEN DEL PROYECTO

TITULO

DISEÑO E IMPLEMENTACIÓN PARA UN PROTOTIPO DE SISTEMA DE INFORMACIÓN GEOGRÁFICO BAJO AMBIENTE WEB, PARA LOS RECURSOS FÍSICOS Y TÉCNICOS DE LA SECCIONAL SOCORRO DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER*.

AUTORES

Renzo Andrés Camargo Tabares
Jose Everardo Rodríguez Ardila**

PALABRAS CLAVE

Open source, Servidor, Aplicaciones Web, Información Geográfica, SIG.

RESUMEN

Los Sistemas de Información Geográfica (SIG) son una tecnología que se ha desarrollado a través de los tiempos para ayudar a la toma de decisiones de toda índole, incluyendo las de tipo administrativo. Tomando en cuenta esta situación, se ha elaborado el presente proyecto de grado.

Este trabajo es el resultado de un gran esfuerzo y un sinnúmero de tareas; las cuales incluyen investigación, consulta, recolección de información, digitalización de la misma, conceptualización de la realidad, implementación de tecnología open source avanzada para el despliegue de información geográfica, programación y diseño de aplicaciones **web**, entre otras. Todas estas tareas estuvieron enfocadas e integradas para el desarrollo de un prototipo de Sistema de Información Geográfica (SIG) que ayudara a la gestión de los recursos físicos de la Universidad Industrial de Santander, Sede Socorro (Santander).

Esta no es más que la primera etapa de un gran proyecto que pretende expandir la cobertura de un Sistema de Información Geográfica (SIG) que permita no solamente la gestión de recursos físicos sino también la gestión de recursos humanos y técnicos de la totalidad de la planta física de la Universidad Industrial de Santander.

Una de las conclusiones principales a las que se llegó mediante el trabajo realizado, es la necesidad de continuar invirtiendo esfuerzos en el desarrollo extensivo del prototipo, para cumplir los alcances a largo plazo del proyecto.

* Proyecto de Grado.

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Civil. Universidad Industrial de Santander.
Director Ing. Carlos A. Camargo.

ABSTRACT

TITLE

IMPLEMENTATION AND DESIGN OF A WEB DEVELOPED GEOGRAPHIC INFORMATION SYSTEM PROTOTYPE, FOR THE TECHNICAL AND PHYSICAL RESOURCES OF THE UNIVERSIDAD INDUSTRIAL DE SANTANDER SECTIONAL IN SOCORRO*.

AUTHORS

Renzo Andrés Camargo Tabares
Jose Everardo Rodríguez Ardila**

KEYWORDS

Open source, Server, Webapps, Geographic Information, GIS.

ABSTRACT

Geographic Information Systems (GIS) are a technology which has been developed through time to aid decision making of all sorts, including the administrative kind. Taking into account this situation, the present undergraduate thesis has been elaborated.

The present undergraduate thesis is the result of a countless number of tasks which include research, consultation, information gathering, digitalization of this information, reality conceptualization, advanced open source technology implementation for the display of geographic information, programming and web application design, among others. All of these efforts were made to develop a Geographic Information System (GIS) prototype which would aid the management of the physical resources of the Universidad Industrial de Santander sectional, in Socorro, Santander.

This is no more than the first stage of a much larger project which seeks after the expansion of a Geographic Information System (GIS) coverage. This final GIS attempts to aid not only the physical resources management, but also the human and technical resources of the whole physical roster of the Universidad Industrial de Santander.

One of the main conclusions reached through the process of developing the prototype subject of this project, is the imperative need to continue the efforts in the furthermore elaboration of the prototype, in order to achieve the long term objectives of the global project.

* Undergraduate thesis.

** Faculty of Physical and Mechanical Engineering. Department of Civil Engineering. Universidad Industrial de Santander. Director Eng. Carlos A. Camargo.

INTRODUCCIÓN

Los Sistemas de Información Geográfica han presentado un desarrollo exponencial durante los últimos años, expandiendo sus posibilidades de trabajo hasta ver su aplicación incursionando en casi todos los campos de las ciencias básicas, políticas, sociales y naturales.

Es por esto que temas como la gestión de recursos, han sentido ya la influencia de esta relativamente nueva tecnología en sus dominios. La facilidad de manejo de grandes volúmenes de información y su despliegue gráfico, hacen de los sistemas de información geográfico una herramienta vital para lograr el buen manejo de los recursos asignados a una institución de grandes proporciones.

Fue así como la Universidad Industrial de Santander, poco a poco fue sintiendo la necesidad de incorporar esta tecnología a su caja de herramientas, para poder así alcanzar nuevos estándares de gestión de recursos y de manera indirecta, ayudar a la eficiencia en el funcionamiento de esta entidad. Sin embargo, la Universidad también se enfrentó al grave problema que conlleva el uso de nuevas y mejores tecnologías: su precio. Fue imperativo entonces conseguir la reducción al máximo del costo de la implementación del S.I.G. (Sistema de Información Geográfica) sin sacrificar su veracidad y eficiencia. Evaluando todas estas variables, se llegó a la conclusión que era necesario desarrollar la aplicación S.I.G. con ayuda de los mismos estudiantes de la Universidad, validando ese trabajo como proyecto de grado, y con la utilización de software de dominio público.

El presente libro, ayuda a la comunidad en general a comprender el complejo proceso llevado a cabo para el desarrollo del S.I.G. que finalmente servirá

como principal soporte para la gestión de los recursos físicos, técnicos y humanos de la Universidad Industrial de Santander.

Este manuscrito puede dividirse claramente en tres partes: una parte correspondiente a la teoría, otra correspondiente a la práctica y la última correspondiente a los anexos. La primera parte, que comprende los capítulos 1, 2 y 3, trata primordialmente del fundamento teórico sobre el cual se basa el desarrollo de aplicaciones de tipo SIG. Es importante leer esta terna de capítulos para comprender la manera como debe comportarse y desarrollarse el sistema. La segunda parte, que comprende los capítulos 4, 5 y 6, explica brevemente la manera como se aplicaron los conceptos teóricos explicados en la primera parte, al caso específico a tomar en cuenta, así como algunas conclusiones extractadas de la experiencia del desarrollo de la aplicación. Por último, la tercera parte que comprende los Anexos del libro, muestra algunos resultados del trabajo realizado.

Esperamos que el presente libro exponga claramente cada una de las etapas de desarrollo del proyecto de grado titulado “**DISEÑO E IMPLEMENTACIÓN PARA UN PROTOTIPO DE SISTEMA DE INFORMACIÓN GEOGRÁFICO BAJO AMBIENTE WEB, PARA LOS RECURSOS FÍSICOS Y TÉCNICOS DE LA SECCIONAL SOCORRO DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER**” y que el trabajo iniciado sea enriquecido en el futuro cercano por algunos de los presentes lectores.

1. SISTEMAS DE INFORMACIÓN GEOGRÁFICA

1.1 DEFINICIÓN DE SIG

Un sistema de Información geográfica (SIG), es un híbrido entre los sistemas de diseño asistido por computadora (CAD) y los productos tradicionales de base de datos. El resultado es algo más que la suma de las partes: un SIG permite trabajar con datos en su verdadero contexto espacial, proporcionando de forma cómoda información en una interfaz con alto impacto visual y gran capacidad de síntesis informativa.

Es un sistema de hardware, software y procedimientos diseñados para soportar la captura, administración, manipulación, análisis, modelamiento y graficación de datos u objetos referenciados espacialmente, para resolver problemas complejos de planeación y administración. Una definición más sencilla es: Un sistema de computador capaz de mantener y usar datos con localizaciones exactas en una superficie terrestre.

Un sistema de información geográfica, es una herramienta de análisis de información. La información debe tener una referencia espacial y debe conservar una inteligencia propia sobre la topología y representación.

En general la tecnología SIG integra las operaciones comunes con bases de datos como consultas y análisis estadísticos con las posibilidades de visualización y análisis geográficos que ofrecen los mapas. Estas posibilidades distinguen a un SIG de otros sistemas de información.

1.2 COMPONENTES DE UN SIG

Todo SIG integra cuatro componentes claves: hardware, software, datos, y recurso humano.

1.2.1 Equipos (Hardware). Es donde opera el SIG. Hoy por hoy, programas de SIG se pueden ejecutar en un amplio rango de equipos, desde servidores hasta computadores personales.

1.2.2 Programas (Software). Los programas de SIG proveen las funciones y las herramientas necesarias para almacenar, analizar y desplegar la información geográfica. Los principales componentes de los programas son:

- Herramientas para la entrada y manipulación de la información geográfica.
- Un sistema de manejador de base de datos (DBMS).
- Herramientas que permitan búsquedas geográficas, análisis y visualización.
- Interfase gráfica para el usuario (GUI) para acceder fácilmente a las herramientas.

1.2.3 Datos. Los datos geográficos y tabulares pueden ser adquiridos por quien implementa el sistema de información, así como por terceros que ya los tienen disponibles. El sistema de información geográfico integra los datos espaciales con otros recursos de datos y puede incluso utilizar los manejadores de base de datos más comunes para manejar la información geográfica.

1.2.4 Recurso Humano. La tecnología de los SIG está limitada si no se cuenta con el personal que opera, desarrolla, administra el sistema y que establece planes para aplicarlo en problemas del mundo real.

1.3 MODELOS DE DATOS

Los Sistemas de Información Geográfica implementan tres modelos de datos: el modelo Vectorial, el Modelo **Raster** y el Modelo de Datos orientado a Objetos.

1.3.1 Modelo Vectorial. Para hacer la descripción de los objetos gráficos se utilizan vectores definidos por pares de coordenadas relativas a algún sistema cartográfico.

Por ejemplo, un punto se genera a partir de un par de coordenadas y su altitud, ahora, con dos puntos se generan una línea, y con una agrupación de líneas se forman los polígonos.

1.3.2 Modelo Raster. Basan su funcionalidad en una concepción implícita de las relaciones de vecindad entre los objetos geográficos. Su forma de proceder es dividir la zona de afección de la base de datos en una malla regular de pequeñas celdas (píxeles), y atribuir un valor numérico a cada celda como representación de su valor temático. Debido a que los píxeles son de tamaño constante y que conocemos la posición en coordenadas del centro de una de las celdas, se puede decir que todos los píxeles están georeferenciados.

1.3.3 Modelo orientado a objetos. Es un modelo basado en una colección de objetos, donde cada uno de ellos contiene valores almacenados en instancias dentro del objeto. Estos valores son objetos por sí mismos, esto es, los objetos contienen objetos a un nivel de anidamiento de profundidad arbitraria.

Los objetos que contienen los mismos tipos de valores y los mismos métodos se agrupan en clases.

El modelo orientado a objetos es más aconsejable para situaciones en las que la naturaleza de los objetos que tratamos de modelar es cambiante en el tiempo y/o espacio.

1.4 BASE DE DATOS GEOGRÁFICA

La esencia de un SIG está constituida por una base de datos geográfica. Esta es, una colección de datos acerca de objetos localizados en una determinada área de interés en la superficie de la tierra, organizados en una forma tal que puede servir eficientemente a una o varias aplicaciones. Una base de datos geográfica requiere de un conjunto de procedimientos que

permitan hacer un mantenimiento de ella tanto desde el punto de vista de su documentación como de su administración. La eficiencia está determinada por los diferentes tipos de datos almacenados en diferentes estructuras. El vínculo entre las diferentes estructuras se obtiene mediante el campo clave que contiene el número identificador de los elementos. Tal número identificador aparece tanto en los atributos gráficos como en los no gráficos. Los atributos no gráficos son guardados en tablas y manipulados por medio de un sistema manejador de bases de datos.

Los atributos gráficos son guardados en archivos y manejados por el software de un sistema SIG.

1.5 FUNCIONES BÁSICAS DE UN SIG

Dentro de las funciones básicas de un sistema de información podemos describir la captura de la información, esta se logra mediante procesos de digitalización, procesamiento de imágenes de satélite, fotografías, videos, procesos aerofotogramétricos, entre otros.

Otra función básica de procesamiento de un SIG hace referencia a la parte del análisis que se puede realizar con los datos gráficos y no gráficos, se puede especificar la función de contigüidad de objetos sobre una área determinada, del mismo modo se puede especificar la función de coincidencia que se refiere a la superposición de objetos dispuestos sobre un mapa.

1.6 APLICACIONES DE UN SIG

La utilidad principal de un Sistema de Información Geográfica radica en su capacidad para construir modelos o representaciones del mundo real a partir de las bases de datos digitales y para utilizar esos modelos en la simulación de los efectos que un proceso de la naturaleza o una acción antrópica produce sobre un determinado escenario en una época específica.

A continuación se presenta una lista de las diferentes posibles aplicaciones de los Sistemas de Información Geográfica:

- Administración de servicios públicos (acueducto, alcantarillado, energía, teléfonos, entre otros).
- Inventario y avalúo de predios.
- Atención de emergencias (incendios, terremotos, accidentes de tránsito, entre otros).
- Estratificación socioeconómica.
- Regulación del uso de la tierra.
- Control ambiental (saneamiento básico ambiental y mejoramiento de las condiciones ambientales, educación ambiental).
- Evaluación de áreas de riesgos (prevención y atención de desastres).
- Localización óptima de la infraestructura de equipamiento social (educación, salud, deporte y recreación).

- Diseño y mantenimiento de la red vial.
- Formulación y evaluación de planes de desarrollo social y económico.
- Por lo general se puede implementar en cualquier área o sector de interés.

2 BASES DE DATOS

2.1 DEFINICIÓN DE BASE DE DATOS

Las bases de datos son herramientas que almacenan la información haciendo que ésta sea más fácil de buscar y ser encontrada mediante cierto tipo de criterios. Es así como se visualiza el hecho de que siempre han existido, han sido útiles y lo mejor, han evolucionado.

Las bases de datos antiguas nos limitaban a buscar un solo dato a la vez y eso obviamente tomaba mucho tiempo y esfuerzo. Las bases de datos modernas tienen la capacidad de manejar índices que permiten acceder a la información más rápidamente y además encontrar múltiples resultados a la vez.

2.2 MODELO CONCEPTUAL

El modelo conceptual constituye la primera fase del desarrollo de la base de datos. Puede dividirse en dos etapas; la primera, es la etapa de percepción, identificación y descripción del mundo real a analizar, la segunda, es la etapa de la conceptualización, que es la transición del mundo real al esquema descriptivo, y de este último al esquema conceptual.

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.

2.3 MODELO LÓGICO

Para poder desarrollar este modelo es necesario tener elaborado el modelo conceptual del diseño, pues es a partir de este que se generan los mecanismos necesarios para pasar de un mundo real a un mundo formado por datos. En este proceso lo que se busca es identificar los elementos o conceptos empleados en el modelo conceptual y transformarlos en lo que denominamos entidades en el modelo Lógico.

El proceso de conversión se logra a partir de las siguientes formas:

- **Clasificación:** consiste en generar una única entidad que maneje conceptos con características comunes, de tal forma que todos ellos tendrán las mismas características y se diferencian unos de otros por los valores que toman dichas características.

- Agregación: consiste en separar cada una de las partes de un concepto para generar distintas entidades.
- Generalización: consiste en ir generando entidades de diferentes niveles, de tal forma que cada entidad de nivel superior agrupe las de nivel inferior.
- Asociación: consiste en la generalización de entidades a partir de entidades ya existentes.

2.4 MODELO ENTIDAD RELACIÓN

Es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

Originalmente, el modelo entidad-relación sólo incluía los conceptos de entidad, relación y atributo. Más tarde, se añadieron otros conceptos, como los atributos compuestos y las jerarquías de generalización, en lo que se ha denominado modelo entidad-relación extendido.

- Entidad: cualquier tipo de objeto o cosa sobre la que se recopila información.
- Relación: es una correspondencia o asociación entre dos o más entidades.
- Atributo: es una característica de interés o un hecho sobre una entidad o una relación.

- **Identificador:** es un atributo o conjunto de atributos que determinan de modo único cada ocurrencia de esa entidad.
- **Jerarquía de Generalización:** una entidad A es una generalización de un grupo de entidades A_1, A_2, \dots, A_n , si cada ocurrencia de cada una de esas entidades es también una ocurrencia de A . Todas las propiedades de la entidad genérica A son heredadas por las subentidades.

2.5 CLASIFICACIÓN DE LAS RELACIONES

2.5.1 Relaciones Uno a Uno (1:1). Cuando un registro de una tabla solo puede estar relacionado con un único registro de la otra tabla y viceversa.

2.5.2 Relaciones Uno a Muchos (1: M). Cuando un registro de una tabla solo puede estar relacionado con un único registro de la otra tabla, y un registro de la otra tabla puede tener más de un registro relacionado en la primera tabla.

2.5.3 Relaciones Mucho a Muchos (M: M). Cuando un registro de una tabla puede estar relacionado con más de un registro de la otra tabla y viceversa. Las relaciones muchos a muchos se suelen representar definiendo una tabla intermedia entre las dos tablas.

3 DESCRIPCIÓN DEL SOFTWARE EMPLEADO

3.1 HERRAMIENTAS DE SOFTWARE LIBRE DISPONIBLE

3.1.1 Servidores Web. Actualmente se pueden encontrar los siguientes servidores Web de manera gratuita en la red:

- **NCSA HTTPd.**
- Apache.
- **Roxen.**
- Otros.

Acerca de **NCSA HTTPd** se puede decir que posee las siguientes características:

1. *Bajo impacto:* Pequeño, para simplificar su seguridad y transporte. Rápido, para manejar la velocidad de carga en la **WWW**.
2. *Compatibilidad:* Diseñado para soportar **HTTP/1.0** además de soportar **HTTP/0.9**
3. *Índice de directorios:* El usuario puede crear índices **HTML** bajo una lista de nombres o el servidor puede generar un índice dinámicamente.

4. *Directorios de usuario*: Habilita fácilmente a los usuarios el acceso a la estructura de directorios, con suficiente seguridad para evitar que dañen el sistema.
5. **Scripts de Servidor CGI**: Posibilidad de configurar el servidor para ejecutar búsquedas y manejar formularios **HTML**.
6. *Soporte para Imagemap incluido*: Soporta archivos **Imagemap** usando formatos **NCSA** y **CERN** sin necesidad de una aplicación CGI externa.
7. *Seguridad*: Acceso limitado al servidor por directorios. Soporte para **HTTP/1.0 Basic Authentication**, **MD5 Digest Authentication** y **Kerberos** versiones 4 y 5. Soporta archivos de grupos y claves usando archivos planos **ASCII**, base de datos **DBM** o **NIS**. Es posible restringir acceso a una cierta **URL**.

Acerca de Apache se puede decir que es el servidor **Web** gratuito más utilizado hoy en día. Es el servidor que posee más grupos de noticias, más listas de distribución, más manuales, más fácil encontrar posibles problemas en la instalación, en el funcionamiento o en el desarrollo, se construyen más módulos que son puestos a disposición de la comunidad.

Apache fue creado como un Servidor **Web** de propósito general, capaz de servir pequeños **Web Sites**, unos pocos requests diarios, hasta grandes sitios **Web** de millones de requests por días. Sus principales características son:

1. Robusto

2. Rápido.
3. Modularizable.
4. Multiplataforma versiones para **Linux, Win32, MacOS, UNIX**, etc.
5. Está dentro del proyecto apache.

Acerca de **Roxen** se puede decir que es un servidor modular, esto quiere decir que las características del mismo se pueden ampliar con "módulos" externos programados por los propios creadores de la aplicación o por cualquier usuario para satisfacer una necesidad concreta.

Este programa realiza las funciones de Servidor **Web** de alto rendimiento con características en cuanto a rendimiento y estabilidad similares a otros servidores del mercado como Apache o **Microsoft IIS**. En lo referente a compatibilidad, soporta la mayoría de estándares del mercado: **Wap**, "**streaming multimedia**" mediante la configuración de los tipos **MIME** adecuados, etc. Además soporta tecnologías propias de otras plataformas como **SSI (Server Side Includes)**, **CGI**, etc. Además de todas estas tecnologías soportadas de otros programas servidores, aporta tecnologías propias como **scripts** en **Pike** y lenguaje **RXML (RoXen Macro Language)**.

3.1.2 Servidores de Bases de Datos. Actualmente se pueden encontrar los siguientes servidores de bases de datos de manera gratuita en la red.

- **PostgreSQL.**

- **MySQL.**
- **Interbase.**
- Otros.

Acerca de **PostgreSQL** se puede decir que es un Sistema de manejo de bases de datos entidad-relación (ORDBMS) basado en **POSTGRES**, versión 4.2, desarrollado en la Universidad de California en el Departamento de Ciencias Computacionales Berkeley.

PostgreSQL es un descendiente **open-source** (código abierto) del código original del proyecto **POSTGRES** de Berkeley. Provee soporte para el lenguaje **SQL92/SQL99** y otras características modernas. **POSTGRES** era pionero en los conceptos entidad-relación los cuales se hacen disponibles en la mayoría de las bases de datos comerciales. Los sistemas tradicionales de manejo de bases de datos (RDBMS) soportan un modelo de datos que consiste en una colección de relaciones nombradas, con atributos de cierto tipo. En los sistemas comerciales actuales, algunos tipos incluyen números reales, enteros, cadenas de caracteres, dinero y fechas. Se reconoce comúnmente que este modelo es inadecuado para las aplicaciones de procesamiento de datos futuras.

El modelo relacional reemplazó con éxito los modelos predecesores debido a su "Simplicidad Espartana". Sin embargo, esta simplicidad hace la implementación de ciertas aplicaciones difícil. **PostgreSQL** incorpora los siguientes conceptos adicionales, que permiten al usuario extender el sistema fácilmente:

- Herencia.
- Tipos de datos.
- Funciones.

- **Constraints.**
- **Triggers.**
- Reglas.
- Integridad de transacciones.

Todas estas características hacen que **PostgreSQL** esté incluida dentro de las bases de datos que hacen referencia al modelo Entidad – Relación. Es de anotar que esto significa una referencia diferente al modelo orientado a objetos, el cual en general no es propio para soportar lenguajes de bases de datos de relaciones tradicionales. Así que a pesar de que **PostgreSQL** posee algunas características orientadas a objetos, se encuentra firmemente en el mundo de las bases de datos relacionales.

Acerca de **MySQL** se puede decir que es un servidor de bases de datos SQL (**Structured Query Language**), el cual es el lenguaje de bases de datos más popular en el mundo. **MySQL** es una implementación cliente servidor que consiste en un **daemon mysqld** y muchos más clientes librerías/programas.

Los principales objetivos de **MySQL** son obtener la mayor velocidad y ser lo más robusto posible. Se basa en un juego de rutinas que han sido utilizadas en un ambiente productivo de altísima demanda por muchos años. Aún cuando **MySQL** se encuentra en proceso de desarrollo, ofrece un juego de funciones altamente útiles.

Las principales características de **MySQL** son:

1. **Multi-threaded:** (multihilo)

2. Soporte múltiple de lenguajes: Soporta los lenguajes **C, C++, JAVA, Perl, Python** y **APIs TCL**.
3. Soporte completo a **SELECT, WHERE, GROUP BY, ORDER BY**.
4. Funciones SQL implementadas a través de una librería de clases optimizada.
5. Soporte para funciones de grupo sum, max y min.
6. Un sistema de privilegios y contraseñas muy flexible y seguro.
7. Permite verificación a través de huéspedes.
8. Tráfico de contraseñas encriptado.
9. Soporta ODBC **Open-DataBase-Connectivity** para **Windows 95/98**, todas las funciones ODBC 2.5 y muchas otras.
10. Puede manejar bases de datos grandes.
11. Todas las columnas tienen valores por defecto.
12. Escrito en **C, C++** y probado con **gcc**.
13. Contiene una utilidad muy rápida para chequeo y reparación de tablas.
14. Todos los comandos tienen ayuda.
15. Soporta mensajes de error a clientes en muchos idiomas.

Acerca de **InterBase** podemos decir que es una base de datos relacional de alto rendimiento, multiplataforma basada en estándares SQL que combina facilidad de uso, bajo costo de mantenimiento y poder de clase empresarial. **InterBase** 6.5 combina características robustas y arquitectura comprada con tecnología que habilita a los desarrolladores a crear aplicaciones poderosas. **InterBase** soporta la mayoría de clientes populares y entornos de desarrollo de aplicaciones incluyendo **Delphi™**, **Kylix™**, **C++ Builder™**, **JBuilder™**, **Microsoft Access**, y clientes JDBC y ODBC.

Las principales características de **Interbase** son:

1. Meta datos Ampliados: Una característica de seguridad incluida en la nueva versión, protege los meta datos de cualquier modificación por parte de usuarios no autorizados.
2. Soporte a VLDB (**Very Large Database: 64 bit**): Permite la creación de archivos grandes sin necesidad de crear una base de datos multi-archivo.
3. Filas SQL: Utilizadas en conjunto con la sentencia "**SQL ORDER BY**" provee la generación de rangos para series de SQL, permitiendo a los desarrolladores Web distribuir las piezas de información desde el servidor Web hacia el navegador cliente.
4. Sentencia de Cancelación Asíncrona: Le da al usuario la posibilidad de cancelar la ejecución de una sentencia DSQL mediante la interfaz de **InterBase**.
5. Generación de Datos XML: Permite a los programadores de **InterBase** generar documentos XML directamente desde **InterBase**.

3.1.3 Servidores de mapas. Actualmente se pueden encontrar los siguientes servidores de mapas de manera gratuita en la red:

- **MapServer.**
- **Mapmaker.**
- Otros.

Acerca de **Mapserver** se puede decir que es un servidor de mapas de código abierto, que corre sin dificultades en plataformas **Linux**, **UNIX** y **Windows NT/98/95**.

El sistema soporta **MapScript**, lo que permite utilizar lenguajes bastante populares como **Perl**, **Python**, **Tk/Tcl**, **Guile** e incluso Java para acceder el API de **MapServer**. Es posible entonces integrar datos de casi cualquier base de datos (**Oracle**, **Sybase**, **MySQL**, etc) con datos de SIG's tradicionales en un simple mapa o página **web**. Además, está disponible ahora un módulo **PHP/MapScript**.

Las principales características de **Mapserver** son:

1. Formatos vectoriales soportados: Archivos **shape ESRI**, **ESRI ArcSDE** (versión alfa).
2. Formatos **raster** soportados (sólo de 8 bits): **TIFF/GeoTIFF**, **GIF**, **PNG**, **ERDAS**, **JPEG** y **EPPL7**.
3. Fuente **True Type** soportada.
4. Leyendas y escalas automáticas.

5. Dibujo dependiente de escala.
6. Construcción de mapas temáticos a partir de clases basadas en expresiones lógicas o regulares.
7. Etiquetado de objetos.

MapServer no es un SIG completo, ni aspira serlo. Sin embargo, si provee la funcionalidad básica para soportar una variedad bastante amplia de aplicaciones de red. Mas allá de explorar datos SIG, **MapServer** permite crear “mapas geográficos de imagen”, eso es, mapas que pueden dirigir a los usuarios a contenidos.

MapServer fue desarrollado inicialmente por el proyecto **ForNet** de la Universidad de Minnesota (UNM) en colaboración con la NASA y el Departamento de Recursos Naturales de Minnesota (MNDNR). Las mejoras adicionales fueron hechas por el MNDNR y el Centro de Información de Manejo de Tierra de Minnesota (LMIC). El desarrollo actual se encuentra a cargo del proyecto TerraSIP, un proyecto patrocinado por la NASA.

Acerca de **Mapmaker** se puede decir que es uno de varios sistemas de información geográfica o de mapeo disponibles para usar con **Microsoft Windows**. Sus principales características son:

1. *Compatibilidad*: los datos pueden ser importados de los programas SIG y **CAD** más comúnmente utilizados. Importa datos vectoriales de

archivos **SHP**, **DXF** y **MIF**, así como texto ASCII y archivos **raster BMP**, **TIF**, **JPG** y **ECW**.

2. *Configuración como visor*: puede ser configurado como un visor de sólo lectura o como un programa de creación de mapas bastante sencillo apropiado para niños.
3. *Gratis*: no sólo es gratuito su uso, sino que además los mapas realizados pueden ser distribuidos sin costo.
4. *Mapas de buen aspecto*: número ilimitado de capas vectoriales y **raster**. Fondos de mapas de Bits en color y escala de grises. Variados estilos de símbolos, rellenos y líneas. Rellenos traslúcidos y semi traslúcidos. Variedades de texto, flechas y dimensiones. Fácil creación de escalas gráficas, norte, leyendas, paneles de texto, imágenes embebidas, etc.
5. *Creación y edición de mapas*: creación de mapas desde cero o importados y editados. Edición y dibujo vectorial. Unión y corte de polígonos intuitiva. Creación de polígonos con islas. Edición simple de límites comunes. Creación de zonas **buffer**. Calibración de imágenes **raster**. Usar datos de archivos DBF para colorear mapas o determinar símbolos. Medición simple de áreas y distancias. Edición de juegos de datos.

3.1.4 Lenguajes de programación para páginas Web. Actualmente se pueden encontrar los siguientes lenguajes de programación para páginas Web de manera gratuita en la red:

- **PHP.**
- **ASP.**
- **Java.**
- **Otros.**

Acerca de **PHP** se puede decir que es un lenguaje sencillo, de sintaxis cómoda y similar a la de otros lenguajes como C o C++, es rápido y dispone de una gran cantidad de librerías que facilitan muchísimo el desarrollo de las aplicaciones.

El código generado es mantenible, se lee muy bien y podemos programar utilizando objetos. **PHP** es un lenguaje ideal tanto para el que comienza a desarrollar aplicaciones **Web** como para el desarrollador experimentado, y está alcanzando unos niveles de uso (más de 1 millón de **Web's** lo utilizan ya) que convierten su conocimiento en algo indispensable para los profesionales del desarrollo en Internet.

Y, además, **PHP** es un lenguaje basado en herramientas con licencia de software libre, es decir, no hay que pagar ni licencias, ni estamos limitados en su distribución y, podemos ampliarlo con nuevas funcionalidades si así lo quisiéramos. Respecto a su licencia, en la versión **PHP** 3.0 era **GPL**, pasando a ser modificada en su versión 4.0, por la incorporación de **Zend**, un nuevo intérprete de **PHP** mucho más rápido que el anterior de **PHP**. Este intérprete **Zend** tiene asociado un optimizador opcional, el cual ya está más dentro del mundo comercial que del software libre, tema que ha levantado

algunas suspicacias dentro de la comunidad de desarrolladores de software libre.

PHP es un lenguaje de programación que está muy orientado al desarrollo de aplicaciones **Web**. Cuando pedimos a nuestro servidor de **Web** una página **PHP**, que no es más que un programa **PHP** que genera **HTML**, antes de enviar dicha página al cliente se la pasa al intérprete de **PHP**. Este la interpreta y es el resultado de esta interpretación del programa **PHP**, contenido en la página **PHP**, lo que termina llegando al cliente.

Siempre que se habla de **PHP** lo primero que se hace es presentar el gran número de gestores de bases de datos a los que puede acceder: **Adabas D**, **dbm**, **dBase**, **filePro**, **Hyperwave**, **Informix**, **InterBase**, **LDAP**, **Microsoft SQL server**, **mSQL**, **MySQL**, **ODBC**, **Oracle**, **PostgreSQL**, **Solid**, **Sybase**, etc.

Pero si este aspecto resulta impresionante, no menos el soporte para: acceso a servidores **IMAP**, envío de correo con **SMTP**, acceso a servidores de **FTP**, acceso a **SNMP** para gestión de redes y equipos, generación dinámica de gráficos y documentos **PDF**, análisis de documentos **XML**, corrector de ortografía y generación de datos en WDDX (Intercambio Web de Datos Distribuidos).

Y todo esto ha sido posible gracias a que **PHP** ha sido diseñado de forma muy modular y ha sido sencillo según han ido surgiendo librerías, utilizarlas desde **PHP**. Toda esta funcionalidad está basada en librerías que en su mayor parte no han sido desarrolladas por el equipo de **PHP**. La gente de **PHP** lo que ha implementado han sido los cabos necesarios para poder acceder a las librerías.

Por último, **PHP (Hypertext Preprocessor)** es un lenguaje interpretado de alto nivel que se integra a páginas **HTML** y se ejecuta en el servidor. Es un típico caso de un **Server Side Scripts** (lenguaje de **scripts** ejecutados en el servidor) y tiene muchas características, entre las que sobresalen las siguientes:

1. Es de uso libre, así que no hay que pagar licencias.
2. Funciona en varias plataformas, entre ellas: **Linux**, **Solaris** y **Windows**.
3. Tiene capacidad de conectarse a las principales bases de datos relacionales, como **Oracle**, **IBM DB2**, **MS SQL Server**, **InterBase**, **Sybase**, y muchas más.
4. Tiene un conjunto de comandos muy completo, que permite crear complejos programas, incluso programación orientada a objetos.

Como se ve, es una muy buena herramienta y sin costo, además la independencia de plataforma permite escoger la que mejor se adapte a las necesidades del sistema.

Acerca de ASP se puede decir que **Active Server Pages (ASP)** es una nueva tecnología creada por **Microsoft**, destinada a la creación de sitios **web**. No se trata de un lenguaje de programación en sí mismo (ya que los **ASP** se pueden programar en **VBScript**, **JavaScript**, **PerlScript** o en varios

otros lenguajes), sino de un marco sobre el que construir aplicaciones basadas en Internet.

La tecnología **ASP** apareció por primera vez (versión 1.0) con el servidor **Internet Information Server** 3.0 de **Microsoft** en Diciembre de 1996. La versión 4.0 de IIS (el **Option Pack** para NT 4.0) incluye la versión 2.0 de **ASP**, y la versión 5.0 de IIS, distribuida con **Windows** 2000, incluye **ASP** 3.0.

Los predecesores de **ASP** incluyen CGI y **Perl**. Las tecnologías de **Microsoft** predecesoras de **ASP** incluyen IDC y **WebDB**. Otras tecnologías que compiten con **ASP** son **ColdFusion** (**Allaire**), **JavaServer Pages** (**Sun Microsystems**) y **PHP** (de libre distribución bajo **Open System**).

Algunas de las características de **ASP** son:

1. **ASP** es totalmente gratuito para **Microsoft Windows NT** o **Windows** 95/98.
2. El código **ASP** se puede mezclar con el código **HTML** en la misma página (no es necesario compilarlo por separado).
3. El código **ASP** se puede escribir con un simple editor de textos como el Bloc de notas de **Windows** o **UltraEdit**.
4. Cómo el código **ASP** se ejecuta en el servidor, y produce como salida código **HTML** puro, su resultado es entendible por todos los navegadores existentes.

5. Mediante **ASP** se pueden manipular bases de datos (consultas, actualizaciones, borrados, etc.) de prácticamente cualquier plataforma, con tal de que proporcione un **driver** OLEDB u ODBC.
6. **ASP** permite usar componentes escritos en otros lenguajes (C++, **Visual Basic**, **Delphi**), que se pueden llamar desde los guiones **ASP**.
7. Sin modificar la instalación, los guiones **ASP** se pueden programar en **JScript** o **VBScript** (este último es el más usado porque más programadores lo dominan), pero también existen otros lenguajes, como **Perlscript** y **Rexx**, que se pueden emplear para programar **ASP**.
8. Se ha portado a la plataforma Java por **ChiliSoft** y **Halcyon Software**, lo que permite que **ASP** sea usado en casi cualquier sistema operativo.

Las principales ventajas que ofrece **ASP** son:

1. Permite acceder a bases de datos de una forma sencilla y rápida.
2. Las páginas se generan dinámicamente mediante el código de **scripts**, (guiones).
3. El código de **script** se ejecuta en el servidor, y no se depende del navegador que se emplee.
4. Desde una página **ASP** se pueden ejecutar servidores **OLE** en el servidor de **web**, lo que abre un abanico de nuevas posibilidades sólo

accesibles previamente usando CGI y filtros ISAPI: acceso a base de datos, acceso a ficheros, **logging** en el sistema, envío de correo, etc.

5. La tecnología **ASP** se emplea principalmente para crear aplicaciones interactivas que funcionan en Internet.

Acerca de **Java** se puede decir que no requiere un tiempo de compilación, los **scripts** se pueden desarrollar en un periodo de tiempo relativamente corto. A esto podemos añadirle las características de interfaz como, por ejemplo, cuadro de diálogo, formularios y otros elementos GUI (Interfaz Gráfico de Usuario), son gestionados por el navegador y por el código **HTML**. Por lo tanto los programadores que utilizan **JavaScript** no se deben preocupar en crear o controlar dichos elementos en sus aplicaciones.

Como **WWW** es independiente de la plataforma hardware o sistema operativo, los programas escritos en **JavaScript** también lo son, siempre y cuando exista un navegador con soporte **JavaScript** para la plataforma en cuestión.

Los programas **JavaScript** tienden a ser pequeños y compactos (en comparación con los **applets** de Java), no requieren mucha memoria ni tiempo adicional de transmisión. Además, al incluirse dentro de las mismas páginas **HTML** se reduce el número de accesos independientes a la red.

3.2 LINUX RED HAT 9.0

Esta aplicación surgió cuando un grupo de programadores en Carolina del Norte (USA) decidió hacer más fácil la experiencia de trabajar con Linux. Su objetivo era empaquetar todos los bytes necesarios en una distribución coherente que permitiera de manera fácil a los usuarios más inexpertos interactuar con el nuevo sistema operativo.

Esta distribución fue concebida bajo el concepto de paquetes, donde cada uno de estos incluye un fragmento diferente de software, configurado, completamente probado y listo para ser ejecutado. Este concepto de paquetes permite que la distribución del Linux pueda ser actualizada fácilmente.

Desde la versión 4.0, **Red Hat Linux** puede ser ejecutado en tres plataformas líderes de mercado: Intel y PCs compatibles, ordenadores **Alpha Digital** y equipos SPARC. Un árbol de fuentes unificado y los beneficios del RPM (Administrador de Paquetes **Red Hat**) permiten distribuir el **Red Hat Linux** a cada plataforma con un mínimo de esfuerzo. Esto permite a los usuarios portar sus softwares entre estas plataformas de manera mucho más simple.

Para el desarrollo del prototipo se plantea el uso de la última versión disponible, **Linux Red Hat 9.0**, de la cual se presentan sus características y ventajas a continuación:

- *Interfaz Mejorado*: presenta una eficiente organización de los menús para un acceso más fácil a las aplicaciones y herramientas más usadas.

- *Aplicaciones y Herramientas integradas:* entre sus principales aplicaciones se pueden encontrar incluidas entre otras, la suite de aplicaciones de Ofimática **Open Source** conocida como **OpenOffice**, la cual permite la creación de documentos, hojas de cálculo y presentaciones. También se encuentra Mozilla el cual permite una excelente navegación en el **Web**, junto con un servicio de cliente de correo electrónico, agenda de direcciones y autor **HTML**. Además entre sus herramientas más importantes se destacan las que permiten configurar mediante interfaz grafica los diferentes dispositivos periféricos, así como el sonido, video y resolución de visualización.
- *Componentes de Núcleo actualizado:* los principales componentes incluidos en **Red Hat Linux 9.0** son el **Kernel 2.4.20**, el **Servidor Web Apache httpd 2.0**, la versión 7.3.2 de **PostgreSQL**, entre otros.

3.3 APACHE VERSIÓN 2.0

Apache es el servidor **Web** más utilizado en Internet, con gran diferencia sobre el resto, según los datos proporcionados por **Netcraft** (Organismo que realiza estadísticas a través del Internet acerca del uso de Software).

Puede ser utilizado, sin ningún problema, incluso con **Windows 9x**, o **ME**, aunque no se recomienda su utilización en sistemas de producción, por los conocidos problemas de utilización de estos sistemas operativos.

Se configura de forma sencilla, ajustando unos parámetros en el archivo de configuración, con cualquier editor de texto.

A continuación se presentan las ventajas del uso de Apache:

- Es seguro y confiable.
- Tiene un excelente rendimiento.
- Se integra muy bien con lenguajes como **PERL**, **PYTHON** y **PHP**.
- Utilizable en **WINDOWS**, **LINUX**, **UNIX**, etc.
- Es gratis.

Las anteriores son las principales razones por las cuales se considera que este servidor **Web** es el que más está acorde a las necesidades del proyecto.

3.4 PHP VERSIÓN 4.2.2

Aunque no es el lenguaje más utilizado en el mundo en la actualidad (cuarto puesto en popularidad), se espera que alcance el primer lugar dentro de un par de años, pues presenta una serie de ventajas altamente atractivas para el programador novato y experimentado.

Entre estas se encuentran:

- Alto rendimiento.

- Interfaces a diferentes sistemas de bases de datos, resaltando que es de fácil integración con **Postgresql**.
- Librerías para realizar muchas tareas comunes en el desarrollo **Web**.
- Es gratis.
- De fácil aprendizaje y uso.
- Disponibilidad del código fuente.

Son por éstas y otras razones que se decidió emplear **PHP** para el desarrollo del prototipo, recalando una vez más que la mayor razón es aquella que tiene que ver con la fácil integración a **Postgresql**.

3.5 POSTGRESQL VERSIÓN 7.3.2

Postgresql ofrece muchas ventajas respecto a otros sistemas de Bases de Datos, entre ellas cabe resaltar:

- Instalación ilimitada, ya que no existe costo asociado a la licencia del software.
- Amplio soporte.

- Ahorros considerables en costos de operación.
- Estabilidad y confiabilidad.
- Multiplataforma: está disponible en casi cualquier **Unix**, también existe una versión para **Windows** usando la plataforma **Cygwin**.
- Posee herramientas gráficas de diseño y administración de bases de datos como **pgAdmin** y **pgAcces**.

Entre sus características técnicas más importantes se encuentran:

- Cumple con **ANSI SQL**.
- Interfaces nativas para ODBC, JDBC, C, C++, PHP, **Perl**, TCL, ECPG, **Python** y Ruby.
- **Triggers**
- **Unicode**
- Una **API** abierta.
- Herramientas para generar **SQL** portable para compartir con otros sistemas compatibles con **SQL**.

3.6 POSTGIS VERSION 0.7.5

Fue creado por **Refractions Research Inc.** Como un proyecto de investigación de Tecnologías de Bases de Datos Espaciales. Está publicado bajo licencia GNU.

Postgis no es más que una extensión para **Postgresql** que permite el almacenamiento de datos en la base de Objetos Geográficos, esto lo hace definiendo nuevos tipos de datos, crea dos tablas con información relevante al sistema (proyección de los datos y columna que posee la información geográfica) y define también las funciones de manejo de información como procedimientos almacenados.

Además trae interfaces para servir de fuente de datos a **MapServer**.

3.7 MAPSERVER VERSIÓN 3.6.6

Es una aplicación desarrollada para trabajar bajo un ambiente de Internet, la cual corre bajo plataformas **Linux/Apache**, **Windows NT/98/95**, dando amplias posibilidades de trabajo.

Constantemente se están ofreciendo nuevas versiones y está en constante desarrollo.

Para la generación de una interfaz, se puede utilizar programación **HTML**, y **JavaScript**. Se permite el despliegue de las capas geográficas, clasificación de la leyenda, simbología y componentes adicionales como colores, ancho de línea etc., cada uno de los componentes se debe definir en el archivo .map.

Según la versión seleccionada, el despliegue de los datos puede hacerse utilizando el protocolo **OGC** de servicios de mapas que permitirá la conexión fácilmente con otro servidor.

3.8 PHP/MAPSCRIPT VERSIÓN MAPSERVER 3.6.6

Es un módulo **PHP** dinámicamente capaz de cargarse, que hace que las funciones y clases **Mapscript** de **Map Server** estén disponibles en un ambiente **PHP**.

Este módulo fue desarrollado inicialmente por **PHP-3.0.14** y trabaja con las versiones mas recientes de **PHP3**. También ha sido desarrollado para ser soportado por **PHP4**, estando disponible a partir de la versión 3.4 de **MapServer**.

PHP Mapscript ha alcanzado un punto estable, pero sin embargo se encuentra bajo constante desarrollo.

Se comenzó a incluir en la versión 3.3.011 de **MapServer**, hasta versiones más recientes del mismo, incluida la actual

4 METODOLOGÍA

4.1 RECOLECCIÓN Y PROCESAMIENTO DE LA INFORMACIÓN

La información empleada proviene en algunos casos de la oficina de planeación de la sede de la UIS en Bucaramanga; sin embargo, dicha información no era del todo válida, debido a la antigüedad de los edificios (esto implica remodelaciones y cambios en los mismos) de la sede UIS Socorro; es por esto que gran parte de ésta información fue necesaria volverla a levantar, además de la información inexistente.

Lo que se realizó fue un levantamiento de todos aquellos elementos que eran de interés y cada uno de ellos con sus características principales. El procedimiento general para un recinto fue el siguiente:

- Medir los diferentes lados del polígono (recinto) y realizar un gráfico representativo del mismo. Una vez hecho esto, se le asignaba el correspondiente código establecido por la Institución.
- Después se identificaron cada uno de los elementos interiores del recinto, tales como interruptores, luminarias, tomacorrientes, puertas, columnas, aparatos sanitarios, elementos de la red estructurada, etc., a los cuales se les reconocieron sus características más importantes, como dimensiones, tipos de elementos, cantidad de elementos, etc.

En fin, cada uno de los diferentes sitios de la Institución fue levantado de forma cuidadosa y detallada hasta obtener toda la información necesaria.

Una vez obtenida toda la información, lo que se hizo fue organizarla y clasificarla para así hacer más fácil el proceso de digitalización de la misma.

El proceso de digitalización se realizó con el software **Autocad Map 2000** y **Excel XP** tal y como se tenía previsto desde un principio.

Ya digitalizada la información, se separó la misma en diferentes bloques de acuerdo a su tema, para luego ser convertida en topologías. Estos bloques mencionados manejan por separado los puntos, las líneas y los polígonos, pues se debe hacer así, para poder crear las topologías.

4.2 DISEÑO DE LA BASE DE DATOS

En el Anexo B se puede observar todo el resultado del proceso de diseño de la Base de Datos.

A continuación se hace una breve explicación del procedimiento que se debe tener en cuenta para el diseño de una Base de Datos.

4.2.1 Modelo Conceptual. Inicialmente es necesario establecer los diferentes tipos de usuarios que van a emplear el sistema. Dichos usuarios se establecieron de acuerdo al tipo de información y a los requerimientos establecidos. (Ver Anexo B).

En esta etapa se construyó un esquema de la información empleada, en el cual se plantearon y establecieron las entidades, atributos y relaciones necesarias para el buen funcionamiento del sistema.

Este esquema es una fuente de información para el diseño lógico de la base de datos.

4.2.2 Modelo Lógico. Esta es la etapa en la que se construyó el esquema de la información empleada, basándose en uno de los modelos de Bases de Datos específicos, independientemente del Sistema de Gestión de Base de Datos a utilizar y de cualquier otra consideración física.

En esta etapa se transformó el modelo Conceptual en el modelo Lógico, bajo la notación del Modelo Entidad Relación, pues se consideró que este era el más apropiado para el sistema. A medida que se fue desarrollando el Modelo Lógico, este se fue probando y validando con los requisitos.

El esquema lógico es una fuente de información para el diseño físico. Además, juega un papel importante durante la etapa de mantenimiento del sistema, ya que permite que los futuros cambios que se realicen sobre los programas de aplicación o sobre los datos, se representen correctamente en la base de datos.

El diseño del Modelo Conceptual y del Modelo Lógico, son procesos iterativos que van mejorando consecutivamente, es decir, evolucionan de tal forma que llega el momento en que el diseñador encuentra los modelos conceptuales y lógicos definitivos.

En el Anexo B se encuentran las descripciones de las Entidades, la clasificación y codificación de los elementos.

4.2.3 Diccionario de Datos. En esta etapa del proyecto lo que se hizo fue definir cada uno de los diferentes objetos de acuerdo a su naturaleza; además, cada uno de estos fue clasificado dentro del tipo de objeto que podía ser, ya fuera punto, línea o polígono.

La descripción detallada se encuentra en el Anexo B.

4.2.4 Modelo Físico. Esta etapa es la implementación de los modelos mencionados en el software empleado para el desarrollo de la aplicación SIG.

4.3 DEFINICIÓN DE TOPOLOGÍAS

Una de las principales características de los SIG es la capacidad que tienen para distinguir las relaciones existentes entre entidades geográficas.

Para automatizar la representación de los objetos del mundo real en un SIG, se debe especificar explícitamente que información se va a almacenar, que estructura informática se va a utilizar y que uso se le dará a los datos. Un SIG provee un modelo de datos que permite representar información contenida en mapas y para el marco de relaciones espaciales se hace uso de datos espaciales. Dentro de un SIG se pueden manipular los objetos que existen dentro de una connotación espacial. Estos objetos poseen información referente a su tamaño y ubicación. Su situación con los demás objetos describen sus relaciones topológicas.

La relación entre todos los objetos espaciales (Topología) se entiende como la capacidad de los SIG de relacionar diferentes elementos en el espacio, y esto se alcanza mediante el concepto de topología, el cual tiene que ver con las construcciones de los atributos de relación entre los puntos, líneas y polígonos.

4.3.1 Tipos de Topologías. Existen tres tipos de topologías: la topología de nodos, la topología de redes y la topología de polígonos.

- **Topología de Nodos.** Es un sistema de objetos el cual describe como una serie de puntos independientes se relacionan entre si. Dichos puntos independientes son conocidos como nodos, los cuales pueden ser representados de forma simple como objeto punto y a su vez como objetos con punto de inserción. Los nodos tiene solo localización (x, y) y no ocupan área.

Cuando se crea una topología de nodos, se deben especificar cuales nodos deben ser incluidos en la topología, dar un nombre a la topología y agregar una descripción. Así mismo, al ser creada la topología se asigna un número como identificador único para cada punto seleccionado que es almacenado en una tabla de datos de objetos. Esta información puede ser usada para hacer consultas o análisis.

A continuación se relacionan los diferentes elementos que se consideraron y se crearon como topologías de nodos:

Puntos Eléctricos

Luminarias

Tomacorrientes

Tableros

Medidores

Interruptores

Puntos de Voz y Datos

Subestación Eléctrica

Puntos Hidráulicos

Grifos

Válvulas

Duchas

Tanques

Medidores

Lavaplatos

Lavamanos

Sanitarios

Lavaderos

Puntos Sanitarios

Sumideros

Pozos de Inspección

Cajas de Inspección

Sifones

Sanitarios

Lavaplatos

Lavamanos

Lavaderos

Puntos Especiales

Aire Acondicionado

Puntos proveedores de Oxígeno

Puntos proveedores de Gas Natural

Puntos proveedores de CO₂

- **Topología de redes.** Es un sistema de vínculos (segmentos) conectados usado para desarrollar análisis en entidades topográficas lineales. Se les puede asignar un valor de dirección a los segmentos en una red topológica. Los vínculos son objetos que conectan dos puntos en el mapa. Cada uno de estos vínculos tiene un punto de inicio y un punto final. Los vínculos o segmentos pueden representar cualquier entidad lineal del mapa.

En una topología de redes se describen las relaciones que existen entre los segmentos conectados unos con otros.

A continuación se relacionan una serie de elementos que fueron considerados y creados bajo el criterio de Topología de Redes: Redes Eléctricas, Redes Hidráulicas, Redes Sanitarias, entre otras.

- **Topología de polígonos.** La topología de polígonos incluye áreas o polígonos, los cuales se forman de la interconexión de vínculos (Segmentos). Los segmentos forman los límites de los polígonos en una topología de polígonos. Un segmento puede servir como límite entre dos o más polígonos.

Las topologías de polígonos pueden ser usadas para analizar relaciones entre superficies de tierra.

En una topología de polígonos es de total y absoluta importancia asegurarse que los objetos estén cerrados.

Tal y como se puede observar en el diccionario de datos presentado en la parte final de este documento (**ANEXO B**), la mayoría de los elementos fueron considerados y creados bajo el criterio de Topologías de polígonos pues la gran mayoría son elementos de área. Dichos elementos fueron clasificados en diferentes grupos de acuerdo a sus características principales. Entre los principales se pueden nombrar: Muros, columnas, pisos, edificios, etc.

4.3.2 Creación de Topologías.

Antes de poder crear una topología, se debe estar seguro que la información del dibujo no contiene errores en la geometría. Por esto es esencial hacer uso de las herramientas de limpieza brindadas por Autocad Map 2000 para depurar la información que se pretende emplear en topologías.

A continuación se presenta un listado de los posibles problemas que evitan la creación de topologías de redes o de polígonos:

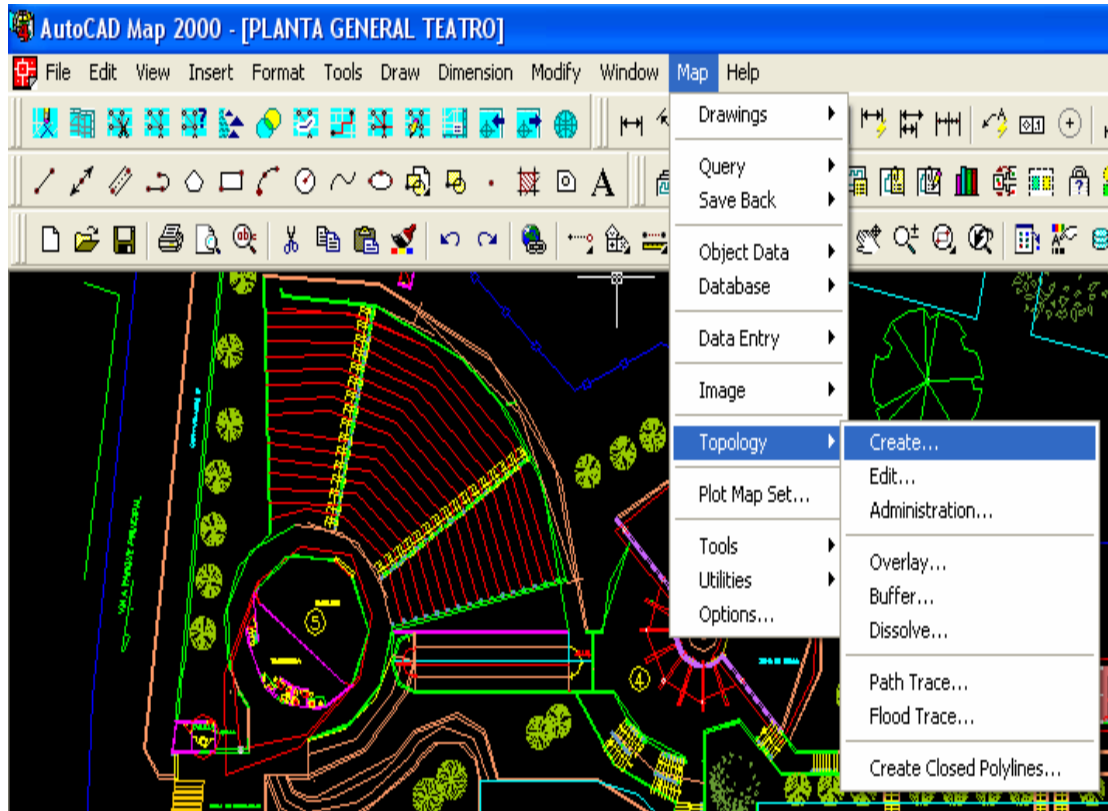
- **Objetos Insuficientes:** son los que se encuentran dentro de una tolerancia con respecto a otro.
- **Objetos sueltos:** son los que tienen al menos un punto final que no es compartido con otro objeto.
- **Objetos cortados:** son los que se cruzan unos con otros sin un nodo de intersección.
- **Objetos duplicados:** son los que comparten el mismo punto de inicio y el mismo punto final.
- **Objetos cortos:** son los que son más cortos en longitud que la tolerancia especificada.
- **Nodos agrupados:** son los nodos que se encuentra dentro de una tolerancia uno del otro.

Tan pronto se haya finalizado la limpieza, y toda la información del dibujo esté lista, se puede proceder a la creación de la topología.

A continuación se presenta la manera práctica en que se crea la topología:

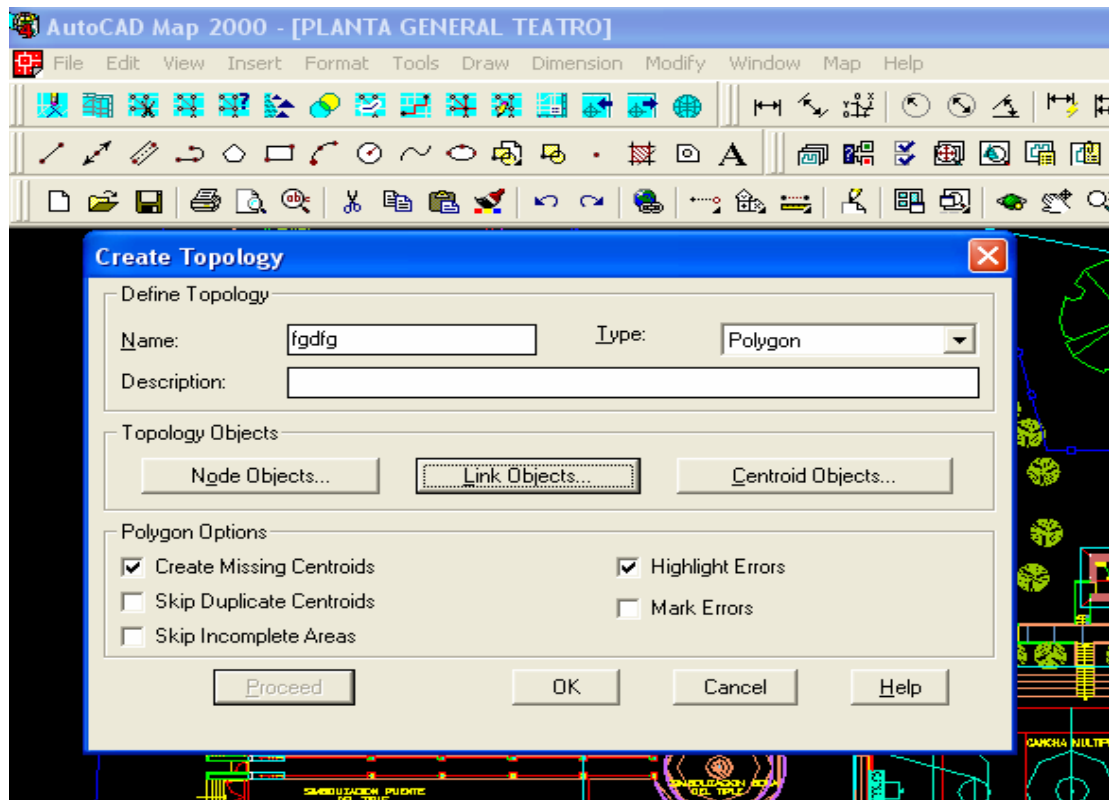
- Lo primero que se debe hacer es ir al menú **map**, buscar la opción **topology** y hacer clic en la opción **create** (ver figura 1).

Figura 1. Creación de Topología: Menú Map



- Una vez hecho esto se despliega una caja de dialogo (ver figura 2), en la cual se debe especificar un nombre único para la topología y elegir el tipo de topología a crear. A su vez se deberán seleccionar los objetos a incluir en la topología a través del botón **Link Objects**, en donde se da la orden selección automática.

Figura 2. Creación de topología: Caja de diálogo



- Finalmente se da la orden **OK** para terminar el proceso de creación de la topología.
- Una vez están creadas todas las topologías lo que se debe realizar es la conversión de los archivos para que estos puedan ser leídos e interpretados por el SIG. Esto se logra generando los archivos **shape** (.shp)

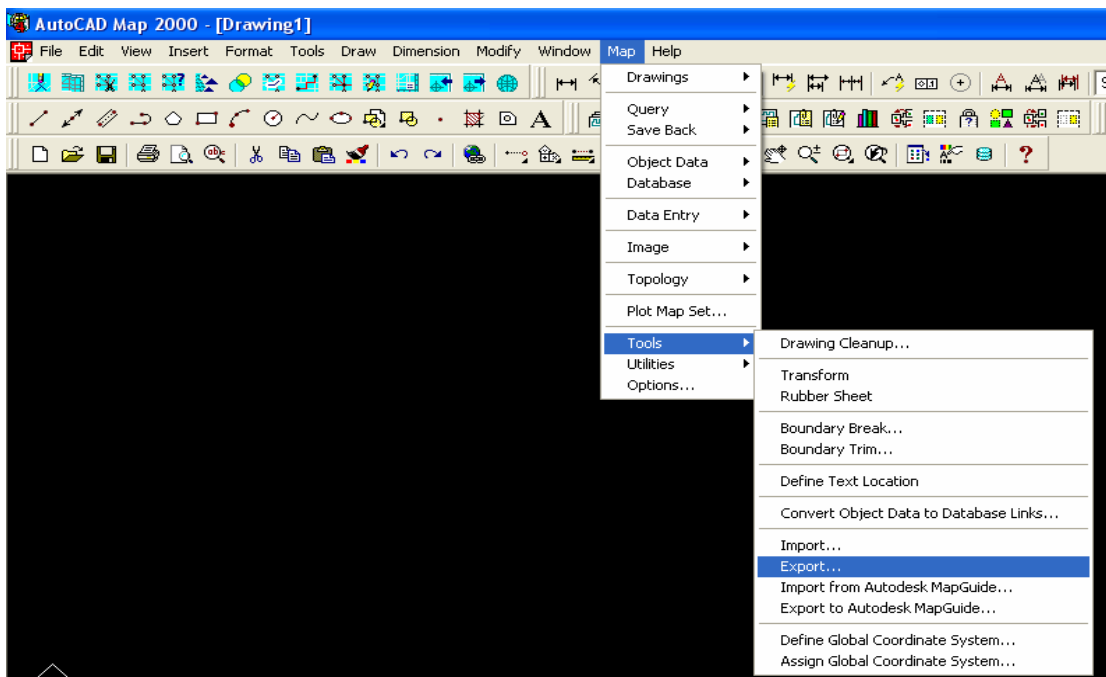
4.3.3 Exportar como archivos Shape.

Tal y como se menciona anteriormente para que los archivos generados en **Autocad Map** (*.dwg) puedan ser interpretados y leídos por el SIG,

es necesario convertir estos en archivos **shape**. Un archivo **shape** es un conjunto de 3 archivos con igual nombre y extensiones SHP, SHX y DBF. Para lograr la conversión de estos archivos se debe realizar el procedimiento que a continuación se muestra:

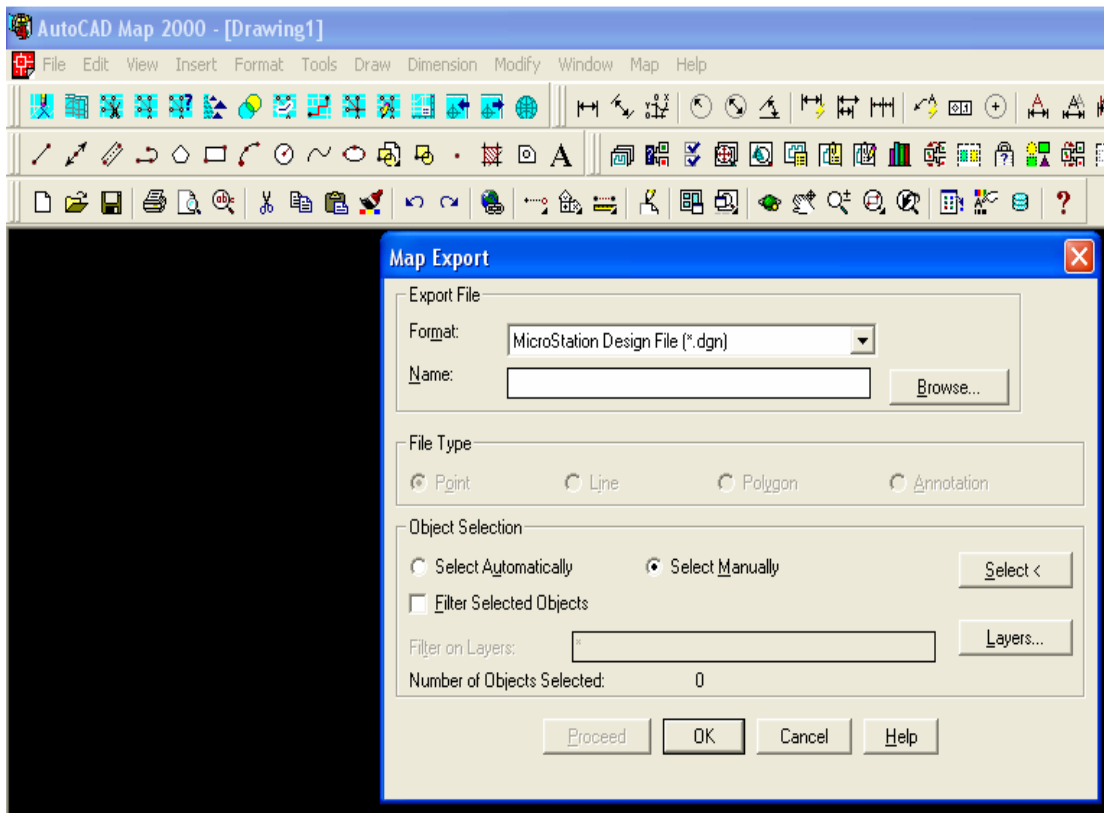
- Estando en **Autocad Map** se debe ir al Menú **Map**, seleccionar la opción **Tools**, allí se despliega un nuevo menú en donde se debe seleccionar la opción **Export**, la cual a su vez hace el despliegue de una ventana de diálogo (ver figura 3).

Figura 3. Exportar como Shape: Menú Map



- Estando ubicados en la ventana de diálogo (ver figura 4) se debe seleccionar la ruta del archivo que se va a exportar, y se debe seleccionar el formato al cual se quiere realizar la conversión, es decir **.shp**. También se debe elegir el tipo de topología y seleccionar los objetos de forma automática.

Figura 4. Exportar como Shape: Caja de diálogo



- Finalmente se da la orden aceptar y queda convertido el archivo de extensión **.dwg** en un archivo **shape**.

Este procedimiento se debe realizar para cada uno de los archivos **.dwg** del proyecto.

A continuación se explican cada una de las extensiones que hacen parte del conjunto denominado archivo **shape**:

SHP: extensión shape (geometría del objeto).

DBF: es una tabla asociada, la cual va a ser registrada en la base de datos y en ella se almacena la geometría.

SHX: es la extensión que se encarga de unir la base con el **shape (SHP y DBF)**.

4.4 METODOLOGÍA DE PROTOTIPOS

Siempre que se requiere construir o elaborar un sistema, se deben seguir una serie de pasos que ayuden a obtener el resultado oportuno de calidad. Este proceso es llevado a cabo por personas idóneas en la materia, las cuales se encargan de gestionar y adaptar los recursos disponibles de acuerdo a las necesidades del sistema. Para esto es necesario desde un principio adoptar el proceso o modelo apropiado para la creación del mismo. De acuerdo al sistema propuesto se estima que el método más apropiado para el desarrollo del mismo, es el “Modelo de Construcción de Prototipos”.

¿Por qué el Modelo de Construcción de prototipos?, se eligió este Modelo ya que desde el principio del proyecto se planteó la idea de que se debía mantener una interacción constante entre el cliente – usuario y desarrollador, con el fin de identificar los requisitos del sistema desde todos los puntos de vista de las personas implicadas; además, porque este proceso de interacción iterativo, llevaría a tener un proceso evolutivo del sistema , lo cual permitiría un desarrollo de alta calidad y familiaridad para los usuarios.

4.4.1 Determinación de Requerimientos.

La etapa de determinación de requerimientos necesitó de un trabajo inicial en grupo elaborado tanto por el cliente como por el desarrollador. Una vez se obtuvieron los requerimientos básicos para el desarrollo del sistema, se interactuó con los diferentes tipos de usuarios establecidos con el ánimo de conocer sus inquietudes. Todo esto llevo a identificar las diferentes características necesarias tanto de hardware como de software.

Este proceso además de buscar dar la mejor orientación posible al sistema, también buscaba la disminución de gastos tanto en recursos financieros, físicos, como humanos.

4.4.2 Desarrollo del Prototipo.

Para la elaboración del prototipo lo primero que se hizo, fue establecer mediante un proceso investigativo cuales serían los diferentes paquetes (Software) a emplear para el desarrollo del proyecto. Una vez hecha la elección, se instalaron estos mismos paquetes en uno de los computadores de la oficina de planeación de la universidad y en el computador personal de uno de los desarrolladores con el ánimo de desarrollar el proyecto. Al mismo tiempo que se hacía la elección e instalación del Software, se llevaba a cabo la recolección, organización y actualización de la información necesaria, para su posterior digitalización.

Seguido de este proceso se hizo el modelamiento de la base de datos con todos sus elementos, lo cual permitió la elaboración de la base de datos de la manera más óptima posible. Una vez creada la base de datos se inició el proceso de llenado y enriquecimiento de la misma, haciendo pruebas constantes de su funcionamiento.

4.4.3 Revisión del Prototipo.

Realizada por los usuarios, en donde ellos aprueban, rechazan o cambian el prototipo, además de la presentación de las observaciones y sugerencias.

A partir de esta decisión se efectúan las diferentes modificaciones al prototipo, dejando constancia escrita de las mismas.

Esta es considerada la etapa final del presente proyecto de grado, la cual será efectuada posteriormente a la presentación de los resultados del prototipo.

4.5 METODOLOGÍA DE INGENIERÍA WEB

La Ingeniería **Web** es una metodología que permite y facilita la creación de páginas Web simples o complejas a las cuales se les denomina **WebApp** (aplicación **Web**).

Para el desarrollo de estas **WebApps**, es necesario seguir un cierto procedimiento que como se mencionó anteriormente facilita su desarrollo. Este procedimiento será explicado a continuación en forma clara y breve, pues fue considerado para el desarrollo del proyecto.

Lo primero que se debe hacer es identificar las metas y objetivos de la aplicación **Web**, ésta etapa es conocida como Formulación. Una vez se ha terminado esta etapa se pasa a la parte de la planificación del proyecto, en donde se hace una estimación global del costo del mismo, se evalúan los riesgos y además se define una planificación detallada del desarrollo del proyecto. Las anteriores dos etapas se consideran fundamentales pues son la base de cualquier proyecto a desarrollar sin importar la clase y funcionalidad del mismo. Después de hecha la planificación se procede a realizar un análisis del proyecto en el cual lo que se busca es establecer los requisitos técnicos para la aplicación Web, identificar los elementos del contenido que se van a incorporar y además definir los requisitos del diseño gráfico. A continuación se entra a la etapa de mayor trabajo del proyecto, conocida como etapa de Ingeniería, en donde se desarrollan paralelamente dos tareas. La primera hace referencia a todas las tareas de diseño que se deben realizar (Diseño Arquitectónico, Diseño de Navegación y Diseño de Interfaces) y la segunda hacia los procesos de producción y adquisición del contenido, ya sea texto, gráfico o vídeo. Es en ésta etapa en donde se da todo el ingenio de las personas a cargo del proyecto y se pone a prueba el conocimiento y creatividad de los mismos, ya que las tareas de Diseño

Arquitectónico y de Navegación requieren de uso del conocimiento, mientras que el Diseño de Interfaces tiene que ver más con la creatividad y aptitudes artísticas y estéticas de los creadores.

En esta etapa de Ingeniería se debe elegir el tipo de estructura (Lineal, Reticular, Jerárquica, en Red, Combinada) a emplear en la aplicación Web de tal forma que sea la más coherente y de sentido lógico para el usuario. Así mismo, se deben definir para cada uno de los tipos de usuarios, las diferentes rutas de navegación que le permitan acceder al contenido y servicios de la aplicación Web. Esta definición de rutas es también conocida como Creación de Unidades Semánticas de Navegación (USN) y se realizan para cada uno de los roles de usuario, tal y como se mencionó anteriormente. También se debe tener especial cuidado en el diseño de la interfaz, y para esto se recomienda hacer una buena elección de la estructura de la aplicación Web. Tal y como se mencionó anteriormente esta es la etapa de la creatividad y es por eso que se debe prestar alta atención al diseño estético de la Interfaz.

Una vez creados los diseños, se avanza a la etapa de generación de páginas y pruebas, la cual es una actividad de construcción, en la cual se fusionan los diseños Arquitectónicos, de Navegación y de Interfaces para elaborar las páginas Web ejecutables. En cuanto a las pruebas ejecutadas para la revisión de la aplicación se pueden resaltar, la prueba de el modelo de contenido, la de el modelo de diseño (descubrir errores de navegación), y por último la prueba que consiste en implementar la aplicación Web en una variedad de configuraciones diferentes de entornos, para así comprobar la compatibilidad con cada configuración.

Por último está la etapa de la evaluación del cliente, que es la etapa en la cual la aplicación Web tiene su primer contacto con el mundo exterior y es en

ese momento en el que se empiezan a recibir solicitudes de cambios, que comúnmente tienen que ver con la parte estética del diseño y no con el fondo o contenido del mismo, pues anteriormente como fue mencionado en el párrafo anterior, la aplicación ya fue presa de un cierto número de pruebas que buscaban que la misma fuera bajamente sensible a la solicitud de cambios por parte de los usuarios.

4.6 DESCRIPCIÓN SITIO WEB

El sitio **Web** se creó tomando como guía, el propio sitio de la Universidad, con el ánimo de establecer una similitud en cuanto a colores, espacios y distribuciones, que le permitieran al mismo identificarse como un portal más, perteneciente a la Universidad.

Para la elaboración del sitio se emplearon las siguientes herramientas computacionales que permiten el diseño y desarrollo de aplicaciones **Web**:

- Macromedia Dreamweaver MX
- Macromedia Flash MX
- Macromedia Fireworks MX

A continuación se muestra la estructura del sitio diseñada por los desarrolladores (ver figura 5) en el cual se tuvieron en cuenta las pautas y metodologías para la realización de **WebApps**. Está contemplado dentro del diseño, que diferentes usuarios encuentren diferentes paginas de inicio así como diferentes opciones. También se presenta la forma en que está almacenada la información de cada uno de los shapes considerados en el desarrollo de la aplicación. En total son 3 shapes almacenados en tres carpetas diferentes (shapes1, shapes2, shapes3) pertenecientes a cada uno

de los diferentes pisos existentes en la sede, además cada una de estas carpetas contiene todos los elementos necesarios y contemplados para la creación de los shapes (archivos .dbf, .shp, .shx) para cada layer empleado (ver figura 6). En la figura 6 se muestra como caso general la forma en que se encuentran almacenados los archivos del piso 1 (shape1).

Figura 5. Directorio que contiene los archivos del sitio

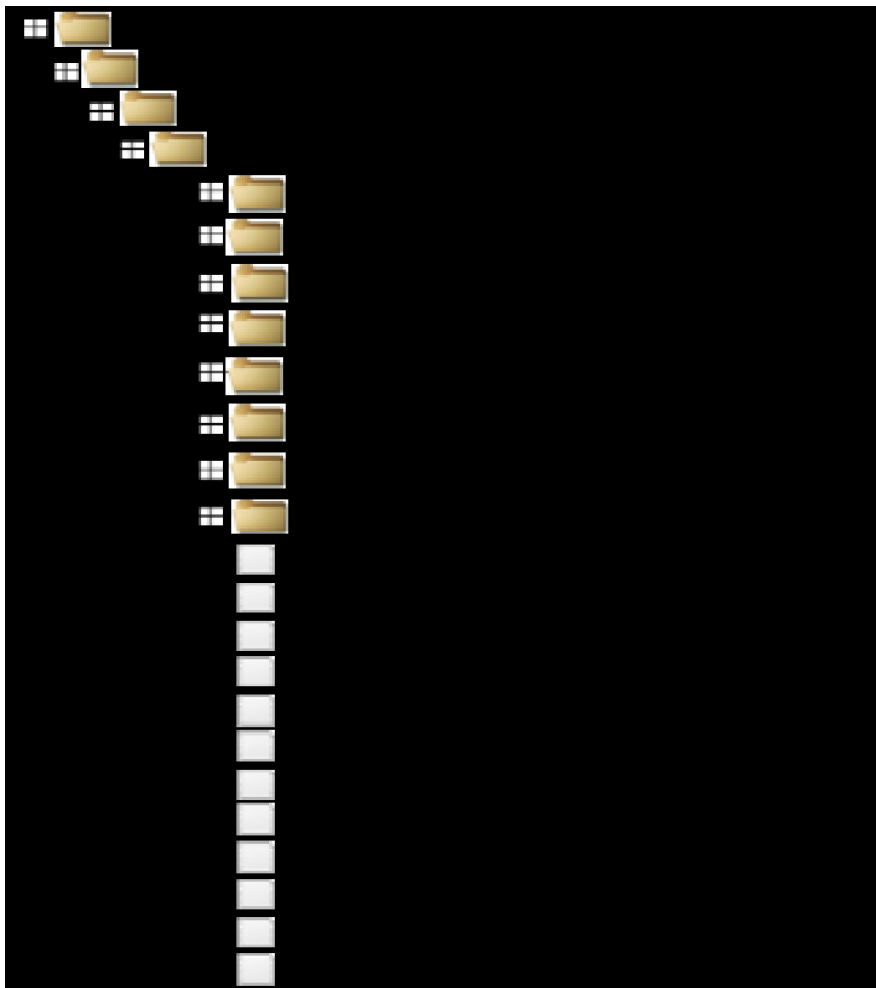
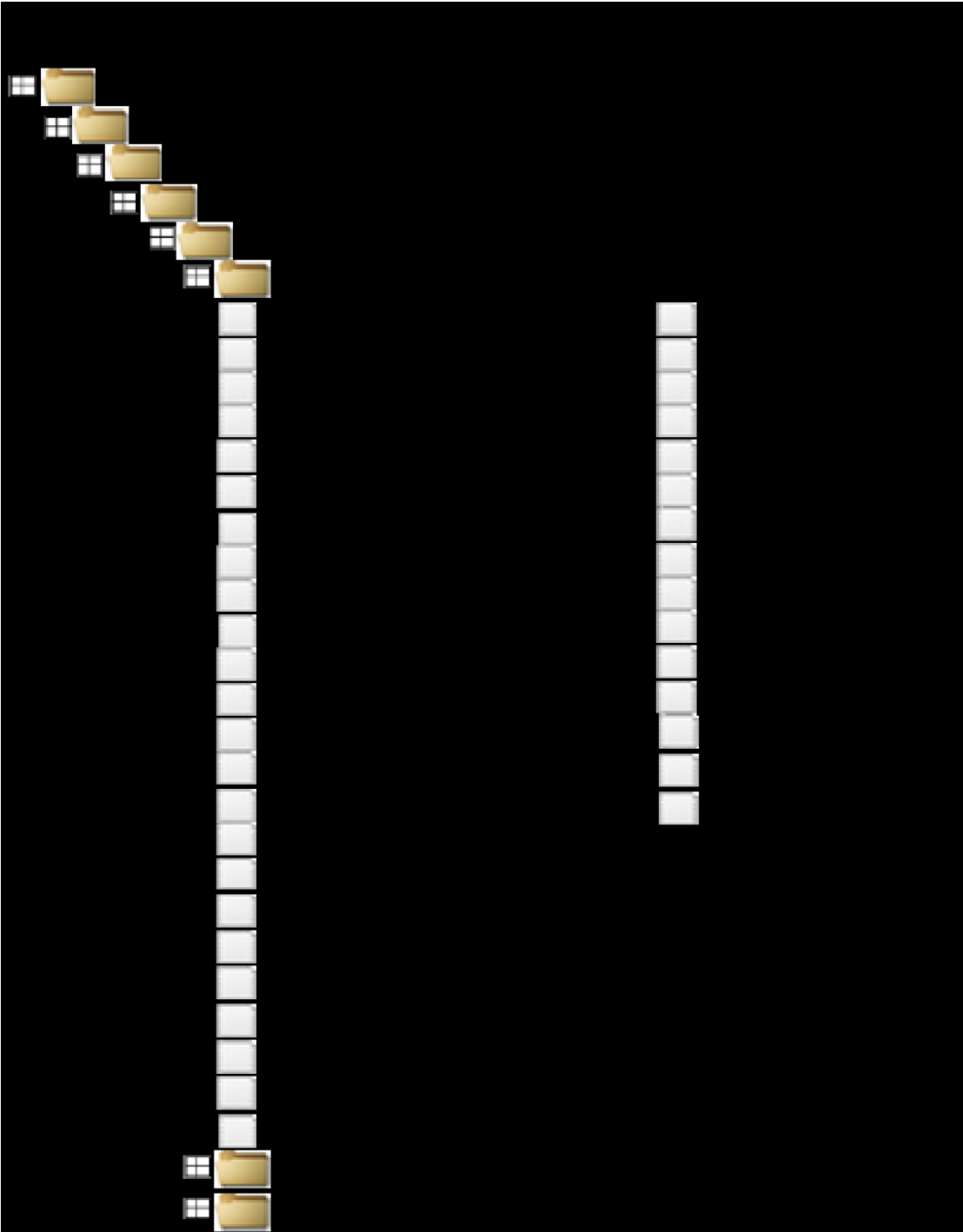


Figura 6. Directorio que contiene los shapes



5 DISEÑO Y DESARROLLO DEL SISTEMA

El sistema se diseñó y desarrolló bajo la teoría de Prototipos e Ingeniería Web, de los cuales se tomó la metodología planteada en cada uno de éstos. Lo que se hizo simplemente fue integrar las dos metodologías, pues son complementarias para la creación de sistemas tales como el planteado y creado en el presente proyecto. El proceso de diseño y desarrollo del sistema fue realizado procurando el seguimiento de dichas metodologías, razón por la cual se consideró de alta importancia el hacer una clara explicación de las mismas en el capítulo predecesor al actual.

5.1 Descripción del Sistema

Este sistema desarrollado tiene como función principal gestionar los recursos físicos y técnicos de la Sede UIS Socorro. Cuando se habla de sede UIS Socorro se incluyen a todos aquellos Edificios, áreas, lotes, y elementos que hacen parte de la infraestructura física y técnica de la misma. La gestión de estos recursos se pretende realizar de la manera más óptima posible y fue por esto que se desarrollo la **WebApp** presente. Una de las consecuencias más importantes de la creación de esta **WebApp**, aparte de la gestión de los recursos, es la elaboración, actualización y la sistematización de la información referente a la distribución de espacios en la Sede UIS Socorro.

La **WebApp** desarrollada facilita a los usuarios identificarse y conocer de forma virtual la Sede UIS Socorro. También les facilita la opción de consultas y de hacer reportes acerca del estado de los recursos físicos y técnicos de la Sede.

Para que todos los usuarios puedan identificarse, conocer virtualmente, hacer consultas y reportes, fue necesario elaborar los diferentes elementos que permitieron que dichas acciones puedan realizarse. Dichos elementos son todos los procesos y etapas desarrolladas durante el tiempo de elaboración de esta **WebApp**, y que han sido mencionados y aclarados en los capítulos anteriores. A continuación se mencionan los procesos fundamentales que permitieron el desarrollo del sistema.

Los procesos fundamentales fueron dos, el primero fue la generación de una base de datos lo suficientemente capaz de manejar toda la información necesaria en el momento (presente) y en un futuro, de tal forma que dicha base de datos y el sistema no se volvieran obsoletos en un futuro cercano. El segundo proceso fundamental fue la creación de las páginas Web y la integración de las mismas con la base de datos.

En fin lo que el sistema busca es generar un acercamiento e integración entre los usuarios y la Sede con el ánimo de mantener una relación mas estrecha, la cual permitiría un mejor control y administración de los recursos.

El sistema ha sido diseñado con el ánimo de ofrecer al usuario información veraz y actualizada, de tal forma que se le permita a éste realizar una gestión directa o indirecta de los recursos, dependiendo del tipo de usuario. A su vez se espera que el sistema produzca unos alcances mínimos que se muestran a continuación:

- Fichas descriptivas de cada uno de los recursos técnicos y físicos.
- Brinda a los usuarios autorizados la posibilidad de realizar reportes de daños directamente al sistema.

- Ubicación y posicionamiento de cada uno de los elementos pertenecientes a las diferentes infraestructuras presentadas en los Edificios de la Sede.
- Generación de reportes de daños, averías, o malos estados de cada uno de los diferentes elementos.

Se espera que en un futuro cercano se incorporen en el sistema nuevos datos que permitan mejorar el mismo, de tal forma que los alcances mínimos sean los mencionados anteriormente más los que se mencionan a continuación:

- Detalle de las estructuras de cada uno de los Edificios
- Cálculos de áreas de los diferentes espacios existentes en cada uno de los edificios de la sede.
- Inventario de Equipos Especiales.
- Ubicación de personal de acuerdo a horarios establecidos.
- De acuerdo a la disponibilidad horaria de los salones, controlar su disponibilidad.
- Elaboración y control de inventarios de cada uno de los recursos físicos no estipulados en el presente sistema, tales como sillas, libros, televisores, VCR, revistas, etc.
- Elaboración de avalúos y presupuestos.

Por último, para conocer detalladamente el sistema es necesario acudir al Manual del Usuario que se adiciona al contenido de este libro como Anexo A, pues allí, se describe de manera didáctica el funcionamiento del mismo.

5.2 Requisitos del Sistema

Para un óptimo funcionamiento del sistema se recomienda ser “montado” el mismo en una máquina con las mismas características que posee el computador donde se diseñó y desarrolló el sistema.

Dichas características son relacionadas a continuación:

Placa base

- Tipo de procesador Intel Pentium 4A, 2000 MHz (5 x 400)
- Nombre de la Placa Base Intel Pendleton 2 D845EPT2
- Chipset de la Placa Base Intel Brookdale i845E
- Memoria del Sistema 256 MB (PC2100 DDR SDRAM)
- Tipo de BIOS AMI (04/22/02)

Monitor

- Tarjeta gráfica AG315-64 (64 MB)
- Acelerador 3D SiS 315
- Monitor Samsung Sync Master 551v

Multimedia

- Tarjeta de sonido Intel 82801DB(M) ICH4(-M) - AC'97

Almacenamiento

- Disco duro 60 GB, 7200 RPM

Dispositivos de entrada

- Teclado Teclado estándar de 101/102 teclas
- Ratón Mouse PS/2 de Microsoft

Red

- Tarjeta de Red Fast Ethernet PCI Familia RTL8139 de Realtek

Por último se puede agregar que el sistema esta en capacidad de funcionar en un equipo con características técnicas y recursos de poco menor rendimiento que las antes mencionadas, siempre y cuando se le realicen las pruebas necesarias que garanticen que el sistema no colapsará. Sobra decir que si es montado el sistema en un aparato con mejores recursos y características técnicas el sistema funcionará también y de manera más eficiente en cuanto a velocidad de operación se refiere.

6 CONCLUSIONES

- Se debe buscar la continuidad del proyecto para integrarle nueva información y nuevas capacidades al sistema final.
- Para poder darle uso al prototipo es necesario capacitar a un ingeniero en el manejo y desarrollo del mismo, para que haga las veces de administrador del prototipo. Dicha capacitación consiste en: manejo de todas las herramientas software empleadas, programación en PHP/mapsript, manejo de SQL, HTML, manejo de internet y manejo de Linux.
- La información existente en la oficina de planeación UIS se encontraba desactualizada en gran medida y fue necesario levantar en su gran mayoría de nuevo la información. Se recomienda que para la continuación del trabajo sobre S.I.G. para el manejo de recursos físicos, técnicos y humanos de la Universidad se deben dividir en dos etapas muy claras: Una recolección y actualización de información y la otra diseño y montaje del prototipo. Cada una de estas etapas podría considerarse como un proyecto aparte en modalidades de práctica empresarial y proyecto de grado.
- Es necesario incentivar en la comunidad estudiantil el manejo de herramientas de software libre sobre plataformas GNU que ofrecen grandes posibilidades de trabajo a muy poco costo.
- Se debe realizar una campaña publicitaria de la herramienta S.I.G. entre toda la comunidad que es un usuario potencial de la misma, para empezar a ver funcionando el prototipo y evaluar así su respuesta bajo condiciones de uso masivo.

BIBLIOGRAFÍA

COMER, DOUGLAS E., 1988, Internet working with TCP / IP: Principles, protocols, and Architecture, New Jersey; Prentice Hall.

CLARKE, KEITH C., 2001, Getting Started with Geographic information Systems, New Jersey, Prentice Hall.

GOMEZ GOMEZ, JORGE H. Introducción a los Sistemas de Información Geográfica, Ediciones Publicaciones UIS, Colombia, 1999.

ANGARITA COGOLLO, RAÚL ALBERTO; RODRIGUEZ AVELLANEDA, ALEXIS HERLEYM, Sistema de Información Geográfico de la Planta Física del edificio de Ingeniería Civil, Universidad Industrial de Santander, Colombia, 1999.

DUARTE GARCIA, LUZ MARINA; GOMEZ CADENA, LUZ STELLA; MORENO ACEROS, LUZ MARINA, Sistema de Información Geográfica bajo ambiente web para la gestión de recursos físicos y técnicos en la Facultad de Salud de la Universidad Industrial de Santander, UIS, Colombia, 2002.

PRESSMAN, ROGER S., Ingeniería del Software, Quinta Edición, México, Mc Graw Hill, 2002.

<http://www.apache.org>

<http://mapserver.gis.umn.edu>

<http://postgis.refrations.net>

<http://www.postgresql.org>

<http://www.php.net>

<http://wombat.doc.ic.ac.uk/foldoc/>

<http://www.fcagr.unr.edu.ar/catedras/mdt/GTS/Zonaedu/pagintrod.htm>

<http://www.oreillynet.com/linux/cmd/>

<http://www.dmsolutions.ca>

ANEXO A

MANUAL DEL USUARIO

En el presente anexo se pretende explicar de forma clara y precisa la forma de navegar en el sitio **Web** y la manera en que este mismo interactúa con el prototipo, de tal forma que ninguno de los posibles usuarios que utilicen el mismo, lleguen a confundirse a la hora de emplearlo y por ende, rechacen el uso de dicho prototipo.

PÁGINA INICIO

(<http://prosiguis> o <http://prosiguis/inicio.htm>)

Esta es una página sencilla, la cual busca indicarle al usuario los requisitos mínimos que debe cumplir para poder acceder al sistema. Una vez la página ha cargado todos sus elementos, el usuario puede acceder al sitio **Web** ya sea a través del link encontrado en la parte superior derecha de la pantalla o a través del botón entrar desplegado en la presentación flash de la pagina. Tan pronto como acceda al sitio por medio de cualquiera de las formas anteriores presentadas, el usuario se encontrará en la página que contiene el Menú Principal, y allí encontrará una serie de elementos que le permitirán acceder a las diferentes herramientas creadas para interactuar con el prototipo.

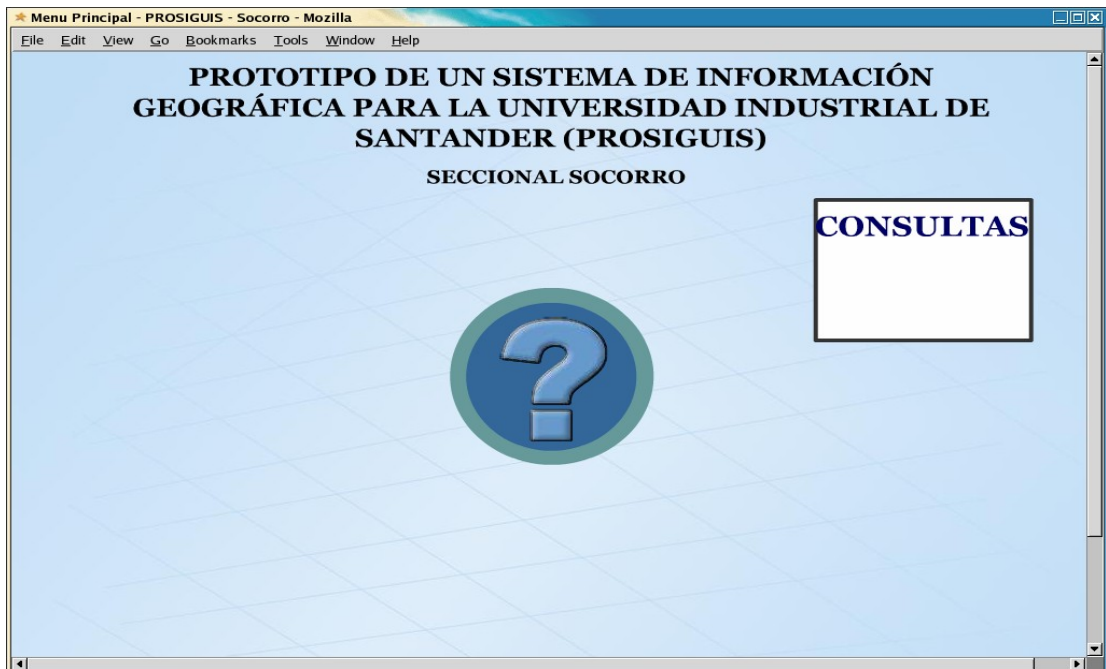


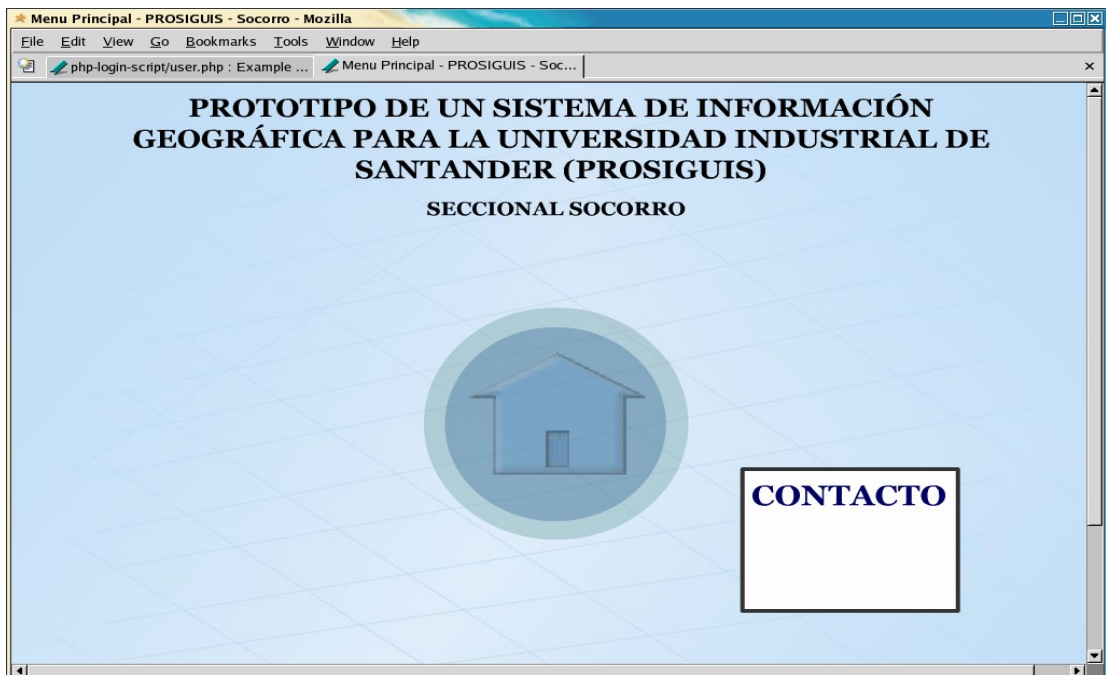
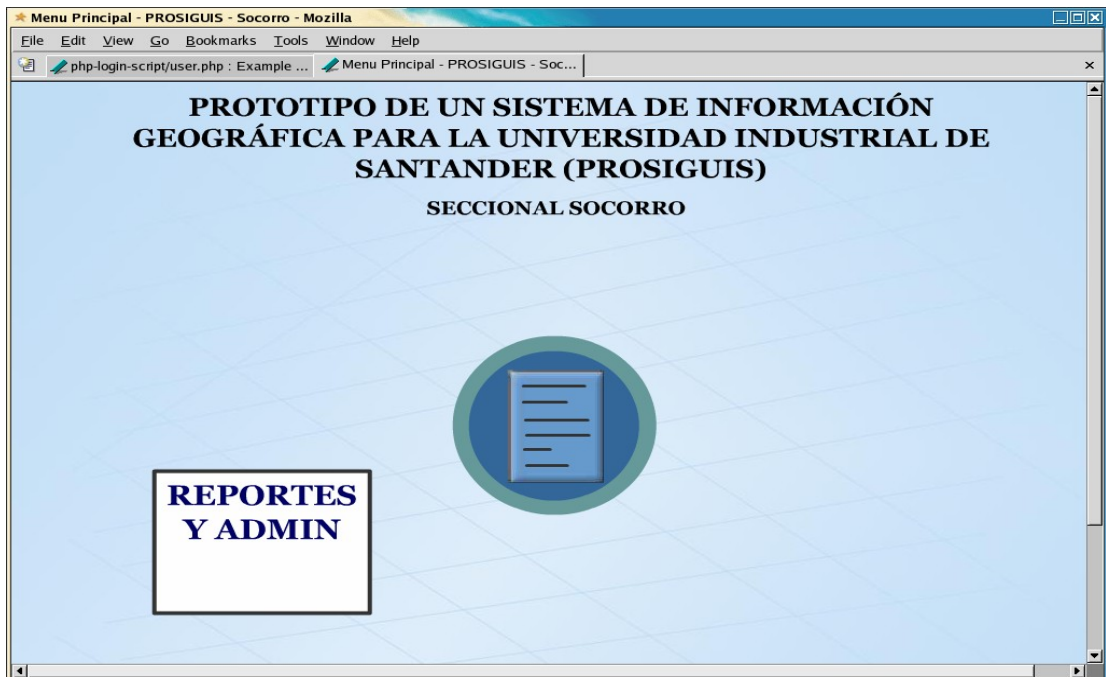
PÁGINA MENÚ PRINCIPAL

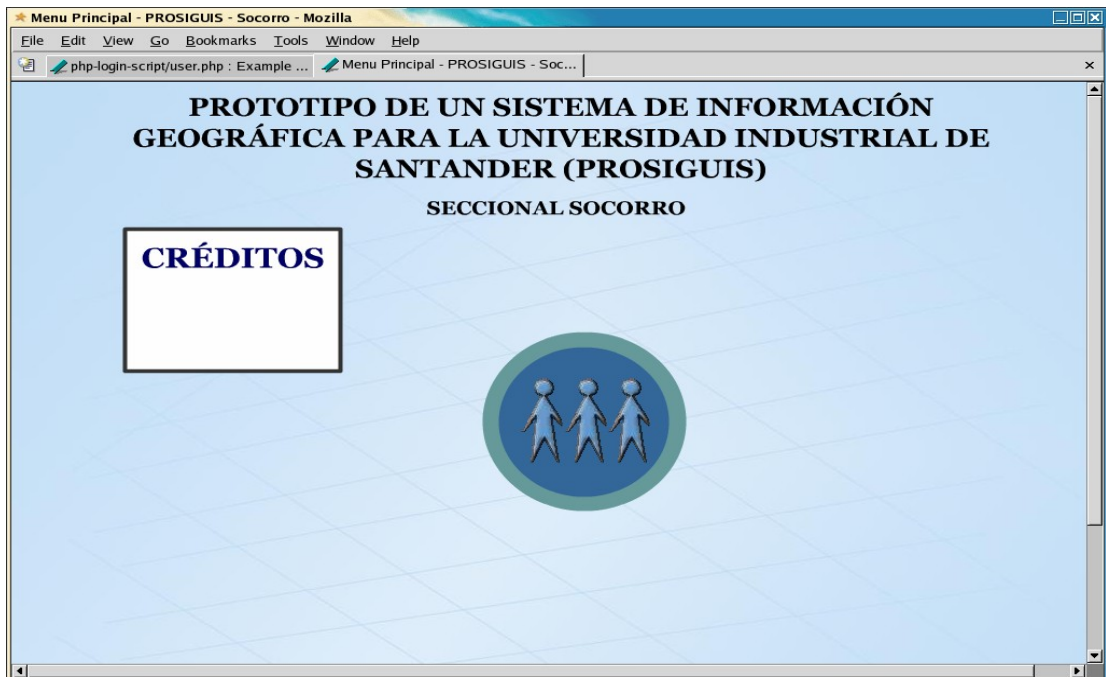
(http://prosiguis/menu_ppal.htm)

Una vez posicionado el usuario en la página del Menú Principal, este tendrá la posibilidad de disponer de un ambiente ampliamente agradable con 4 botones flash que le permitirán acceder a las diferentes opciones implementadas en el prototipo, más un quinto botón que inicialmente se encuentra inactivo (mas adelante se explicará la función de este quinto botón). Haciendo click sobre cualquiera de los botones de la pantalla inicial, se ejecutará una animación flash que al final le mostrara al usuario el nombre de la opción deseada y al mismo tiempo lo enviará hacia la página que contiene e implementa dicha opción en el sitio. Si la opción desplegada definitivamente no es la que el usuario estaba buscando tiene la opción de hacer click sobre un nuevo botón que se activa después de corrida la animación y que le permite a este retornar a la página del Menú Principal y así seleccionar nuevamente la opción que desee. Los cinco botones presentes son: Créditos, Reportes y Administración, Consultas, Inicio y por ultimo Contacto.

A continuación se muestran por medio de figuras el funcionamiento completo de la pagina Menú Principal mencionado en el párrafo anterior de forma general.







PÁGINA DE CONSULTAS

(<http://prosiguis/piso.htm>)

Una vez el usuario ha elegido la opción consultas en el Menú Principal, este será enviado a la página de consultas, la cual le permitirá al usuario tener una visión mas clara del campus universitario.

Como desde un principio se ha considerado la posibilidad de usuarios externos, que poco o nada conocen acerca de la Sede UIS - Socorro, se detalló la necesidad de presentar desde un principio una pagina simple, practica y eficiente que le permitiese al mismo conocer la totalidad del campus Universitario.

Es por esto que aquí se plantea un reconocimiento del campus por Edificios, mas no por pisos, como siempre ha sido estipulado, de tal forma que el usuario al pasar el mouse, ya sea por el plano o por los nombres de los Edificios pueda reconocer en que parte del Campus se encuentra para que así pueda elaborar su propio posicionamiento o ubicación espacial.

Sin embargo, se debe aclarar que las consultas que se realizan por medio de la página y que se muestran por medio del menú desplegable si están definidas y elaboradas para restringir el uso de la herramienta principal a cada uno de los pisos, de tal forma que al oprimir el botón de consultar con la opción elegida, este le envía una variable a la página principal de consulta necesaria para realizar el despliegue de datos, restringiendo la información a mostrar por cada piso.

Por ultimo hay que decir que los símbolos ubicados en la parte superior de la página, tanto a lado derecho como izquierdo son vínculos a las páginas

oficiales de la Universidad Industrial de Santander (UIS) y del Municipio del Socorro.



PÁGINA PRINCIPAL

(<http://prosiguis/prosiguis.phtml>)

Esta página constituye la principal página del sitio, ya que es la encargada de desplegar la información alfanumérica y espacial recogida en la sede Socorro UIS. Esta página está basada en el código de la aplicación GMap, desarrollado por dmsolutions. Se encuentran en la página seis elementos principales, que permiten entre todos ellos, la manipulación de los datos.

El primer elemento, se encuentra en la parte izquierda de la interfaz, y consiste en una lista de las capas o coberturas existentes en el prototipo. A través de esta lista, y de unas casillas de verificación que se encuentran al lado, se puede decidir que coberturas se dibujarán, las cuales a su vez estarán disponibles para realizar consultas. Para “encender” o “apagar” alguna capa, simplemente se hace click sobre su casilla de verificación, y luego sobre el botón Redibujar.

El segundo elemento, se encuentra en la parte inferior izquierda de la interfaz, y consiste en un botón que activa o desactiva la aplicación java para navegar y mostrar los datos. Por defecto, la aplicación java estará encendida. Sin embargo, también es posible realizar la navegación y consulta por medio de código html simple, combinado con código php/Mapscript. De acuerdo al código utilizado para realizar la navegación y consulta, se podrán realizar diferentes tareas, además de cambiar la forma en q se consultan los datos. Activando el código java, aparecerá la barra de herramientas para navegación y consulta en la esquina superior izquierda del mapa de navegación central. Es posible realizar consultas y navegación por zonas definidas por el usuario a través de un rectángulo. Los elementos seleccionados en la consulta, serán todos aquellos que pertenezcan a una

capa que en el momento esté “encendida” y que corten o estén totalmente contenidas en el área definida previamente. Activando el código html php/mapsript, aparecen las herramientas a utilizar para navegación y consulta, en la parte inferior derecha de la interfaz. Sólo es posible realizar consultas de modo puntual, y el elemento seleccionado será aquel que contenga al punto señalado y se encuentre en la primer capa “encendida”. El orden de búsqueda de capas encendidas, es el siguiente: Sede, Lote, Edificio, Piso, Área, División o tabique, Punto Sanitario, Punto Hidráulico, Punto Eléctrico y por último Punto Especial. Las capas Posición Luminaria, Puerta y Escalones no están disponibles para su consulta en ninguno de los modos de operación.

El tercer elemento, se encuentra en la parte central de la interfaz, y consiste en la ventana principal de navegación y despliegue de datos espaciales. Allí se mostrarán los datos espaciales consultados, además de poder escoger la herramienta java a utilizar para navegación o consulta. En la parte inferior de esta ventana, se puede encontrar la barra de escala, necesaria para conocer la dimensión real de los objetos representados.

El cuarto elemento, se encuentra en la parte superior derecha de la interfaz, y contiene el plano de referencia. En ésta imagen se muestra el plano total de la sede, además de un recuadro rojo que representa la localización actual de la ventana principal de navegación, con respecto a su ubicación en la sede.

El quinto elemento, se encuentra en la parte inferior derecha de la interfaz, exactamente debajo del plano de referencia y contiene el menú de despliegue que permite cambiar la resolución del mapa, además de tres botones de selección y un botón llamado “Observar”, para escoger el piso que se desea observar.

Los 5 elementos mencionados con anterioridad se pueden apreciar en la siguiente figura, por razones de despliegue de información de la interfaz es necesario mostrar el sexto elemento en una nueva imagen para poder apreciar de forma clara y detallada dicho elemento.



El sexto elemento, se encuentra en la parte central de la interfaz, exactamente debajo de la ventana principal de navegación y consiste de una tabla la cual despliega la información alfanumérica de los elementos seleccionados. En la siguiente figura se muestra un ejemplo de la posible información generada y desplegada en dicha tabla.

Información General											
Resultados de la consulta division, tabique o estructura											
Total de resultados = 1											
id_division	fecha_colocacion	fecha_revision	id_espec_division	tipo	cantidad_m2	id_area					
300200568	1997-06-24	2003-09-12	14	TABIQUE	20	10020					
Resultados de la consulta area											
Total de resultados = 1											
id_area	fecha_construccion	fecha_revision	nombre	id_espec_area	capacidad	foto	id_edificio	id_sede	id_piso	uso	id_uua
10020	1997-06-24	2003-09-12	AULA 303	1	30		20002	3	30002	ACADEMICO	2220
Resultados de la consulta edificio											
Total de resultados = 2											
id_edificio	id_sede	nombre	area_edificio	area_const	diseñador	constructor					
20002	30002	ADMINISTRACION Y AULAS	1986.10	2325.50	CARLOS A. CAMARGO	JAVIER JOSE PEREIRA					
20008	30002	AREAS EXTERIORES	10856.22	3524.18	CARLOS A. CAMARGO	JAVIER JOSE PEREIRA					
Resultados de la consulta sede											
Total de resultados = 1											
id_sede	id_municipio	direccion	ubicacion	telefono	director	nombre					
30002	1	CALLE 14 NO. 06 - 07	URBANA	977274931	ING. ARNULFO GALAN CADENA	SEDE UIS - SOCORRO					

En esta tabla se despliega la información de cada uno de los elementos seleccionados a través de las herramientas de consulta, sin importar el tipo de código seleccionado; es decir, sin importar si este o no, el código java activado.

Como ejemplo se procederá a explicar la información desplegada para el elemento sede. En primera instancia (primera fila) aparece el nombre de la capa consultada con el total de resultados, para este caso el total de resultados siempre será 1, pues solo esta contemplada la sede UIS – Socorro. En la segunda fila aparecen los campos que integran la tabla que contiene la información de cada una de las capas creadas. Dichos campos son en este caso, id_sede , id_municipio , direccion , ubicación , telefono , director , nombre. Por ultimo, aparecen en la fila inferior todos los atributos de los elementos seleccionados.

PÁGINA DE REGISTRO DE USUARIOS

(<http://prosiguis/reportes.php3>)

Para poder realizar un reporte de daños en la planta física de la sede UIS – Socorro, es necesario ser un usuario registrado dentro del sistema. Por tal motivo cada una de las personas que quieran realizar un reporte deberá primero validar su entrada al sistema, para posteriormente ser enviados a la página de reporte de daños, la cual será explicada a continuación, de tal forma que puedan llenar el reporte de daño estipulado allí.

Todos aquellos usuarios que no posean clave deberán enviar un correo electrónico al administrador, sugiriendo la posibilidad de obtener un usuario y una contraseña dentro del sistema.

Reportes y Administración - PROSIGUIS - Socorro - Mozilla

File Edit View Go Bookmarks Tools Window Help

UIS

**PROTOTIPO DE UN SISTEMA DE INFORMACION
GEOGRAFICO PARA LA UNIVERSIDAD INDUSTRIAL DE
SANTANDER (PROSIGUIS)**

SECCIONAL SOCORRRO

INGRESO DE USUARIOS REGISTRADOS

Usuario

Contraseña

Ingresar

Ingreso únicamente para usuarios registrados. Quien no esté registrado, debe solicitarlo por correo electrónico al administrador del PROSIGUIS. Para enviarle un correo haga [click aqui](#) o ingrese por el menú principal, sección contacto.

PÁGINA USUARIO REGISTRADO, PARA EL REPORTE DE DAÑOS

(<http://prosiguis/usuario.php3>)

Después de haberse acreditado como usuario registrado, se puede acceder a la página que presenta el formulario a llenar para reportar uno, dos o tres daños en la planta física. Se le presenta al usuario una lista desplegable con todos los nombres de los daños de la base de datos, de la cual se debe escoger el daño que presenta el elemento a reportar. Luego de escoger el daño que presenta el elemento, se procede a digitar el identificador único del elemento dañado. Por último, el usuario hace click sobre el botón reportar, para llenar el formulario de reporte en la base de datos.

Si se desea, se puede continuar a cualquier parte del sitio, desde la barra inferior.

The screenshot shows a web form titled "PROTOTIPO DE UN SISTEMA DE INFORMACION GEOGRAFICO (PROSIGUIS) SECCIONAL SOCORRO". The form is for reporting damages and includes the following elements:

- Header:** "PROTOTIPO DE UN SISTEMA DE INFORMACION GEOGRAFICO (PROSIGUIS)" and "SECCIONAL SOCORRO".
- Welcome Message:** "Bienvenido a la página de Reporte de daños. Para Realizar un reporte simplemente llene los datos solicitados a continuación. Recuerde que todos los elementos reportados deben pertenecer a la misma área."
- Damage Selection:** Three dropdown menus labeled "Escoja el primer daño a reportar", "Escoja el segundo daño a reportar", and "Escoja el tercer daño a reportar". The selected options are "Daño 1", "Daño 2", and "Daño 3".
- Element Identification:** Three text input fields labeled "Elemento que presenta el daño (Id)".
- Area Identification:** A text input field labeled "Id del área".
- Observations:** A large text area labeled "Observaciones".
- Buttons:** "Reportar" and "Cerrar Sesión".
- Footer:** A navigation bar with icons for Home, Mail, Users, Help, and Settings.

PÁGINA CREDITOS

(<http://prosiguis/creditos.htm>)


En esta página, se encuentran los autores del proyecto, así como los directores y codirectores del mismo, junto con una foto de los mismos. Cada vez que se pasa el mouse sobre el nombre de alguno de los gestores del proyecto, se despliega su foto.

PROTOTIPO DE UN SISTEMA DE INFORMACION GEOGRAFICO (PROSIGUIS)
SECCIONAL SOCORRO

CREADORES DEL PROYECTO

AUTORES

Renzo Andrés Camargo Tabares
Jose Everardo Rodriguez Ardila



Director: Ing. Carlos Alonso Camargo
Codirector: Ing. Luz Marina Duarte

Créditos

ANEXO B

DISEÑO DE LA BASE DE DATOS

MODELO DE DATOS

¿QUIEN O QUIENES VAN A UTILIZAR EL SISTEMA?

- Estudiantes UIS para poderse ubicar en el campus universitario realizando consultas a la base de datos.
- Externos, para ubicarse espacialmente en el campus realizando consultas a la base de datos.
- Integrantes de las unidades académico administrativas para ubicarse en el campus y reportar daños del mismo.
- Técnicos de planta física para programar la ejecución de reparaciones.
- Administrador del sistema para realizar consultas, análisis, mantenimiento sobre el prototipo, manipulando su base de datos sin restricciones. Además podrá realizar reportes acerca del estado de los recursos físicos y técnicos.

Para poder dar acceso y manejar los privilegios de cada uno de estos usuarios es necesario agruparlos en tres clases:

- *Invitado*: En este grupo se encuentran todos los usuarios a los cuales no les es permitido reportar daños pero si navegar consultando la

información geográfica. A esta categoría pertenecen todos los usuarios externos a la Universidad y los estudiantes. Estos usuarios no necesitarán de una clave para poder acceder al prototipo.

- *Usuario registrado*: En este grupo se encuentran todos los usuarios a los cuales les es permitido reportar daños y navegar consultando la información geográfica. A esta categoría pertenecen todos los usuarios integrantes de las diferentes Unidades Académico Administrativas. Estos usuarios necesitarán de una clave para poder acceder al prototipo.
- *Administrador*: En este grupo se encuentra el usuario al cual le es permitido realizar toda clase de funciones sobre el prototipo y editar la totalidad de la base de datos. Este usuario necesitará de una clave para poder acceder al prototipo, además deberá tener conocimiento pleno del funcionamiento del mismo.

MODELO CONCEPTUAL Y LÓGICO

Las entidades a manejar y sus respectivos atributos se encuentran en las siguientes tablas.

ENTIDAD = PUNTO_SANITARIO		
ATRIBUTO	TIPO	LONGITUD
ID_PTO_SANIT	INT4	-
FECHA_COLOCACION	DATE	-
FECHA_REVISION	DATE	-
ENTREGA	INT4	-
ID_ESPEC_SANIT	INT4	-
TIPO	VARCHAR	50
ID_AREA	INT4	-

ENTIDAD = PUNTO_HIDRAULICO		
ATRIBUTO	TIPO	LONGITUD
ID_PTO_HIDRA	INT4	-
FECHA_COLOCACION	DATE	-
FECHA_REVISION	DATE	-
ENTREGA	INT4	-
ID_ESPEC_HIDRA	INT4	-
TIPO	VARCHAR	50
PROCEDENCIA	INT4	-
CALIENTE	BOOLEAN	-
ID_AREA	INT4	-

ENTIDAD = PUNTO_ELECTRICO		
ATRIBUTO	TIPO	LONGITUD
ID_PTO_ELEC	INT4	-
FECHA_COLOCACION	DATE	-
FECHA_REVISION	DATE	-
INFLUENCIA	INT4	-
ID_ESPEC_ELEC	INT4	-
TIPO	VARCHAR	150
ID_AREA	INT4	-
PROCEDENCIA	INT4	-

ENTIDAD = PUNTO_ESPECIAL		
ATRIBUTO	TIPO	LONGITUD
ID_PTO_ESPEC	INT4	-
FECHA_COLOCACION	DATE	-
FECHA_REVISION	DATE	-
ID_ESPEC_ESPEC	INT4	-
TIPO	VARCHAR	50
ID_AREA	INT4	-
PROCEDENCIA	INT4	-

ENTIDAD = DIVISION_TABIQUE_ESTRUCTURA		
ATRIBUTO	TIPO	LONGITUD
ID_DIVISION	INT4	-
FECHA_COLOCACION	DATE	-
FECHA_REVISION	DATE	-
ID_ESPEC_DIVISION	INT4	-
ID_AREA	INT4	-
CANTIDAD_M2	INTEGER	50
TIPO	VARCHAR	50

ENTIDAD = AREA		
ATRIBUTO	TIPO	LONGITUD
ID_AREA	INT4	-
FECHA_CONSTRUCCION	DATE	-
FECHA_REVISION	DATE	-
NOMBRE	TEXT	-
ID_ESPEC_AREA	INT4	-
CAPACIDAD	VARCHAR	50
FOTO	BYTEA	-
ID_EDIFICIO	INT4	-
ID_SEDE	INT4	-
ID_PISO	INT4	-
USO	VARCHAR	50
ID_UAA	INT4	-

ENTIDAD = PISO		
ATRIBUTO	TIPO	LONGITUD
ID_PISO	INT4	-
ID_EDIFICIO	INT4	-
NOMBRE	TEXT	-

ENTIDAD = EDIFICIO		
ATRIBUTO	TIPO	LONGITUD
ID_EDIFICIO	INT4	-
ID_SEDE	INT4	-
NOMBRE	TEXT	-
AREA_EDIFICIO	VARCHAR	50
AREA_CONST	VARCHAR	50
DISENADOR	TEXT	-
CONSTRUCTOR	TEXT	-

ENTIDAD = SEDE		
ATRIBUTO	TIPO	LONGITUD
ID_SEDE	INT4	-
ID_MUNICIPIO	INT4	-
DIRECCION	VARCHAR	50
UBICACION	BOOLEAN	-
TELEFONO	VARCHAR	50
DIRECTOR	TEXT	-
NOMBRE	TEXT	-

ENTIDAD = MUNICIPIO		
ATRIBUTO	TIPO	LONGITUD
ID_MUNICIPIO	INT4	-
ID_DEPARTA	INT4	-
NOMBRE	TEXT	-

ENTIDAD = DEPARTAMENTO		
ATRIBUTO	TIPO	LONGITUD
ID_DEPARTA	INT4	-
NOMBRE	TEXT	-

ENTIDAD = DANO		
ATRIBUTO	TIPO	LONGITUD
ID_DANO	INT4	-
NOMBRE	TEXT	-
DESCRIPCION	TEXT	-

ENTIDAD = RED		
ATRIBUTO	TIPO	LONGITUD
ID_RED	INT4	-
FECHA_COLOCACION	DATE	-
FECHA_REVISION	DATE	-
ID_ESPEC_RED	INT4	-
TIPO	VARCHAR	50
INICIO	INT4	-
FIN	INT4	-
ID_AREA	INT4	-

ENTIDAD = LOTE		
ATRIBUTO	TIPO	LONGITUD
MATRICULA_INMOB	VARCHAR	50
NUM_ESCRITURA	VARCHAR	50
ID_SEDE	INT4	-
ID_MUNICIPIO	INT4	-
AREA	VARCHAR	50
VALOR	NUMERIC	50
LINDEROS	TEXT	-
PERSON_INTERV	TEXT	-

ENTIDAD = USUARIO		
ATRIBUTO	TIPO	LONGITUD
ID_USUARIO	VARCHAR	50
NOMBRE	TEXT	-
EMAIL	VARCHAR	50
CONTRASENA	VARCHAR	50
TIPO	VARCHAR	50

ENTIDAD = REPORTE		
ATRIBUTO	TIPO	LONGITUD
ID_REPORTE	INT4	-
ID_USUARIO	INT4	-
FECHA	DATE	-
HORA	TIME	-
ID_AREA	INT4	-
ID_UAA	INT4	-
NOMBRE1_DANO	TEXT	-
NOMBRE2_DANO	TEXT	-
NOMBRE3_DANO	TEXT	-
ID_ELEMENTO1	INT4	-
ID_ELEMENTO2	INT4	-
ID_ELEMENTO3	INT4	-
OBSERVACIONES	TEXT	-

ENTIDAD = ESPEC_PTO_SANIT		
ATRIBUTO	TIPO	LONGITUD
ID_ESPEC_SANIT	INT4	-
REFERENCIA	VARCHAR	150
FOTO	BYTEA	-
MATERIAL	VARCHAR	50
DIAMETRO	VARCHAR	50
PROFUNDIDAD	VARCHAR	50
REJILLA	BOOLEAN	-
AREA	VARCHAR	50
OBSERVACIONES	TEXT	-

ENTIDAD = ESPEC_PTO_ESPEC		
ATRIBUTO	TIPO	LONGITUD
ID_ESPEC_ESPEC	INT4	-
REFERENCIA	VARCHAR	150
FOTO	BYTEA	-
OBSERVACIONES	TEXT	-

ENTIDAD = ESPEC_PTO_HIDRA		
ATRIBUTO	TIPO	LONGITUD
ID_ESPEC_HIDRA	INT4	-
REFERENCIA	VARCHAR	150
FOTO	BYTEA	-
MATERIAL	VARCHAR	50
CAPACIDAD	VARCHAR	50
FLUXOMETRO	BOOLEAN	-
GRIFOS	INTEGER	50
PRESION_VALVULA	VARCHAR	50
POTENCIA	VARCHAR	50
CAUDAL	VARCHAR	50
VOLUMEN	VARCHAR	50
OBSERVACIONES	TEXT	-

ENTIDAD = ESPEC_DIVISION		
ATRIBUTO	TIPO	LONGITUD
ID_ESPEC_DIVISION	INT4	-
REFERENCIA	VARCHAR	150
FOTO	BYTEA	-
ALTURA	VARCHAR	50
MATERIAL	VARCHAR	50
OBSERVACIONES	TEXT	-

ENTIDAD = ESPEC_PTO_ELEC		
ATRIBUTO	TIPO	LONGITUD
ID_ESPEC_ELEC	INT4	-
REFERENCIA	VARCHAR	150
FOTO	BYTEA	-
MATERIAL	VARCHAR	50
VOLTAJE	VARCHAR	50
WATIAJE	VARCHAR	50
NUMERO	INTEGER	50
POTENCIA	VARCHAR	50
PUESTOS	VARCHAR	50
BREAKERS	VARCHAR	50
OBSERVACIONES	TEXT	-

ENTIDAD = UAA		
ATRIBUTO	TIPO	LONGITUD
ID_UAA	INT4	-
NOMBRE	TEXT	-
DIRECTOR	TEXT	-

ENTIDAD = ESPEC_AREA		
ATRIBUTO	TIPO	LONGITUD
ID_ESPEC_AREA	INT4	-
REFERENCIA	VARCHAR	150
FOTO	BYTEA	-
MATERIAL	VARCHAR	50
OBSERVACIONES	TEXT	-

ENTIDAD = ESPEC_RED		
ATRIBUTO	TIPO	LONGITUD
ID_ESPEC_RED	INT4	-
REFERENCIA	VARCHAR	150
MATERIAL	VARCHAR	50
DIAMETRO	VARCHAR	50
PENDIENTE	VARCHAR	50
OBSERVACIONES	TEXT	-

ENTIDAD = DANO_ELEMENTO		
ATRIBUTO	TIPO	LONGITUD
ID_DANO	INT4	-
ID_ELEM	INT4	-

DESCRIPCIÓN DE CADA ATRIBUTO

ENTIDAD = PUNTO_SANITARIO	
ATRIBUTO	DESCRIPCIÓN
ID_PTO_SANIT	Número serial único que identifica a cada punto sanitario.
FECHA_COLOCACION	Fecha en la cual se colocó o construyó por primera vez el elemento dentro del espacio físico.
FECHA_REVISION	Fecha en la cual se le realizó la última inspección visual al elemento, corroborando su estado.
ENTREGA	Identificador del punto sanitario al cual se le entrega la carga sanitaria. Siguiendo punto en la red, siguiendo la línea de flujo.
ID_ESPECIFICACION	Identificador del tipo de especificación que posee el punto sanitario. En esta especificación se encuentran ciertos atributos técnicos del punto.
TIPO	Clasifica al punto sanitario entre: sumidero, pozo de inspección, sifón, caja de inspección, salida de inodoro, salida de orinal, salida de lavamanos, contador, etc...
ID_AREA	Identificador del área al cual pertenece el punto sanitario.

ENTIDAD = DEPARTAMENTO	
ATRIBUTO	DESCRIPCIÓN
ID_DEPARTA	Número serial único que identifica a cada municipio.
NOMBRE	Nombre con el cual se conoce al municipio.

ENTIDAD = PISO	
ATRIBUTO	DESCRIPCIÓN
ID_PISO	Número serial único que identifica a cada piso de un edificio.
ID_EDIFICIO	Identificador del edificio al cual pertenece el piso.
NOMBRE	Nombre que nos identifica la ubicación del piso. Ej.: primer piso, segundo piso, tercer piso, etc...

ENTIDAD = PUNTO_ESPECIAL	
ATRIBUTO	DESCRIPCIÓN
ID_PTO_ESPEC	Número serial único que identifica a cada punto especial. Por punto especial entendemos todos los puntos de redes diferentes a la eléctrica, hidráulica y sanitaria. Ej.: Red de gas, aire acondicionado, oxígeno, etc...
FECHA_COLOCACION	Fecha en la cual se colocó o construyó por primera vez el elemento dentro del espacio físico.
FECHA_REVISION	Fecha en la cual se le realizó la última inspección visual al elemento, corroborando su estado.
ID_ESPECIFICACION	Identificador del tipo de especificación que posee el punto especial. En esta especificación se encuentran ciertos atributos técnicos del punto.
TIPO	Clasifica al punto especial entre: salida de aire, unidad central, salida de gas, contador, etc...
PROCEDENCIA	Identificador del punto especial que alimenta al punto.
ID_AREA	Identificador del área al cual pertenece el punto especial.

ENTIDAD = PUNTO_ELECTRICO	
ATRIBUTO	DESCRIPCIÓN
ID_PTO_ELEC	Número serial único que identifica a cada punto eléctrico.
FECHA_COLOCACION	Fecha en la cual se colocó o construyó por primera vez el elemento dentro del espacio físico.
FECHA_REVISION	Fecha en la cual se le realizó la última inspección visual al elemento, corroborando su estado.
INFLUENCIA	Identificador del punto eléctrico que puede apagar o encender a éste punto. Ej.: El interruptor que apaga una luminaria.
ID_ESPECIFICACION	Identificador del tipo de especificación que posee el punto eléctrico. En esta especificación se encuentran ciertos atributos técnicos del punto.
TIPO	Clasifica al punto eléctrico entre: tomacorriente, interruptor, tablero, luminaria, subestación, toma de voz, toma de datos, contador, etc...
PROCEDENCIA	Identificador del punto eléctrico que alimenta eléctricamente al punto.
ID_AREA	Identificador del área al cual pertenece el punto eléctrico.

ENTIDAD = PUNTO_HIDRAULICO	
ATRIBUTO	DESCRIPCIÓN
ID_PTO_HIDRA	Número serial único que identifica a cada punto hidráulico.
FECHA_COLOCACION	Fecha en la cual se colocó o construyó por primera vez el elemento dentro del espacio físico.
FECHA_REVISION	Fecha en la cual se le realizó la última inspección visual al elemento, corroborando su estado.
ENTREGA	Identificador del punto hidráulico al cual se le entrega la carga hidráulica. Siguiendo punto en la red, siguiendo la línea de flujo.
ID_ESPECIFICACION	Identificador del tipo de especificación que posee el punto hidráulico. En esta especificación se encuentran ciertos atributos técnicos del punto.
TIPO	Clasifica al punto hidráulico entre: grifo, válvula, tanque, siamesa, salida de ducha, salida de inodoro, salida de lavamanos, bomba, contador, etc...
PROCEDENCIA	Identificador del punto hidráulico que alimenta hidráulicamente al punto.
CALIENTE	Toma el valor de VERDADERO cuando el punto hidráulico posee acometida de agua caliente y de FALSO cuando no la posee.
ID_AREA	Identificador del área al cual pertenece el punto hidráulico.

ENTIDAD = MUNICIPIO	
ATRIBUTO	DESCRIPCIÓN
ID_MUNICIPIO	Número serial único que identifica a cada municipio.
ID_DEPARTA	Identificador del departamento al cual pertenece el municipio.
NOMBRE	Nombre con el cual se conoce al municipio.

ENTIDAD = DANO	
ATRIBUTO	DESCRIPCIÓN
ID_DANO	Número serial único que identifica a cada daño o irregularidad que presente en algún momento determinado un elemento perteneciente a la planta física.
NOMBRE	Nombre con el cual se conoce comúnmente el tipo de daño o irregularidad.
DESCRIPCION	Descripción del tipo de daño o irregularidad.

ENTIDAD = DIVISION_TABIQUE_ESTRUCTURA	
ATRIBUTO	DESCRIPCIÓN
ID_DIVISION	Número serial único que identifica a cada división, tabique o estructura. Por división se entiende cualquier elemento que separe un espacio físico de otro, sin la utilización de mampostería o concreto para tal fin. Tabique son los elementos que cumplen la misma función pero utilizando mampostería. Estructura son los elementos que soportan las cargas de la edificación o construcción.
FECHA_COLOCACION	Fecha en la cual se colocó o construyó por primera vez el elemento dentro del espacio físico.
FECHA_REVISION	Fecha en la cual se le realizó la última inspección visual al elemento, corroborando su estado.
ID_ESPECIFICACION	Identificador del tipo de especificación que posee la división, tabique o estructura. En esta especificación se encuentran ciertos atributos técnicos del elemento.
TIPO	Clasifica el elemento entre: división, tabique, muro de cortante, columna, etc...
CANTIDAD_M2	Es la cantidad en metros cuadrados que posee la superficie de una de las caras más grandes del elemento.
ID_AREA	Identificador del área al cual pertenece la división, tabique o estructura.

ENTIDAD = ESPEC_DIVISION	
ATRIBUTO	DESCRIPCIÓN
ID_DIVISION	Número serial único que identifica a cada especificación de la división, tabique o estructura.
REFERENCIA	Referencia de fábrica del producto que se convierte en elemento.
FOTO	Foto del elemento.
ALTURA	Altura de la división, tabique o estructura entre pisos consecutivos.
MATERIAL	Principal material con el cual está hecho el elemento.
OBSERVACIONES	Cualquier tipo de observación con características adicionales.

ENTIDAD = DANO_ELEMENTO	
ATRIBUTO	DESCRIPCIÓN
ID_DANO	Identificador del daño asociado al elemento.
ID_ELEM	Identificador del elemento al cual se le encuentra el daño.

ENTIDAD = AREA	
ATRIBUTO	DESCRIPCIÓN
ID_AREA	Número serial único que identifica a cada área.
FECHA_CONSTRUCCION	Fecha en la cual se colocó o construyó por primera vez el elemento dentro del espacio físico.
FECHA_REVISION	Fecha en la cual se le realizó la última inspección visual al elemento, corroborando su estado.
NOMBRE	Nombre con el cual se conoce al área de espacio físico.
ID_ESPEC_AREA	Identificador del tipo de especificación que posee el piso del área. En esta especificación se encuentran ciertos atributos técnicos del elemento.
CAPACIDAD	Capacidad máxima en número de personas a las que puede alojar cómodamente el área.
FOTO	Foto del área.
ID_EDIFICIO	Identificador del edificio al cual pertenece el área.
ID_PISO	Identificador del piso del edificio al cual pertenece el área.
ID_SEDE	Identificador de la sede a la cual pertenece el área.
USO	Tipo de uso que se le da al área. Ej.: académico, investigación, administrativo ó complementario.
ID_UAA	Identificador de la Unidad Académico Administrativa a la cual pertenece el elemento.

ENTIDAD = EDIFICIO	
ATRIBUTO	DESCRIPCIÓN
ID_EDIFICIO	Número serial único que identifica a cada edificio.
ID_SEDE	Identificador de la sede a la cual pertenece el edificio.
NOMBRE	Nombre con el cual se conoce al edificio.
AREA_EDIFICIO	Área que ocupa el primer piso de la edificación.
AREA_CONST	Suma de las áreas en proyección que se encuentran construidas en la edificación.
DISEÑADOR	Ingeniero civil, hidráulico, sanitario, eléctrico, mecánico y arquitecto que realizó los cálculos de la edificación, al momento de su construcción.
CONSTRUCTOR	Persona jurídica o natural que realizó las tareas de construcción de la edificación.

ENTIDAD = SEDE	
ATRIBUTO	DESCRIPCIÓN
ID_SEDE	Número serial único que identifica a cada sede de la Universidad Industrial de Santander.
ID_MUNICIPIO	Identificador del municipio al cual pertenece la sede.
DIRECCION	Dirección de la sede. Si no posee dirección se describe la ubicación de la sede, relativa a sitios o características del paisaje fácilmente identificables.
UBICACION	Toma el valor de VERDADERO cuando la sede se ubica dentro del casco urbano de algún municipio y de FALSO cuando se ubica fuera del casco urbano.
TELEFONO	Teléfono de la sede.
DIRECTOR	Director de la sede.
NOMBRE	Nombre con el cual se conoce a la sede.

ENTIDAD = LOTE	
ATRIBUTO	DESCRIPCIÓN
MATRICULA_INMOB	Número de la matrícula inmobiliaria de cada lote.
NUM_ESCRITURA	Número de la escritura del lote.
ID_SEDE	Identificador de la sede al cual pertenece el lote.
ID_MUNICIPIO	Identificador del municipio al cual pertenece el lote.
AREA	Área que encierra la proyección horizontal de los linderos del lote, según escritura pública.
VALOR	Valor en pesos de la totalidad del lote, sin tener en cuenta las edificaciones construidas.
LINDEROS	Descripción en prosa de los límites que demarcan al lote, según escritura pública.
PERSON_INTERV	Personas que realizaron la última transacción de compra-venta del lote, según escritura pública.

ENTIDAD = USUARIO	
ATRIBUTO	DESCRIPCIÓN
ID_USUARIO	Combinación alfanumérica única para cada usuario que permite identificar cada usuario.
NOMBRE	Nombre completo del usuario del prototipo.
EMAIL	Dirección de correo electrónico del usuario registrado.
CONTRASEÑA	Combinación alfanumérica única para cada usuario que permite validar la entrada al sistema.
TIPO	Clasifica al usuario entre: invitado, usuario registrado ó administrador. Cada usuario posee privilegios específicos dentro del prototipo.

ENTIDAD = RED	
ATRIBUTO	DESCRIPCIÓN
ID_RED	Número serial único que identifica a cada elemento conductor de la red.
FECHA_COLOCACION	Fecha en la cual se colocó o construyó por primera vez el elemento dentro del espacio físico.
FECHA_REVISION	Fecha en la cual se le realizó la última inspección visual al elemento, corroborando su estado.
ID_ESPECIFICACION	Identificador del tipo de especificación que posee el elemento conductor de la red. En esta especificación se encuentran ciertos atributos técnicos del elemento.
TIPO	Clasifica el elemento entre: tubo eléctrico, canal hidráulico, tubo hidráulico, canal de aguas lluvias, tubo sanitario, canaleta de cableado estructurado, etc...
INICIO	Identificador del punto hidráulico, sanitario o eléctrico desde donde inicia la conducción del elemento de la red. Se entiende inicio como el punto aguas arriba.
FIN	Identificador del punto hidráulico, sanitario o eléctrico donde finaliza la conducción del elemento de la red. Se entiende final como el punto aguas abajo.
ID_AREA	Identificador del área al cual pertenece el elemento conductor de la red.

ENTIDAD = ESPEC_RED	
ATRIBUTO	DESCRIPCIÓN
ID_RED	Número serial único que identifica a cada especificación del elemento conductor de la red.
REFERENCIA	Referencia de fábrica del producto que se convierte en elemento.
MATERIAL	Principal material con el cual está hecho el elemento.
DIAMETRO	Diámetro del elemento conductor de la red.
PENDIENTE	Pendiente a la cual está inclinado el elemento conductor de la red.
OBSERVACIONES	Cualquier tipo de observación con características adicionales.

ENTIDAD = ESPEC_PTO_SANIT	
ATRIBUTO	DESCRIPCIÓN
ID_PTO_SANIT	Número serial único que identifica a cada especificación de punto sanitario.
REFERENCIA	Referencia de fábrica del producto que se convierte en elemento.
FOTO	Foto del elemento.
MATERIAL	Principal material con el cual está hecho el elemento.
DIAMETRO	Diámetro de la salida sanitaria.
PROFUNDIDAD	Profundidad del pozo.
REJILLA	Toma el valor de VERDADERO cuando el sifón posee rejilla y de FALSO cuando no la posee.
AREA	Descripción de las dimensiones de las cajas de inspección, ó área efectiva de una cañuela.
OBSERVACIONES	Cualquier tipo de observación con características adicionales.

ENTIDAD = ESPEC_PTO_ELEC	
ATRIBUTO	DESCRIPCIÓN
ID_PTO_ELEC	Número serial único que identifica a cada especificación de punto eléctrico.
REFERENCIA	Referencia de fábrica del producto que se convierte en elemento.
FOTO	Foto del elemento.
MATERIAL	Principal material con el cual está hecho el elemento.
VOLTAJE	Voltaje al cual trabaja el punto eléctrico.
WATIAJE	Wattiaje consumido por el punto eléctrico.
NUMERO	Número de salidas en un tomacorriente, o de apagadores en un interruptor.
POTENCIA	Potencia a la cual trabaja una subestación.
PUESTOS	Número de puestos para colocar breakers en un tablero.
BREAKERS	Número de breakers que posee un tablero.
OBSERVACIONES	Cualquier tipo de observación con características adicionales.

ENTIDAD = REPORTE	
ATRIBUTO	DESCRIPCIÓN
ID_REPORTE	Número serial único que identifica a cada reporte enviado por un usuario.
ID_USUARIO	Identificador del usuario que envía el reporte.
FECHA	Fecha en la cual el reporte fue enviado.
HORA	Hora en la cual el reporte fue enviado.
ID_AREA	Identificador del área al cual pertenecen los elementos reportados con daños.
ID_UAA	Identificador de la Unidad Académico Administrativa a la cual pertenecen los elementos reportados con daños.
NOMBRE1_DANO	Nombre del primer daño reportado en el primer elemento del reporte. El nombre debe ser escogido entre los nombres de daño existentes.
NOMBRE2_DANO	Nombre del segundo daño reportado en el segundo elemento del reporte. El nombre debe ser escogido entre los nombres de daño existentes.
NOMBRE3_DANO	Nombre del tercer daño reportado en el tercer elemento del reporte. El nombre debe ser escogido entre los nombres de daño existentes.
ID_ELEMENTO1	Identificador del primer elemento reportado con daños.
ID_ELEMENTO2	Identificador del segundo elemento reportado con daños.
ID_ELEMENTO3	Identificador del tercer elemento reportado con daños.
OBSERVACIONES	Observaciones anotadas por el usuario que envía el reporte.

ENTIDAD = ESPEC_AREA	
ATRIBUTO	DESCRIPCIÓN
ID_AREA	Número serial único que identifica a cada especificación del piso del área.
REFERENCIA	Referencia de fábrica del producto que se convierte en elemento.
FOTO	Foto del elemento.
MATERIAL	Principal material con el cual está hecho el elemento.
OBSERVACIONES	Cualquier tipo de observación con características adicionales.

ENTIDAD = ESPEC_PTO_ESPEC	
ATRIBUTO	DESCRIPCIÓN
ID_PTO_ESPEC	Número serial único que identifica a cada especificación de punto especial.
REFERENCIA	Referencia de fábrica del producto que se convierte en elemento.
FOTO	Foto del elemento.
OBSERVACIONES	Cualquier tipo de observación con características adicionales.

ENTIDAD = ESPEC_PTO_HIDRA	
ATRIBUTO	DESCRIPCIÓN
ID_PTO_HIDRA	Número serial único que identifica a cada especificación de punto hidráulico.
REFERENCIA	Referencia de fábrica del producto que se convierte en elemento.
FOTO	Foto del elemento.
MATERIAL	Principal material con el cual está hecho el elemento.
CAPACIDAD	Capacidad de un sanitario o lavadero.
FLUXOMETRO	Toma el valor de VERDADERO cuando la salida hidráulica posee fluxómetro y de FALSO cuando no lo posee.
GRIFOS	Número de grifos que posee el punto.
PRESION_VALVULA	Presión máxima que soporta una válvula.
POTENCIA	Potencia en HP de la bomba.
CAUDAL	Caudal suministrado por el punto.
VOLUMEN	Volumen de un tanque de almacenamiento hidráulico.
OBSERVACIONES	Cualquier tipo de observación con características adicionales.

CONECTORES

Reporte – Dano: Nombre(s)_dano en Reporte y Nombre(s) en Dano.

Reporte – Usuario: Id_Usuario en Reporte y Id_Usuario en Usuario.

Dano – Dano_Elemento: Id_Dano en Dano y Id_Dano en Dano_Elemento.

Punto_Especial – Espec_Pto_Espec: Id_Espec_Espec en Punto_Especial y Id_Espec_Espec en Espec_Pto_Espec.

Punto_Especial – Dano_Elemento: Id_Pto_Espec en Punto_Especial y Id_Pto_Espec en Dano_Elemento.

Division_Tabique_Estructura – Espec_Division: Id_Espec_Division en Division_Tabique_Estructura y Id_Espec_Division en Espec_Division.

Division_Tabique_Estructura – Dano_Elemento: Id_Division en Division_Tabique_Estructura y Id_Division en Dano_Elemento.

Punto_Sanitario – Espec_Pto_Sanit: Id_Espec_Sanit en Punto_Sanitario y Id_Espec_Sanit en Espec_Pto_Sanit.

Punto_Sanitario – Dano_Elemento: Id_Pto_Sanit en Punto_Sanitario y Id_Pto_Sanit en Dano_Elemento.

Punto_Hidraulico – Espec_Pto_Hidra: Id_Espec_Hidra en Punto_Hidraulico y Id_Espec_Hidra en Espec_Pto_Hidra.

Punto_Hidraulico – Dano_Elemento: Id_Pto_Hidra en Punto_Hidraulico y Id_Pto_Hidra en Dano_Elemento.

Punto_Electrico – Espec_Pto_Elec: Id_Espec_Elec en Punto_Electrico y Id_Espec_Elec en Espec_Pto_Elec.

Punto_Electrico – Dano_Elemento: Id_Pto_Elec en Punto_Electrico y Id_Pto_Elec en Dano_Elemento.

Red – Espec_Red: Id_Espec_Red en Red y Id_Espec_Red en Espec_Red.

Red – Dano_Elemento: Id_Red en Red y Id_Red en Dano_Elemento.

Area – Espec_Area: Id_Espec_Area en Area y Id_Espec_Area en Espec_Area.

Area – Dano_Elemento: Id_Area en Area y Id_Area en Daño_Elemento.

Area – Punto_Especial: Id_Area en Area y Id_Area en Punto_Especial.

Area – Punto_Sanitario: Id_Area en Area y Id_Area en Punto_Sanitario.

Area – Punto_Hidraulico: Id_Area en Area y Id_Area en Punto_Hidraulico.

Area – Punto_Electrico: Id_Area en Area y Id_Area en Punto_Electrico.

Area – Red: Id_Area en Area y Id_Area en Red.

Area – Division_Tabique_Estructura: Id_Area en Area y Id_Area en Division_Tabique_Estructura.

Area – Piso: Id_Piso en Area y Id_Piso en Piso.

Area – UAA: Id_UAA en Area y Id_UAA en UAA.

Piso – Edificio: Id_Edificio en Piso y Id_Edificio en Edificio.

Edificio – Sede: Id_Sede en Edificio y Id_Sede en Sede.

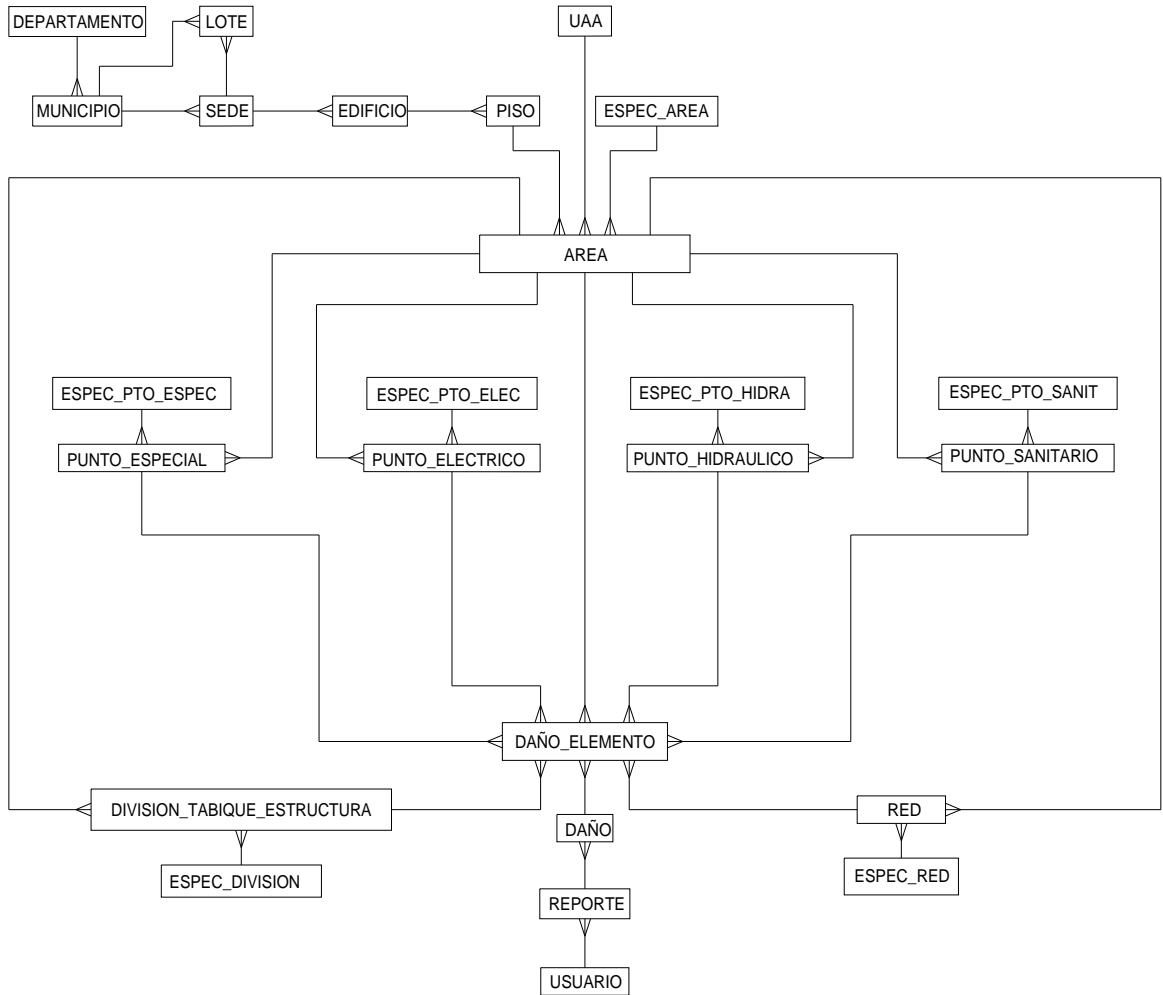
Sede – Municipio: Id_Municipio en Sede y Id_Municipio en Municipio.

Sede – Lote: Id_Sede en Sede y Id_Sede en Lote.

Municipio – Lote: Id_Municipio en Lote y Id_Municipio en Municipio.

Municipio – Departamento: Id_Departamento en Municipio y
Id_Departamento en Departamento.

MODELO ENTIDAD RELACIÓN



DICCIONARIO DE DATOS

DEFINICIÓN Y TIPO DE CADA UNO DE LOS OBJETOS:

Punto Sanitario: A esta entidad pertenecen todos aquellos elementos que pertenecen a la infraestructura física que tienen como función principal permitir la recolección y la evacuación de todas las aguas servidas y pluviales que pueden ser considerados como puntos. Entre estos podemos encontrar: Sumideros, pozos de inspección, sifones, cajas de inspección, etc. Esta entidad es de tipo punto.

Punto hidráulico: A esta entidad pertenecen todos aquellos elementos que pertenecen a la infraestructura física que tiene como función principal proveer de agua tratada a las diferentes instalaciones del edificio, que pueden ser considerados como puntos. Entre estos podemos encontrar: Grifo, válvula, ducha, sanitarios, tanque, siamesa, medidor, etc. Esta entidad es de tipo punto.

Punto Eléctrico: A esta entidad pertenecen todos aquellos elementos que pertenecen a la infraestructura física que tiene como función principal proveer el fluido eléctrico a las diferentes instalaciones del edificio y el intercambio de información (voz y datos) entre las mismas, que pueden ser considerados como puntos. Entre estos podemos encontrar: Tomacorriente, tablero, medidor, luminarias, punto de voz, punto de datos, etc. Esta entidad es de tipo punto.

Punto Especial: A esta entidad pertenecen todos aquellos elementos que pertenecen a la infraestructura física que tiene como función principal

proveer fluidos especiales diferentes al agua y al eléctrico a las diferentes instalaciones del edificio, que pueden ser considerados como puntos. Entre estos podemos encontrar: puntos proveedores de oxígeno, de gas natural, CO₂, aire acondicionado, etc. Esta entidad es de tipo punto.

División_Tabique_Estructura: A esta entidad pertenecen todos aquellos elementos arquitectónicos estructurales y no estructurales de posición vertical que permiten la separación física de espacios internos y externos. Entre estos podemos encontrar: Columnas, muros de carga, muros de mampostería, divisiones modulares, etc. Esta entidad es de tipo polígono.

Área: A esta entidad pertenecen todos aquellos recintos cerrados y espacios físicos no limitados asignados a un uso específico. Entre estos podemos encontrar: contabilidad, planta física, biblioteca, laboratorio física, cancha de microfútbol, etc. Esta entidad es de tipo polígono.

Piso: A esta entidad pertenecen todas las agrupaciones de áreas que tienen como característica principal pertenecer a una misma Edificación y que se consideran que se encuentran en el mismo nivel o división horizontal. Esta entidad es de tipo polígono.

Edificio: A esta entidad pertenecen todas las agrupaciones de pisos que tienen como característica principal pertenecer a una misma Edificación. Esta entidad es de tipo polígono.

Sede: A esta entidad pertenecen todas las agrupaciones de edificios y áreas que pertenecen a un mismo lugar de emplazamiento en el cual funcionan las instalaciones de la Universidad y que poseen como característica común que son referidas con una sola nomenclatura. Esta entidad es de tipo polígono.

Lote: A esta entidad pertenecen todos aquellos elementos que poseen como característica principal un número predial y una escritura que define su ubicación y límites de manera precisa, enmarcando un área – terreno. Esta entidad es de tipo polígono.

Municipio: A esta entidad pertenecen todos los elementos que se encuentran dentro de un territorio el cual se somete a la jurisdicción administrativa de un Alcalde.

Departamento: A esta entidad pertenecen todos los elementos que se encuentran dentro de un territorio el cual se somete a la jurisdicción administrativa de un Gobernador.

Red: A esta entidad pertenecen todos aquellos elementos que pertenecen a la infraestructura física, que hacen posible la distribución de los diferentes fluidos (eléctricos, agua, oxígeno, etc) y además pueden ser considerados como líneas. Entre estos podemos encontrar: Tubería hidráulica, canal hidráulico, canaleta de voz y datos, tubería eléctrica, etc. Esta entidad es de tipo línea.

Daño: A esta entidad pertenecen todas las averías que se pueden presentar en cada uno de los elementos físicos registrados en la base de datos.

Usuario: A esta entidad pertenecen las personas potenciales que podrían acceder al uso del prototipo.

Reporte: A esta entidad pertenecen cada una de las novedades acerca del estado de los recursos físicos y técnicos reseñadas por los diferentes usuarios del prototipo.

Espec_Pto_San: A esta entidad pertenecen todas aquellas características generales y técnicas que poseen los elementos que conforman la entidad punto_sanitario.

Espec_Pto_Hidra: A esta entidad pertenecen todas aquellas características generales y técnicas que poseen los elementos que conforman la entidad punto_hidráulico.

Espec_Pto_Elec: A esta entidad pertenecen todas aquellas características generales y técnicas que poseen los elementos que conforman la entidad punto_eléctrico.

Espec_Pto_Espec: A esta entidad pertenecen todas aquellas características generales y técnicas que poseen los elementos que conforman la entidad punto_especial.

Espec_División: A esta entidad pertenecen todas aquellas características generales y técnicas que poseen los elementos que conforman la entidad división_tabique_estructura.

Espec_Area: A esta entidad pertenecen todas aquellas características generales y técnicas que poseen los elementos que conforman la entidad area.

Espec_Red: A esta entidad pertenecen todas aquellas características generales y técnicas que poseen los elementos que conforman la entidad red.

Daño_elemento: Es una entidad que se utiliza como artificio para poder manejar el concepto de cardinalidad muchos a muchos dentro de la base de datos.

FUENTE DE LOS DATOS

Los datos fueron recolectados a través de levantamientos en cada una de las diferentes locaciones y luego fueron corroborados con la información existente en la oficina de planeación física de la Universidad Industrial de Santander en su sede principal.

También se realizaron entrevistas con el profesional de planeación física encargado de la supervisión del proyecto constructivo en la sede UIS Socorro.

DESCRIPCIÓN DE LAS ENTIDADES

ENTIDAD = PUNTO_SANITARIO					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNIC O
ID_PTO_SANIT	INT4	-	LLP	NN	U
FECHA_COLOCACION	DATE	-		N	
FECHA_REVISION	DATE	-		NN	
ENTREGA	INT4	-		N	
ID_ESPEC_SANIT	INT4	-	LLF	N	
TIPO	VARCHAR	50		NN	
ID_AREA	INT4	-	LLF	NN	

ENTIDAD = PUNTO_HIDRAULICO					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_PTO_HIDRA	INT4	-	LLP	NN	U
FECHA_COLOCACION	DATE	-		N	
FECHA_REVISION	DATE	-		NN	
ENTREGA	INT4	-		N	
ID_ESPEC_HIDRA	INT4	-	LLF	N	
TIPO	VARCHAR	50		NN	
PROCEDENCIA	INT4	-		N	
CALIENTE	BOOLEAN	-		NN	
ID_AREA	INT4	-	LLF	NN	

ENTIDAD = PUNTO_ELECTRICO					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_PTO_ELEC	INT4	-	LLP	NN	U
FECHA_COLOCACION	DATE	-		N	
FECHA_REVISION	DATE	-		NN	
INFLUENCIA	INT4	-		N	
ID_ESPEC_ELEC	INT4	-	LLF	N	
TIPO	VARCHAR	150		NN	
ID_AREA	INT4	-	LLF	NN	
PROCEDENCIA	INT4	-		N	

ENTIDAD = PISO					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_PISO	INT4	-	LLP	NN	U
ID_EDIFICIO	INT4	-	LLF	NN	
NOMBRE	TEXT	-		NN	

ENTIDAD = PUNTO_ESPECIAL					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_PTO_ESPEC	INT4	-	LLP	NN	U
FECHA_COLOCACION	DATE	-		N	
FECHA_REVISION	DATE	-		NN	
ID_ESPEC_ESPEC	INT4	-	LLF	N	
TIPO	VARCHAR	50		NN	
ID_AREA	INT4	-	LLF	NN	
PROCEDENCIA	INT4	-		N	

ENTIDAD = DIVISION_TABIQUE_ESTRUCTURA					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_DIVISION	INT4	-	LLP	NN	U
FECHA_COLOCACION	DATE	-		N	
FECHA_REVISION	DATE	-		NN	
ID_ESPEC_DIVISION	INT4	-	LLF	N	
ID_AREA	INT4	-	LLF	NN	
CANTIDAD_M2	INTEGER	50		N	
TIPO	VARCHAR	50		NN	

ENTIDAD = AREA					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_AREA	INT4	-	LLP	NN	U
FECHA_CONSTRUCCION	DATE	-		N	
FECHA_REVISION	DATE	-		NN	
NOMBRE	TEXT	-		N	
ID_ESPEC_AREA	INT4	-	LLF	N	
CAPACIDAD	VARCHAR	50		N	
FOTO	BYTEA	-		N	
ID_EDIFICIO	INT4	-		NN	
ID_SEDE	INT4	-		NN	
ID_PISO	INT4	-	LLF	NN	
USO	VARCHAR	50		N	
ID_UAA	INT4	-	LLF	N	

ENTIDAD = MUNICIPIO					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_MUNICIPIO	INT4	-	LLP	NN	U
ID_DEPARTA	INT4	-	LLF	NN	
NOMBRE	TEXT	-		NN	

ENTIDAD = EDIFICIO					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_EDIFICIO	INT4	-	LLP	NN	U
ID_SEDE	INT4	-	LLF	NN	
NOMBRE	TEXT	-		NN	
AREA_EDIFICIO	VARCHAR	50		N	
AREA_CONST	VARCHAR	50		N	
DISENADOR	TEXT	-		N	
CONSTRUCTOR	TEXT	-		N	

ENTIDAD = SEDE					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_SEDE	INT4	-	LLP	NN	U
ID_MUNICIPIO	INT4	-	LLF	NN	
DIRECCION	VARCHAR	50		N	U
UBICACION	BOOLEAN	-		N	
TELEFONO	VARCHAR	50		N	U
DIRECTOR	TEXT	-		N	
NOMBRE	TEXT	-		NN	

ENTIDAD = LOTE					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
MATRICULA_INMOB	VARCHAR	50	LLP	NN	U
NUM_ESCRITURA	VARCHAR	50		NN	U
ID_SEDE	INT4	-	LLF	NN	
ID_MUNICIPIO	INT4	-	LLF	NN	
AREA	VARCHAR	50		N	
VALOR	NUMERIC	50		N	
LINDEROS	TEXT	-		N	
PERSON_INTERV	TEXT	-		N	

ENTIDAD = REPORTE					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_REPORTE	INT4	-	LLP	NN	U
ID_USUARIO	INT4	-	LLF	NN	
FECHA	DATE	-		NN	
HORA	TIME	-		NN	
ID_AREA	INT4	-		NN	
ID_UAA	INT4	-		NN	
NOMBRE1_DANO	TEXT	-	LLF	NN	
NOMBRE2_DANO	TEXT	-	LLF	N	
NOMBRE3_DANO	TEXT	-	LLF	N	
ID_ELEMENTO1	INT4	-		NN	
ID_ELEMENTO2	INT4	-		N	
ID_ELEMENTO3	INT4	-		N	
OBSERVACIONES	TEXT	-		N	

ENTIDAD = DEPARTAMENTO					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_DEPARTA	INT4	-	LLP	NN	U
NOMBRE	TEXT	-		NN	U

ENTIDAD = RED					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_RED	INT4	-	LLP	NN	U
FECHA_COLOCACION	DATE	-		N	
FECHA_REVISION	DATE	-		NN	
ID_ESPEC_RED	INT4	-	LLF	N	
TIPO	VARCHAR	50		NN	
INICIO	INT4	-		N	
FIN	INT4	-		N	
ID_AREA	INT4	-	LLF	NN	

ENTIDAD = ESPEC_PTO_ELEC					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_ESPEC_ELEC	INT4	-	LLP	NN	U
REFERENCIA	VARCHAR	150		N	
FOTO	BYTEA	-		N	
MATERIAL	VARCHAR	50		N	
VOLTAJE	VARCHAR	50		N	
WATIAJE	VARCHAR	50		N	
NUMERO	INTEGER	50		N	
POTENCIA	VARCHAR	50		N	
PUESTOS	VARCHAR	50		N	
BREAKERS	VARCHAR	50		N	
OBSERVACIONES	TEXT	-		N	

ENTIDAD = USUARIO					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_USUARIO	VARCHAR	50	LLP	NN	U
NOMBRE	TEXT	-		NN	U
EMAIL	VARCHAR	50		N	U
CONTRASENA	VARCHAR	50		NN	U
TIPO	VARCHAR	50		NN	

ENTIDAD = DANO					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_DANO	INT4	-	LLP	NN	U
NOMBRE	TEXT	-		NN	U
DESCRIPCION	TEXT	-		NN	U

ENTIDAD = DANO_ELEMENTO					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_DANO	INT4	-	LLF	NN	U
ID_PTO_ELEC	INT4	-	LLF	NN	U
ID_AREA	INT4	-	LLF	NN	U
ID_PTO_HIDRA	INT4	-	LLF	NN	U
ID_PTO_SANIT	INT4	-	LLF	NN	U
ID_RED	INT4	-	LLF	NN	U
ID_DIVISION	INT4	-	LLF	NN	U
ID_PTO_ESPEC	INT4	-	LLF	NN	U

ENTIDAD = ESPEC_PTO_SANIT					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_ESPEC_SANIT	INT4	-	LLP	NN	U
REFERENCIA	VARCHAR	150		N	
FOTO	BYTEA	-		N	
MATERIAL	VARCHAR	50		N	
DIAMETRO	VARCHAR	50		N	
PROFUNDIDAD	VARCHAR	50		N	
REJILLA	BOOLEAN	-		N	
AREA	VARCHAR	50		N	
OBSERVACIONES	TEXT	-		N	

ENTIDAD = UAA					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_UAA	INT4	-	LLP	NN	U
NOMBRE	TEXT	-		NN	
DIRECTOR	TEXT	-		N	

ENTIDAD = ESPEC_PTO_ESPEC					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_ESPEC_ESPEC	INT4	-	LLP	NN	U
REFERENCIA	VARCHAR	150		N	
FOTO	BYTEA	-		N	
OBSERVACIONES	TEXT	-		N	

ENTIDAD = ESPEC_PTO_HIDRA					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_ESPEC_HIDRA	INT4	-	LLP	NN	U
REFERENCIA	VARCHAR	150		N	
FOTO	BYTEA	-		N	
MATERIAL	VARCHAR	50		N	
CAPACIDAD	VARCHAR	50		N	
FLUXOMETRO	BOOLEAN	-		N	
GRIFOS	INTEGER	50		NN	
PRESION_VALVULA	VARCHAR	50		N	
POTENCIA	VARCHAR	50		N	
CAUDAL	VARCHAR	50		N	
VOLUMEN	VARCHAR	50		N	
OBSERVACIONES	TEXT	-		N	

ENTIDAD = ESPEC_DIVISION					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_ESPEC_DIVISION	INT4	-	LLP	NN	U
REFERENCIA	VARCHAR	150		N	
FOTO	BYTEA	-		N	
ALTURA	VARCHAR	50		N	
MATERIAL	VARCHAR	50		N	
OBSERVACIONES	TEXT	-		N	

ENTIDAD = ESPEC_AREA					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_ESPEC_AREA	INT4	-	LLP	NN	U
REFERENCIA	VARCHAR	150		N	
FOTO	BYTEA	-		N	
MATERIAL	VARCHAR	50		N	
OBSERVACIONES	TEXT	-		N	

ENTIDAD = ESPEC_RED					
ATRIBUTO	TIPO	LONGITUD	LLP / LLF	NO NULO	UNICO
ID_ESPEC_RED	INT4	-	LLP	NN	U
REFERENCIA	VARCHAR	150		N	
MATERIAL	VARCHAR	50		N	
DIAMETRO	VARCHAR	50		N	
PENDIENTE	VARCHAR	50		N	
OBSERVACIONES	TEXT	-		N	

ANEXO C

INFORMACIÓN EN COBERTURAS

INFORMACION EN COBERTURAS

En este apartado se pretende mostrar la forma en que fueron creadas las coberturas de acuerdo al tipo de información empleada para el desarrollo de la aplicación Web. La información que se almacenó de cada uno de los edificios, fue clasificada en pequeños grupos que reúnen características similares que permitieron almacenar un sinnúmero de elementos bajo un mismo nombre (grupo).

Se consideró que todos los edificios debían ser tratados de igual forma. Fue por esto que todas las coberturas creadas se hicieron de manera general para todos los edificios (Portería y Papelería, Administración y aulas, Cafetería, Teatro al aire libre José A. Morales, Aulas, Parque y Jardín, Áreas Exteriores). A continuación se muestra la clasificación mencionada para el edificio considerado como caso general: Administración y Aulas

EDIFICIO ADMINISTRACION Y AULAS

<i>Posición Luminarias</i>	Esta cobertura contiene el contorno de las luminarias presentes en la sede.
<i>Escalones</i>	Esta cobertura contiene el contorno de las contrahuellas intermedias de cada una de las escaleras.

<i>Puertas</i>	Esta cobertura contiene los elementos que representan gráficamente una puerta.
<i>Sede</i>	Esta cobertura contiene los elementos que definen el cerramiento de una sede.
<i>Lote</i>	Esta cobertura contiene los elementos que definen los linderos de cada uno de los lotes pertenecientes a la Universidad.
<i>Edificio</i>	Esta cobertura contiene los elementos que representan la proyección horizontal del edificio.
<i>Piso</i>	Esta cobertura contiene los elementos que representan la proyección horizontal de cada nivel de un Edificio.
<i>Area</i>	Esta cobertura contiene todos aquellos recintos cerrados y espacios físicos no limitados asignados a un uso específico
<i>Division_Tabique_Estructura</i>	Esta cobertura contiene todos aquellos elementos arquitectónicos estructurales y no estructurales de posición vertical que permiten la separación física de espacios internos y externos.

<i>Punto Hidráulico</i>	Esta cobertura contiene todos aquellos elementos que pertenecen a la infraestructura física que tienen como función principal permitir la recolección y la evacuación de todas las aguas servidas y pluviales que pueden ser considerados como puntos.
<i>Punto Sanitario</i>	Esta cobertura contiene todos aquellos elementos que pertenecen a la infraestructura física que tienen como función principal permitir la recolección y la evacuación de todas las aguas servidas y pluviales que pueden ser considerados como puntos.
<i>Punto Especial</i>	Esta cobertura contiene todos aquellos elementos que pertenecen a la infraestructura física que tiene como función principal proveer fluidos especiales diferentes al agua y al eléctrico a las diferentes instalaciones del edificio, que pueden ser considerados como puntos.
<i>Punto Eléctrico</i>	Esta cobertura contiene todos aquellos elementos que pertenecen a la infraestructura física que tiene como función principal proveer el fluido eléctrico a las diferentes instalaciones del edificio y el intercambio de información (voz y datos) entre las mismas, que pueden ser considerados como puntos.

ANEXO D

SCRIPTS BASE DE DATOS

SCRIPTS BASE DE DATOS

Para la creación correcta de la base de datos se generaron una serie de scripts que permiten una rápida, eficiente y perfecta creación de la base de datos mediante el llamado inmediato y ejecución de la información almacenada en los diferentes archivos creados.

El siguiente es el script principal empleado para la generación de las tablas de la base de datos.

```
create table espec_pto_sanit (id_espec_sanit int4 primary key, referencia varchar(150), foto bytea, material varchar(50), diámetro varchar(50), profundidad varchar(50), rejilla boolean, área varchar(50), observaciones text);
```

```
create table espec_pto_hidra (id_espec_hidra int4 primary key, referencia varchar(150), foto bytea, material varchar(50), capacidad varchar(50), fluxometro boolean, grifos integer not null, presion_valvula varchar(50), potencia varchar(50), caudal varchar(50), volumen varchar (50), observaciones text);
```

```
create table espec_pto_elec (id_espec_elec int4 primary key, referencia varchar(150), foto bytea, material varchar(50), voltaje varchar(50), watiaje varchar(50), numero integer, potencia varchar(50), puestos varchar(50), breakers varchar(50), observaciones text);
```

```
create table espec_pto_espec (id_espec_espec int4 primary key, referencia varchar(150), foto bytea, observaciones text);
```

```
create table espec_division (id_espec_division int4 primary key, referencia
varchar(150), foto bytea, altura varchar(50), material varchar(50),
observaciones text);
```

```
create table espec_area (id_espec_area int4 primary key, referencia
varchar(150), foto bytea, material varchar(50), observaciones text);
```

```
create table espec_red (id_espec_red int4 primary key, referencia
varchar(150), material varchar(50), diámetro varchar(50), pendiente
varchar(50), observaciones text);
```

```
create table uaa (id_uaa int4 primary key, nombre text not null, director text);
create table usuario (id_usuario varchar(50) primary key, nombre text not null
unique, email varchar(50) unique, contrasena varchar(50) not null unique,
tipo varchar(50) not null);
```

```
create table dano (id_dano int4 primary key, nombre text not null unique,
descripción text not null unique);
```

```
create table departamento (id_departa int4 primary key, nombre text not null
unique);
```

```
create table municipio (id_municipio int4 primary key, id_departa int4 not null,
nombre text not null);
```

```
create table sede (id_sede int4 primary key, id_municipio int4 not null, foreign
key(id_municipio) references municipio, direccion varchar(50) unique,
ubicacion boolean, telefono varchar(50) unique, director text, nombre text not
null);
```

```
create table edificio (id_edificio int4 primary key, id_sede int4 not null, foreign
key(id_sede) references sede, nombre text not null, area_edificio varchar(50),
area_const varchar(50), disenador text, constructor text);
```

```
create table piso (id_piso int4 primary key, id_edificio int4 not null, foreign
key(id_edificio) references edificio, nombre text not null);
```

```
create table area (id_area int4 primary key, fecha_construccion date,
fecha_revision date not null, nombre text, id_espec_area int4, foreign
key(id_espec_area) references espec_area, capacidad varchar(50), foto
bytea, id_edificio int4 not null, id_sede int4 not null, id_piso int4 not null, uso
varchar(50), id_uaa int4, foreign key(id_uaa) references uaa);
```

```
create table punto_sanitario (id_pto_sanit int4 primary key, fecha_colocacion
date, fecha_revision date not null, entrega int4, id_espec_sanit int4, tipo
varchar(50) not null, id_area int4 not null, foreign key(id_espec_sanit)
references espec_pto_sanit, foreign key(id_area) references area);
```

```
create table punto_hidraulico (id_pto_hidra int4 primary key,
fecha_colocacion date, fecha_revision date not null, entrega int4,
id_espec_hidra int4, foreign key(id_espec_hidra) references
espec_pto_hidra, tipo varchar(50) not null, procedencia int4, caliente boolean
not null, id_area int4 not null, foreign key(id_area) references area);
```

```
create table punto_electrico (id_pto_elec int4 primary key, fecha_colocacion
date, fecha_revision date not null, influencia int4, id_espec_elec int4, foreign
key(id_espec_elec) references espec_pto_elec, tipo varchar(150) not null,
procedencia int4, id_area int4 not null, foreign key(id_area) references area);
```

```
create table punto_especial (id_pto_espec int4 primary key, fecha_colocacion date, fecha_revision date not null, id_espec_espec int4, foreign key(id_espec_espec) references espec_pto_espec, tipo varchar(50) not null, procedencia int4, id_area int4 not null);
```

```
create table division_tabique_estructura (id_division int4 primary key, fecha_colocacion date, fecha_revision date not null, id_espec_division int4, foreign key(id_espec_division) references espec_division, tipo varchar(50) not null, cantidad_m2 varchar(50), id_area int4 not null, foreign key(id_area) references area);
```

```
create table lote (matricula_inmob varchar(50) primary key, num_escritura varchar(50) not null unique, id_sede int4 not null, foreign key(id_sede) references sede , id_municipio int4 not null, foreign key(id_municipio) references municipio, area varchar(50), valor numeric(50), linderos text, person_interv text);
```

```
create table red (id_red int4 primary key, fecha_colocacion date, fecha_revision date not null, id_espec_red int4, foreign key(id_espec_red) references espec_red, tipo varchar(50) not null, inicio int4, fin int4, id_area int4 not null, foreign key(id_area) references area);
```

```
create table reporte (id_reporte int4 primary key, id_usuario int4 not null, foreign key(id_usuario) references usuario, fecha date not null, hora time not null, id_area int4 not null, id_uaa int4 not null, nombre1_dano text not null, foreign key(nombre1_dano) references dano (nombre), nombre2_dano text, foreign key(nombre2_dano) references dano (nombre) , nombre3_dano text, foreign key(nombre3_dano) references dano (nombre), id_elemento1 int4 not null, id_elemento2 int4, id_elemento3 int4, observaciones text);
```

```
create table dano_elemento (id_dano int4, foreign key(id_dano) references
dano, id_pto_elec int4 not null unique, foreign key(id_pto_elec) references
punto_electrico, id_area int4 not null unique, foreign key(id_area) references
area, id_pto_hidra int4 not null unique, foreign key(id_pto_hidra) references
punto_hidraulico, id_pto_sanit int4 not null unique, foreign key(id_pto_sanit)
references punto_sanitario, id_red int4 not null unique, foreign key(id_red)
references red, id_division int4 not null unique, foreign key(id_division)
references division_tabique_estructura, id_pto_espec int4 not null unique,
foreign key(id_pto_espec) references punto_especial );
```

```
drop table espec_pto_sanit cascade;
```

```
create table espec_pto_sanit (id_espec_sanit int4 primary key, referencia
varchar(150), foto bytea, material varchar(50), diametro varchar(50),
profundidad varchar(50), rejilla boolean, area varchar(50), observaciones
text);
```

```
copy espec_pto_sanit from
```

```
 '/mapserver/shapes_prosiguis/datos/espec_pto_sanit.txt' with null as '';
```

```
drop table espec_pto_hidra cascade;
```

```
create table espec_pto_hidra (id_espec_hidra int4 primary key, referencia
varchar(150), foto bytea, material varchar(50), capacidad varchar(50),
fluxometro boolean, grifos integer not null, presion_valvula varchar(50),
potencia varchar(50), caudal varchar(50), volumen varchar (50),
observaciones text);
```

```
copy espec_pto_hidra from
```

```
 '/mapserver/shapes_prosiguis/datos/espec_pto_hidra.txt' with null as '';
```

```
drop table espec_pto_elec cascade;
```

```
create table espec_pto_elec (id_espec_elec int4 primary key, referencia
varchar(150), foto bytea, material varchar(50), voltaje varchar(50), watiaje
```

```
varchar(50), numero integer, potencia varchar(50), puestos varchar(50),  
breakers varchar(50), observaciones text);
```

```
copy                espec_pto_elec                from  
'/mapserver/shapes_prosiguis/datos/espec_pto_elec.txt' with null as '';
```

```
drop table espec_pto_espec cascade;
```

```
create table espec_pto_espec (id_espec_espec int4 primary key, referencia  
varchar(150), foto bytea, observaciones text);
```

```
copy espec_pto_espec from  
'/mapserver/shapes_prosiguis/datos/espec_pto_espec.txt' with null as '';
```

```
drop table espec_division cascade;
```

```
create table espec_division (id_espec_division int4 primary key, referencia  
varchar(150), foto bytea, altura varchar(50), material varchar(50),  
observaciones text);
```

```
copy                espec_division                from  
'/mapserver/shapes_prosiguis/datos/espec_division.txt' with null as '';
```

```
drop table espec_area cascade;
```

```
create table espec_area (id_espec_area int4 primary key, referencia  
varchar(150), foto bytea, material varchar(50), observaciones text);
```

```
copy espec_area from '/mapserver/shapes_prosiguis/datos/espec_area.txt'  
with null as '';
```

```
drop table uaa cascade;
```

```
create table uaa (id_uaa int4 primary key, nombre text not null, director text);
```

```
copy uaa from '/mapserver/shapes_prosiguis/datos/uaa.txt' with null as '';
```

```
drop table usuario cascade;
```

```
create table usuario (id_usuario varchar(50) primary key, nombre text not null
unique, email varchar(50) unique, contrasena varchar(50) not null unique,
tipo varchar(50) not null);
```

```
copy usuario from '/mapserver/shapes_prosiguis/datos/usuario.txt' with null
as '';
```

```
drop table dano cascade;
```

```
create table dano (id_dano int4 primary key, nombre text not null unique,
descripcion text not null unique);
```

```
copy dano from '/mapserver/shapes_prosiguis/datos/dano.txt' with null as '';
```

```
drop table departamento cascade;
```

```
create table departamento (id_departa int4 primary key, nombre text not null
unique);
```

```
copy                                departamento                                from
'/mapserver/shapes_prosiguis/datos/departamento.txt' with null as '';
```

```
drop table municipio cascade;
```

```
create table municipio (id_municipio int4 primary key, id_departa int4 not null,
nombre text not null);
```

```
copy municipio from '/mapserver/shapes_prosiguis/datos/municipio.txt' with
null as '';
```

```
\i /mapserver/sql/sede.sql
```

```
\i /mapserver/sql/edificio.sql
```

```
\i /mapserver/sql/piso.sql
```

```
\i /mapserver/sql/area.sql
```

```
\i /mapserver/sql/punto_sanitario.sql
```

```
\i /mapserver/sql/punto_hidraulico.sql
```

```
\i /mapserver/sql/punto_electrico.sql
```

\i /mapserver/sql/punto_especial.sql
\i /mapserver/sql/division_tabique_estructura.sql
\i /mapserver/sql/lote.sql
\i /mapserver/sql/posicion_luminaria.sql
\i /mapserver/sql/puerta.sql
\i /mapserver/sql/escalones.sql

A continuación se presenta el contenido de uno de los archivos txt y de uno de los archivos sql mencionados en el anterior script:

- Archivo dano.txt, es aquel que contiene la lista de cada uno de los posibles daños, fallas o averías de cada uno de los diferentes elementos encontrados en la Sede.

- 1 ROTOSE REFIERE AL DAÑO DE UN ELEMENTO QUE PRESENTA LA ROTURA DE ALGUN COMPONENTE, O DE SU TOTALIDAD.
- 2 INCOMPLETO SE REFIERE AL DAÑO DE UN ELEMENTO QUE SE ENCUENTRA INCOMPLETO, POR FALTA DE ALGUNO DE SUS COMPONENTES.
- 3 CON FUGA SE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE HIDRAULICO O SANITARIO, QUE PRESENTA UNA FUGA.
- 4 OXIDADO SE REFIERE AL DAÑO DE UN ELEMENTO, QUE PRESENTA SIGNOS DE OXIDACIÓN EN ALGUNO DE SUS COMPONENTES O EN SU TOTALIDAD.

- 5 SULFATADO SE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE ELÉCTRICO, QUE PRESENTA SIGNOS DE SULFATACIÓN EN ALGUNO DE SUS COMPONENTES O EN SU TOTALIDAD.
- 6 OBSTRUIDOSE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE DE RED, QUE PRESENTA SIGNOS DE OBSTRUCCIÓN EN SU INTERIOR.
- 7 FUNDIDO SE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE ELECTRICO, QUE PRESENTA FALLAS EN SU FUNCIONAMIENTO POR FLUCTUACIÓN DEL VOLTAJE.
- 8 FALTA DE SUMINISTRO SE REFIERE AL DAÑO DE UN ELEMENTO QUE NO FUNCIONA DEBIDO A QUE NO RECIBE SUMINISTRO DE SU PROCEDENCIA.
- 9 EN CORTO CIRCUITO SE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE ELÉCTRICO, QUE NO FUNCIONA DEBIDO A QUE SE HA CAUSADO UN CORTO CIRCUITO EN EL, O QUE CAUSA EL CORTO CIRCUITO.
- 10 QUEMADO SE REFIERE AL DAÑO DE UN ELEMENTO QUE PRESENTA SIGNOS DE HABER SOPORTADO TEMPERATURAS MÁS ALTAS DE LAS ADECUADAS PARA SU CORRECTO FUNCIONAMIENTO.
- 11 SIN FLUORESCENTE O INCANDESCENTE SE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE LUMINARIAS, QUE

NO FUNCIONAN DEBIDO A LA FALTA DE SU FLUORESCENTE O INCANDESCENTE.

- 12 SIN CARCASA SE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE LUMINARIAS, QUE NO TIENEN CARCASA.
- 13 NO IMPERMEABILIZADO SE REFIERE AL DAÑO DE UN ELEMENTO QUE PRESENTA SIGNOS DE HUMEDAD, DEBIDO A SU FALTA DE IMPERMEABILIZACION.
- 14 HUMEDAD PRESENTE SE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE TABIQUES, QUE PRESENTAN SIGNOS DE HUMEDAD AFECTANDO SU INTEGRIDAD.
- 15 FISURAS SE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE TABIQUES Y COLUMNAS, QUE PRESENTAN SIGNOS DE AGRIETAMIENTO.
- 16 PINTURA SOPLADA SE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE TABIQUES Y COLUMNAS, LAS CUALES PRESENTAN BURBUJAS DEBAJO DE LA PINTURA.
- 17 PISOS LEVANTADOS SE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE PISOS, QUE PRESENTAN DESNIVELES CONSIDERABLES DEBIDO A LA NO ALINEACION DE SUS COMPONENTES.
- 18 MANCHADO SE REFIERE AL DAÑO DE UN ELEMENTO QUE PRESENTA PARCHES DE COLOR DIFERENTE AL DE SU COLOR

ORIGINAL EN ALGUNO DE SUS COMPONENTES O EN SU TOTALIDAD.

- 19 CAIDO SE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE TABIQUES, QUE PRESENTAN UNA DESTRUCCIÓN PARCIAL O TOTAL.
- 20 CABECEADO SE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE TABIQUES, QUE PRESENTAN UNA DESVIACIÓN DE SU POSICION VERTICAL.
- 21 CABLES ROTOS SE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE ELÉCTRICO, QUE PRESENTA UNA FALLA EN SU FUNCIONAMIENTO DEBIDO A QUE SUS CONDUCTORES PRESENTAN UNA ROTURA.
- 22 SUELTO SE REFIERE AL DAÑO DE UN ELEMENTO QUE NO SE ENCUENTRA BIEN ANCLADO A SU POSICIÓN.
- 23 PINTURA LEVANTADA SE REFIERE AL DAÑO DE UN ELEMENTO, PRINCIPALMENTE CARPINTERIA METÁLICA, QUE PRESENTA SITIOS EN LO CUALES LA PINTURA HA PERDIDO SE ADHERENCIA Y PARECIERA DESCASCARSE.
- 24 INSERVIBLE SE REFIERE AL DAÑO DE UN ELEMENTO QUE NO FUNCIONA POR ALGUNA RAZÓN QUE NO SE AJUSTA A NINGÚN OTRO DAÑO.

- Archivo edificio.sql, es aquel que contiene la lista de cada uno de los posibles daños, fallas o averías de cada uno de los diferentes elementos encontrados en la Sede.

```
delete from geometry_columns where f_table_name = 'punto_especial';
drop table punto_especial cascade;
create table punto_especial (gid int4, id_pto_espec int4 primary key,
fecha_colocacion date, fecha_revision date not null, id_espec_espec int4,
foreign key(id_espec_espec) references espec_pto_espec, tipo varchar(50)
not null, procedencia int4, id_area int4 not null);
select
AddGeometryColumn('sig','punto_especial','the_geom','4218','POINT',2);
begin;
```

```
insert          into          punto_especial          values
('0','205000031','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10101',GeometryFromText('POINT          (1090728.08508684
1206845.20714507)',4218) );
```

```
insert          into          punto_especial          values
('1','205000003','19970624','20030912','3','PUNTO
OXIGENO','0','10100',GeometryFromText('POINT          (1090729.40853633
1206850.69995687)',4218) );
```

```
insert          into          punto_especial          values
('2','205000005','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10100',GeometryFromText('POINT          (1090729.11573777
1206849.48473302)',4218) );
```

```
insert          into          punto_especial          values
('3','205000006','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10100',GeometryFromText('POINT          (1090728.99861834
1206848.99864348)',4218) );
```

```
insert          into          punto_especial          values
('4','205000004','19970624','20030912','3','PUNTO
OXIGENO','0','10100',GeometryFromText('POINT          (1090729.22114525
1206849.92221361)',4218) );
```

```
insert          into          punto_especial          values
('5','205000034','19970624','20030912','3','PUNTO
OXIGENO','0','10101',GeometryFromText('POINT          (1090727.67516885
1206843.50583169)',4218) );
```

```
insert          into          punto_especial          values
('6','205000033','19970624','20030912','3','PUNTO
OXIGENO','0','10101',GeometryFromText('POINT          (1090727.85084799
1206844.23496599)',4218) );
```

```
insert          into          punto_especial          values
('7','205000032','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10101',GeometryFromText('POINT          (1090727.93283158
1206844.57522867)',4218) );
```

```
insert          into          punto_especial          values
('8','205000019','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10101',GeometryFromText('POINT          (1090736.44582692
1206843.19269098)',4218) );
```

```
insert          into          punto_especial          values
('9','205000015','19970624','20030912','3','PUNTO
OXIGENO','0','10100',GeometryFromText('POINT          (1090737.76927641
1206848.68550278)',4218) );
```

```
insert          into          punto_especial          values
('10','205000018','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10100',GeometryFromText('POINT          (1090737.35935842
1206846.98418939)',4218) );
```

```
insert          into          punto_especial          values
('11','205000017','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10100',GeometryFromText('POINT          (1090737.47647785
1206847.47027893)',4218) );
```

```
insert          into          punto_especial          values
('12','205000016','19970624','20030912','3','PUNTO
OXIGENO','0','10100',GeometryFromText('POINT          (1090737.58188533
1206847.90775952)',4218) );
```

```
insert          into          punto_especial          values
('13','205000021','19970624','20030912','3','PUNTO
OXIGENO','0','10101',GeometryFromText('POINT          (1090736.21158807
1206842.2205119)',4218) );
```

```
insert          into          punto_especial          values
('14','205000020','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10101',GeometryFromText('POINT          (1090736.29357167
1206842.56077458)',4218) );
```

```
insert          into          punto_especial          values
('15','205000022','19970624','20030912','3','PUNTO
OXIGENO','0','10101',GeometryFromText('POINT          (1090736.03590893
1206841.49137759)',4218) );
```

```
insert          into          punto_especial          values
('16','205000011','19970624','20030912','3','PUNTO
OXIGENO','0','10100',GeometryFromText('POINT          (1090734.94995708
1206849.36479544)',4218) );
```

```
insert          into          punto_especial          values
('17','205000013','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10100',GeometryFromText('POINT          (1090734.65715852
1206848.14957159)',4218) );
```

```
insert          into          punto_especial          values
('18','205000014','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10100',GeometryFromText('POINT          (1090734.54003909
1206847.66348205)',4218) );
```

```
insert          into          punto_especial          values
('19','205000012','19970624','20030912','3','PUNTO
OXIGENO','0','10100',GeometryFromText('POINT          (1090734.762566
1206848.58705218)',4218) );
```

```
insert          into          punto_especial          values
('20','205000010','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10100',GeometryFromText('POINT          (1090731.72071976
1206848.34277471)',4218) );
```

```
insert          into          punto_especial          values
('21','205000009','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10100',GeometryFromText('POINT          (1090731.83783919
1206848.82886425)',4218) );
```

```
insert          into          punto_especial          values
('22','205000008','19970624','20030912','3','PUNTO
OXIGENO','0','10100',GeometryFromText('POINT          (1090731.94324667
1206849.26634483)',4218) );
```

```
insert          into          punto_especial          values
('23','205000007','19970624','20030912','3','PUNTO
OXIGENO','0','10100',GeometryFromText('POINT          (1090732.13063775
1206850.0440881)',4218) );
```

```
insert          into          punto_especial          values
('24','205000023','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10101',GeometryFromText('POINT          (1090733.62650759
1206843.87198364)',4218) );
```

```
insert          into          punto_especial          values
('25','205000027','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10101',GeometryFromText('POINT          (1090730.80718826
1206844.5512763)',4218) );
```

```
insert          into          punto_especial          values
('26','205000030','19970624','20030912','3','PUNTO
OXIGENO','0','10101',GeometryFromText('POINT          (1090730.39727027
1206842.84996291)',4218) );
```

```
insert          into          punto_especial          values
('27','205000029','19970624','20030912','3','PUNTO
OXIGENO','0','10101',GeometryFromText('POINT          (1090730.57294941
1206843.57909722)',4218) );
```

```
insert          into          punto_especial          values
('28','205000028','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10101',GeometryFromText('POINT          (1090730.65493301
1206843.9193599)',4218) );
```

```
insert          into          punto_especial          values
('29','205000025','19970624','20030912','3','PUNTO
OXIGENO','0','10101',GeometryFromText('POINT          (1090733.39226874
1206842.89980456)',4218) );
```

```
insert          into          punto_especial          values
('30','205000024','19970624','20030912','4','PUNTO          DOBLE
GAS','0','10101',GeometryFromText('POINT          (1090733.47425234
1206843.24006724)',4218) );
```

```
insert          into          punto_especial          values
('31','205000026','19970624','20030912','3','PUNTO
OXIGENO','0','10101',GeometryFromText('POINT          (1090733.2165896
1206842.17067025)',4218) );
```

```
insert          into          punto_especial          values
('32','205000038','19970624','20030912','5','EXTRACTOR
AIRE','0','10101',GeometryFromText('POINT          (1090725.89211365
1206842.89342703)',4218) );
```

```
insert          into          punto_especial          values
('33','205000037','19970624','20030912','5','EXTRACTOR
AIRE','0','10101',GeometryFromText('POINT          (1090726.61825408
1206845.90718218)',4218) );
```

```
insert          into          punto_especial          values
('34','205000036','19970624','20030912','5','EXTRACTOR
AIRE','0','10100',GeometryFromText('POINT          (1090727.35915807
1206848.98221162)',4218) );
```

```
insert          into          punto_especial          values
('35','205000035','19970624','20030912','5','EXTRACTOR
AIRE','0','10100',GeometryFromText('POINT          (1090728.08529849
1206851.99596677)',4218) );
```

```
insert          into          punto_especial          values
('36','205000039','19970624','20030912','5','EXTRACTOR
AIRE','0','10100',GeometryFromText('POINT          (1090734.12888841
1206848.94545519)',4218) );
```

```
end;
```

ANEXO E

ARCHIVO .MAP

ARCHIVO DE EXTENSION .MAP

Los archivos de extensión **.map** son el mecanismo principal de configuración del **Mapserver**. Cualquier característica asociada a una aplicación en particular se define en este archivo, aunque la mayoría de opciones puede ser cambiada a través de un formulario **Web**. En este archivo se define la manera como se van a presentar los datos dentro de la aplicación, la manera como se clasificarán, donde se guardarán los archivos gráficos temporales, la fuente de donde se extraen los datos para su visualización, etc.

Para el desarrollo de la presente aplicación fue necesario implementar 3 archivos **.map**, que permitieron hacer más dinámica y eficiente el sitio. A continuación se presenta uno de los 3 archivos **.map** utilizados en la aplicación:

NAME prosiguis
STATUS on
SIZE 550 400
SYMBOLSET "etc/symbols.sym"
SHAPEPATH "/var/www/html/prosiguis/data/shapes1/"
EXTENT 1090679.270522 1206731.665499 1090879.403434
1206904.932907
UNITS meters
IMAGECOLOR -1 -1 -1
FONTSET "etc/fonts.txt"
TRANSPARENT on

#####Inicio de la definición de la interfaz web #####

WEB
IMAGEPATH "/var/www/html/prosiguis/ms_tmp/"
IMAGEURL "/ms_tmp/"
LOG "/tmp/gmap.log"
END # WEB

Mapa de referencia

REFERENCE
IMAGE "images/piso1_3.png"
EXTENT 1090679.270522 1206731.665499 1090879.403434
1206904.932907
STATUS on
COLOR -1 -1 -1
OUTLINECOLOR 255 0 0
SIZE 200 200

END #REFERENCE MAP

Inicio del Objeto Leyenda

LEGEND

KEYSIZE 15 10

LABEL

TYPE bitmap

SIZE medium

COLOR 0 0 0

FONT 'Arial_Normal'

SIZE 8

TYPE truetype

END #LABEL

STATUS on

END #LEGEND

Inicio del Objeto Barra de escala

SCALEBAR

IMAGECOLOR 255 255 255

LABEL

COLOR 0 0 0

SIZE tiny

TYPE bitmap

END #LABEL

SIZE 200 4

COLOR 255 255 255

BACKGROUNDCOLOR 0 0 0

OUTLINECOLOR 0 0 0

UNITS meters
INTERVALS 4
STATUS on
END #SCALEBAR

Inicio del Objeto Mapa de Consulta

QUERYMAP
STYLE SELECTED
COLOR 0 255 255
END #QUERYMAP

Inicio de la definición de coberturas

1. Capa punto_especial

LAYER #punto_especial
NAME "punto especial"
LABELITEM "tipo"
LABELMAXSCALE 300
METADATA
"DESCRIPTION" "Resultados de la consulta punto especial"
"RESULT_TABLE" "punto_especial"
"RESULT_FIELDS" "id_pto_espec fecha_colocacion fecha_revision
id_espec_espec tipo procedencia id_area"
END #METADATA

TYPE point
STATUS off
DATA "punto_especial"

```

CLASS
    NAME "punto especial"
    TEMPLATE "tnt_query.html"
    COLOR 200 15 10
    SYMBOL 'plus'
    SIZE 6
LABEL
    COLOR 200 15 10
SIZE TINY
    SHADOWCOLOR 218 218 218
    SHADOWSIZE 2 2
    POSITION auto
    PARTIALS false
    END # LABEL
END # CLASS
END # LAYER

```

2. Capa punto_electrico

```

LAYER #punto_electrico
    NAME "punto electrico"
    LABELITEM "tipo"
    LABELMAXSCALE 300
    METADATA
        "DESCRIPTION" "Resultados de la consulta punto electrico"
        "RESULT_TABLE" "punto_electrico"
        "RESULT_FIELDS" "id_pto_elec fecha_colocacion fecha_revision
influencia id_espec_elec tipo procedencia id_area"
    END #METADATA
    TYPE point

```

```

STATUS off
DATA "punto_electrico"
CLASS
    NAME "punto electrico"
    TEMPLATE "ttd_query.html"
    COLOR 185 185 0
    SYMBOL 'circle'
SIZE 5
LABEL
    COLOR 150 150 0
SIZE TINY
    SHADOWCOLOR 218 218 218
    SHADOWSIZE 2 2
    POSITION auto
    PARTIALS false
    END # LABEL
END # CLASS
END # LAYER

```

3. Capa punto_hidraulico

```

LAYER #punto_hidraulico
    NAME "punto hidraulico"
    LABELITEM "tipo"
    LABELMAXSCALE 300
    METADATA
        "DESCRIPTION" "Resultados de la consulta punto hidraulico"
        "RESULT_TABLE" "punto_hidraulico"
        "RESULT_FIELDS" "id_pto_hidra fecha_colocacion fecha_revision
entrega id_espec_hidra tipo procedencia caliente id_area"

```

```

END #METADATA
TYPE point
STATUS off
DATA "punto_hidraulico"
CLASS
    NAME "punto hidraulico"
    TEMPLATE "tnt_query.html"
    COLOR 40 0 165
    SYMBOL 'tent'
    SIZE 5
LABEL
    COLOR 40 0 165
    SIZE TINY
    SHADOWCOLOR 218 218 218
    SHADOWSIZE 2 2
    POSITION auto
    PARTIALS false
    END # LABEL
END # CLASS
END # LAYER

```

4. Capa punto_sanitario

```

LAYER #punto_sanitario
    NAME "punto sanitario"
    LABELITEM "tipo"
    LABELMAXSCALE 300
    METADATA
        "DESCRIPTION" "Resultados de la consulta punto sanitario"
        "RESULT_TABLE" "punto_sanitario"

```

```

"RESULT_FIELDS" "id_pto_sanit fecha_colocacion fecha_revision
entrega id_espec_sanit tipo id_area"
END #METADATA
TYPE point
STATUS off
DATA "punto_sanitario"
CLASS
    NAME "punto sanitario"
    TEMPLATE "ttd_query.html"
    COLOR 20 164 172
    SYMBOL 'star'
    SIZE 7
LABEL
    SIZE TINY
    COLOR 20 164 172
    SHADOWCOLOR 218 218 218
    SHADOWSIZE 2 2
    POSITION auto
    PARTIALS false
    END # LABEL
END # CLASS
END # LAYER

```

```
##### 5. Division_Tabique_Estructura #####
```

```

LAYER #Division_Tabique_Estructura
    NAME "division_tabique_estructura"
    METADATA
        "DESCRIPTION" "Resultados de la consulta division, tabique o
estructura"

```

```

"RESULT_TABLE" "division_tabique_estructura"
"RESULT_FIELDS" "id_division fecha_colocacion fecha_revision
id_espec_division tipo cantidad_m2 id_area"
END #METADATA
TYPE polygon
STATUS on
DATA "division_tabique_estructura"
CLASS
    NAME "division_tabique_estructura"
    TEMPLATE "ttd_query.html"
    COLOR 240 200 185
    OUTLINECOLOR 0 0 0
LABEL
    COLOR 0 0 0
    SHADOWCOLOR 218 218 218
    SHADOWSIZE 2 2
    POSITION auto
    PARTIALS false
    END # LABEL
END # CLASS
END # LAYER

```

6. Area

```

LAYER #area
    NAME "area"
    LABELITEM "nombre"
    LABELMAXSCALE 300
    METADATA

```

```

"DESCRIPTION" "Resultados de la consulta area"
"RESULT_TABLE" "area"
"RESULT_FIELDS" "id_area fecha_construccion fecha_revision
nombre
id_espec_area capacidad foto id_edificio id_sede id_piso uso id_uaa"
END #METADATA
TYPE polygon
STATUS on
DATA "area"
TRANSPARENCY 0
CLASS
    NAME "area"
    TEMPLATE "ttt_query.html"
    OUTLINECOLOR 5 5 130
LABEL
    COLOR 50 50 50
    SIZE TINY
    SHADOWCOLOR 218 218 218
    SHADOWSIZE 2 2
    POSITION cc
    PARTIALS false
END # LABEL
END # CLASS
END # LAYER

```

7. Piso

```

LAYER #piso
    NAME "piso"
    METADATA

```

```

"DESCRIPTION" "Resultados de la consulta piso"
"RESULT_TABLE" "piso"
"RESULT_FIELDS" "id_piso id_edificio nombre"
END #METADATA
TYPE polygon
STATUS off
DATA "piso"
TRANSPARENCY 0
CLASS
    NAME "piso"
    TEMPLATE "ttt_query.html"
    OUTLINECOLOR 30 200 30
LABEL
    COLOR 0 0 0
    SHADOWCOLOR 218 218 218
    SHADOWSIZE 2 2
    POSITION auto
    PARTIALS false
    END # LABEL
END # CLASS
END # LAYER

```

8. Capa Edificio

```

LAYER #edificio
    NAME "edificio"
    LABELITEM "nombre"
    LABELMINSCALE 300
    LABELMAXSCALE 2000
    METADATA

```

```

"DESCRIPTION" "Resultados de la consulta edificio"
"RESULT_TABLE" "edificio"
"RESULT_FIELDS" "id_edificio id_sede nombre area_edificio
area_const
diseñador constructor"
END #METADATA
TYPE polygon
STATUS on
DATA "edificio"
TRANSPARENCY 0
CLASS
    NAME "edificio"
    TEMPLATE "ttt_query.html"
    OUTLINECOLOR 175 175 175
LABEL
    COLOR 32 77 117
SIZE MEDIUM
    SHADOWCOLOR 250 250 250
    SHADOWSIZE 2 2
    POSITION CC
    PARTIALS FALSE
    END # LABEL
END # CLASS
END # LAYER

```

9. Capa lote

```

LAYER #lote
    NAME "lote"
    METADATA

```

```

"DESCRIPTION" "Resultados de la consulta lote"
"RESULT_TABLE" "lote"
"RESULT_FIELDS" "matricula_inmob num_escritura id_sede
id_municipio area valor linderos"
END #METADATA
TYPE polygon
STATUS off
DATA "lote"
TRANSPARENCY 0
CLASS
    NAME "lote"
    TEMPLATE "ttt_query.html"
    OUTLINECOLOR 0 0 0
LABEL
    COLOR 0 0 0
    SHADOWCOLOR 218 218 218
    SHADOWSIZE 2 2
    POSITION auto
    PARTIALS false
    END # LABEL
END # CLASS
END # LAYER

```

10. Capa sede

```

LAYER #sede
    NAME "sede"
    LABELITEM "nombre"
    LABELMAXSCALE 2000
    LABELMINSSCALE 300

```

```

TRANSPARENCY 0
METADATA
"DESCRIPTION" "Resultados de la consulta sede"
"RESULT_TABLE" "sede"
"RESULT_FIELDS" "id_sede id_municipio direccion ubicacion telefono
director nombre"
END #METADATA
TYPE polygon
STATUS on
DATA "sede"
CLASS
    NAME "sede"
    TEMPLATE "ttt_query.html"
    OUTLINECOLOR 180 100 30
LABEL
    COLOR 0 0 0
    SIZE GIANT
    SHADOWCOLOR 218 218 218
    SHADOWSIZE 2 2
    POSITION auto
    PARTIALS false
    FORCE TRUE
    END # LABEL
    END # CLASS
END # LAYER

```

11. Capa posicion luminaria

```

LAYER #posicion luminaria
    NAME "posicion luminaria"

```

```
TYPE line
STATUS off
METADATA
"NOQUERY" "1"
END #METADATA
DATA "posicion_luminaria"
CLASS
    NAME "posicion luminaria"
    TEMPLATE "ttd_query.html"
    COLOR 0 0 0
END # CLASS
END # LAYER
```

12. Capa puerta

```
LAYER #puerta
    NAME "puerta"
    TYPE line
    STATUS on
    METADATA
    "NOQUERY" "1"
    END #METADATA
    DATA "puerta"
    CLASS
        NAME "puerta"
        TEMPLATE "ttd_query.html"
        COLOR 250 0 70
    END # CLASS
END # LAYER
```

13. Capa escalones

```
LAYER #escalones
  NAME "escalones"
  TYPE line
  STATUS on
  METADATA
  "NOQUERY" "1"
  END #METADATA
  DATA "escalones"
  CLASS
    NAME "escalones"
    TEMPLATE "ttt_query.html"
    COLOR 100 100 100
  END # CLASS
END # LAYER

END # MAPFILE
```

ANEXO F

ARCHIVOS .PHTML y PHP3

ARCHIVO DE EXTENSION .PHTML

Los archivos de extensión **.phtml** contienen una combinación de código **html** plano, código **php/mapsript** y código **java**, los cuales al ser accedidos a través del servidor Web permiten la visualización de los datos. Sin embargo, gran parte del código necesario para completar las tareas requeridas para la visualización se consigna en otro archivo.

A continuación se presenta el archivo **.phtml** utilizado en la aplicación:

```

<html>
<?php
// Carga los modulos requeridos.
if (PHP_OS == "WINNT" || PHP_OS == "WIN32")
{
    if (!extension_loaded("MapScript"))
    {
        if (phpversion() < 4)
            dl("php3_mapscript.dll"); // PHP3
        else
            dl("php_mapscript_36.dll"); // PHP4
    }
    if (!extension_loaded("DBase"))
    {
        if (phpversion() < 4)
            dl("php3_dbase.dll"); // PHP3
        else
            dl("php_dbase.dll"); // PHP4
    }
}
else
{
    if (!extension_loaded("MapScript"))
        dl("php_mapscript.so"); // Igual para PHP3 y PHP4
}

include("prosiguis.php3");

if ($HTTP_FORM_VARS["MAP_NAME"])
    $gpoMap
ms_newMapObj(strval($HTTP_FORM_VARS["MAP_NAME"]));
else {
    header ("location: piso.htm");
    exit ();
}

// Establece los extents del .map
// Extent máximo = 10%
$dfTmpDeltaX = ($gpoMap->extent->maxx - $gpoMap->extent->minx)/10;
$dfTmpDeltaY = ($gpoMap->extent->maxy - $gpoMap->extent->miny)/10;
// Por ahora se deshabilitan los topes del 10%, hasta tener nuevos datos.
$dfTmpDeltaX = 0;
$dfTmpDeltaY = 0;

$dfMapExtMinX = $gpoMap->extent->minx;

```

```

$dfMapExtMinY = $gpoMap->extent->miny;
$dfMapExtMaxX = $gpoMap->extent->maxx;
$dfMapExtMaxY = $gpoMap->extent->maxy;

$dfMaxExtMinX = $gpoMap->extent->minx - $dfTmpDeltaX;
$dfMaxExtMinY = $gpoMap->extent->miny - $dfTmpDeltaY;
$dfMaxExtMaxX = $gpoMap->extent->maxx + $dfTmpDeltaX;
$dfMaxExtMaxY = $gpoMap->extent->maxy + $dfTmpDeltaY;

GMap75CheckClick();

$gszCommand = $HTTP_FORM_VARS["CMD"];
if ( sizeof($gszCommand) == 0)
{
    $gszCommand = "ZOOM_IN";
}

// Cambia entre el applet Rosa y HTML simple
if (strval($HTTP_FORM_VARS["PREVIOUS_MODE"])==0)
    $gbIsHtmlMode = 0; // Arrancar con Java por defecto.
else
    $gbIsHtmlMode = intval($HTTP_FORM_VARS["PREVIOUS_MODE"]);

if ($HTTP_FORM_VARS[JavaOn_x])
    $gbIsHtmlMode = 0;

if ($HTTP_FORM_VARS[JavaOff_x])
    $gbIsHtmlMode = 1;

function IsHtmlMode()
{
    GLOBAL $gbIsHtmlMode;
    return $gbIsHtmlMode;
}
?>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function MapSizeSelected()
{
    if (document.forms[0].MapSize.selectedIndex != 0 &&
        document.forms[0].MapSize.selectedIndex !=1)
        document.forms[0].submit();
}
</SCRIPT>
<br>

```

```
<TD><DIV ALIGN="center"><B><FONT COLOR="#336699" FACE="Arial"
SIZE="5">PROTOTIPO DE UN SISTEMA DE INFORMACION
GEOGRAFICO (PROSIGUIS)</DIV></B></FONT>
<br>
```

```
<br>
<TD><DIV ALIGN="center"><B><FONT COLOR="#336699" FACE="Arial"
SIZE="3">SECCIONAL SOCORRO</DIV></B></FONT></TD>
<br>
```

```
<TITLE>PROSIGUIS - CONSULTA DE DATOS</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-
8859-1">
</HEAD>
```

```
<BODY BGCOLOR="#336699" BACKGROUND="/images/fondo.jpg">
<FORM METHOD=POST NAME="myform">
<TABLE BORDER="0" CELSPACING="5" CELLPADDING="0">
<TR>
<TD VALIGN="TOP" ALIGN="CENTER">
<TABLE BORDER="0" CELSPACING="0" CELLPADDING="0">
<TR ALIGN="CENTER">
<TD BGCOLOR="#E2EFF5">&nbsp;</TD>
<TD BGCOLOR="#E2EFF5"><FONT FACE="Arial, Helvetica, sans-
serif" SIZE="2"><B>Leyenda</B></FONT></TD>
<TD BGCOLOR="#E2EFF5">&nbsp;</TD>
</TR>
<TR>
<TD BGCOLOR="#E2EFF5"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="1"></TD>
<TD BGCOLOR="#E2EFF5"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="1"></TD>
<TD BGCOLOR="#E2EFF5"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="1"></TD>
</TR>
<TR>
<TD BGCOLOR="#87C6C9"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
<TD BGCOLOR="#87C6C9"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
<TD BGCOLOR="#87C6C9"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
</TR>
<TR>
<TD BGCOLOR="#87C6C9">&nbsp;</TD>
```

```

        <TD BGCOLOR="#87C6C9">
            <TABLE WIDTH="100%" BORDER="0" CELLSPACING="1"
CELLPADDING="2">
                <TR BGCOLOR="#FFFFFF">
                    <TD ALIGN="CENTER"><IMG SRC="images/icon_eye.gif"
WIDTH="17" HEIGHT="11"></TD>
                    <TD ALIGN="CENTER">&nbsp;</TD>
                    <TD><FONT FACE="Arial, Helvetica, sans-serif"
SIZE="2"><B>Layer</B></FONT></TD>
                </TR>

                <TR BGCOLOR="#FFFFFF">
                    <TD ALIGN="CENTER">
                        <INPUT TYPE="checkbox" NAME="punto especial" VALUE="Y"
                        <?php if (GMapGetStatus("punto especial") == 1) echo
"CHECKED"; ?>>
                    </TD>
                    <TD ALIGN="CENTER"><IMG
SRC="images/logo_pto_espec.gif"></TD>
                    <TD><FONT FACE="Arial, Helvetica, sans-serif" SIZE="2">Punto
Especial</FONT></TD>
                </TR>

                <TR BGCOLOR="#FFFFFF">
                    <TD ALIGN="CENTER">
                        <INPUT TYPE="checkbox" NAME="punto electrico" VALUE="Y"
                        <?php if (GMapGetStatus("punto electrico") == 1) echo
"CHECKED"; ?>>
                    </TD>
                    <TD ALIGN="CENTER"><IMG
SRC="images/logo_pto_elec.gif"></TD>
                    <TD><FONT FACE="Arial, Helvetica, sans-serif" SIZE="2">Punto
El&eacute;ctrico</FONT></TD>
                </TR>

                <TR BGCOLOR="#FFFFFF">
                    <TD ALIGN="CENTER">
                        <INPUT TYPE="checkbox" NAME="punto hidraulico" VALUE="Y"
                        <?php if (GMapGetStatus("punto hidraulico") == 1) echo
"CHECKED"; ?>>
                    </TD>
                    <TD ALIGN="CENTER"><IMG
SRC="images/logo_pto_hidra.gif"></TD>
                    <TD><FONT FACE="Arial, Helvetica, sans-serif" SIZE="2">Punto
Hidraulico</FONT></TD>

```

```

</TR>

    <TR BGCOLOR="#FFFFFF">
    <TD ALIGN="CENTER">
    <INPUT TYPE="checkbox" NAME="punto sanitario" VALUE="Y"
        <?php if (GMapGetStatus("punto sanitario") == 1) echo
"CHECKED"; ?>>
    </TD>
    <TD
        ALIGN="CENTER"><IMG
SRC="images/logo_pto_sanit.gif"></TD>
    <TD><FONT FACE="Arial, Helvetica, sans-serif" SIZE="2">Punto
Sanitario</FONT></TD>
    </TR>

    <TR BGCOLOR="#FFFFFF">
    <TD ALIGN="CENTER">
    <INPUT TYPE="checkbox" NAME="division_tabique_estructura"
VALUE="Y"
        <?php if (GMapGetStatus("division_tabique_estructura")
== 1) echo "CHECKED"; ?>>
    </TD>
    <TD
        ALIGN="CENTER"><IMG
SRC="images/logo_division.gif"></TD>
    <TD><FONT FACE="Arial, Helvetica, sans-serif"
SIZE="2">Divisi&oacute;n o tabique</FONT></TD>
    </TR>

    <TR BGCOLOR="#FFFFFF">
    <TD ALIGN="CENTER">
    <INPUT TYPE="checkbox" NAME="area" VALUE="Y"
        <?php if (GMapGetStatus("area") == 1) echo
"CHECKED"; ?>>
    </TD>
    <TD
        ALIGN="CENTER"><IMG
SRC="images/logo_area.gif"></TD>
    <TD><FONT FACE="Arial, Helvetica, sans-serif"
SIZE="2">Area</FONT></TD>
    </TR>

    <TR BGCOLOR="#FFFFFF">
    <TD ALIGN="CENTER">
    <INPUT TYPE="checkbox" NAME="piso" VALUE="Y"
        <?php if (GMapGetStatus("piso") == 1) echo
"CHECKED"; ?>>
    </TD>

```

```

        <TD
        ALIGN="CENTER"><IMG
SRC="images/logo_piso.gif"></TD>
        <TD><FONT
        FACE="Arial, Helvetica, sans-serif"
SIZE="2">Piso</FONT></TD>
</TR>

```

```

        <TR BGCOLOR="#FFFFFF">
        <TD ALIGN="CENTER">
        <INPUT TYPE="checkbox" NAME="edificio" VALUE="Y"
        <?php if (GMapGetStatus("edificio") == 1) echo
"CHECKED"; ?>>
        </TD>
        <TD
        ALIGN="CENTER"><IMG
SRC="images/logo_edificio.gif"></TD>
        <TD><FONT
        FACE="Arial, Helvetica, sans-serif"
SIZE="2">Edificio</FONT></TD>
</TR>

```

```

        <TR BGCOLOR="#FFFFFF">
        <TD ALIGN="CENTER">
        <INPUT TYPE="checkbox" NAME="lote" VALUE="Y"
        <?php if (GMapGetStatus("lote") == 1) echo "CHECKED";
?>>
        </TD>
        <TD
        ALIGN="CENTER"><IMG
SRC="images/logo_lote.gif"></TD>
        <TD><FONT
        FACE="Arial, Helvetica, sans-serif"
SIZE="2">Lote</FONT></TD>
</TR>

```

```

        <TR BGCOLOR="#FFFFFF">
        <TD ALIGN="CENTER">
        <INPUT TYPE="checkbox" NAME="sede" VALUE="Y"
        <?php if (GMapGetStatus("sede") == 1) echo
"CHECKED"; ?>>
        </TD>
        <TD
        ALIGN="CENTER"><IMG
SRC="images/logo_sede.gif"></TD>
        <TD><FONT
        FACE="Arial, Helvetica, sans-serif"
SIZE="2">Sede</FONT></TD>
</TR>

```

```

        <TR BGCOLOR="#FFFFFF">
        <TD ALIGN="CENTER">

```

```

        <INPUT TYPE="checkbox" NAME="posicion luminaria"
VALUE="Y"
        <?php if (GMapGetStatus("posicion luminaria") == 1)
echo "CHECKED"; ?>>
        </TD>
        <TD ALIGN="CENTER"><IMG
SRC="images/logo_pos_lum.gif"></TD>
        <TD><FONT FACE="Arial, Helvetica, sans-serif"
SIZE="2">Posicion Luminaria</FONT></TD>
</TR>

<TR BGCOLOR="#FFFFFF">
<TD ALIGN="CENTER">
<INPUT TYPE="checkbox" NAME="puerta" VALUE="Y"
        <?php if (GMapGetStatus("puerta") == 1) echo
"CHECKED"; ?>>
        </TD>
        <TD ALIGN="CENTER"><IMG
SRC="images/logo_puerta.gif"></TD>
        <TD><FONT FACE="Arial, Helvetica, sans-serif"
SIZE="2">Puerta</FONT></TD>
</TR>

<TR BGCOLOR="#FFFFFF">
<TD ALIGN="CENTER">
<INPUT TYPE="checkbox" NAME="escalones" VALUE="Y"
        <?php if (GMapGetStatus("escalones") == 1) echo
"CHECKED"; ?>>
        </TD>
        <TD ALIGN="CENTER"><IMG
SRC="images/logo_escalones.gif"></TD>
        <TD><FONT FACE="Arial, Helvetica, sans-serif"
SIZE="2">Escalones</FONT></TD>
</TR>

<TR BGCOLOR="#FFFFFF">
<TD ALIGN="CENTER">
<INPUT TYPE=IMAGE SRC="images/icon_redraw.gif"
WIDTH="19" HEIGHT="19" NAME=redraw BORDER=0>
        </TD>
        <TD COLSPAN="2"><FONT FACE="Arial, Helvetica, sans-serif"
SIZE="2">Redibujar</FONT></TD>
</TR>
</TABLE>

```

```

        </TD>
        <TD BGCOLOR="#87C6C9">&nbsp;</TD>
    </TR>

    <TR>
        <TD      BGCOLOR="#87C6C9"><IMG      SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
        <TD      BGCOLOR="#87C6C9"><IMG      SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
        <TD      BGCOLOR="#87C6C9"><IMG      SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
    </TR>
</TABLE>

</TD>
<TD VALIGN=TOP ALIGN="CENTER">
    <TABLE BORDER="0" CELLSPACING="0" CELLPADDING="0">
        <TR>
            <TD><IMG      SRC="images/corner_TL_LB.jpg"      WIDTH="4"
HEIGHT="4"></TD>
            <TD      BGCOLOR="#E2EFF5"><IMG      SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
            <TD><IMG      SRC="images/corner_TR_LB.jpg"      WIDTH="4"
HEIGHT="4"></TD>
        </TR>
        <TR ALIGN="CENTER">
            <TD BGCOLOR="#E2EFF5">&nbsp;</TD>
            <TD BGCOLOR="#E2EFF5"><FONT FACE="Arial, Helvetica, sans-
serif" SIZE="2"><B>SEDE SOCORRO UIS</B></FONT></TD>
            <TD BGCOLOR="#E2EFF5">&nbsp;</TD>
        </TR>
        <TR>
            <TD      BGCOLOR="#87C6C9"><IMG      SRC="images/pixel.gif"
WIDTH="1" HEIGHT="1"></TD>
            <TD      BGCOLOR="#87C6C9"><IMG      SRC="images/pixel.gif"
WIDTH="1" HEIGHT="1"></TD>
            <TD      BGCOLOR="#87C6C9"><IMG      SRC="images/pixel.gif"
WIDTH="1" HEIGHT="1"></TD>
        </TR>
        <TR>
            <TD      BGCOLOR="#87C6C9"><IMG      SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
            <TD      BGCOLOR="#87C6C9"><IMG      SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>

```

```

        <TD BGCOLOR="#87C6C9"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
    </TR>
    <TR>
        <TD BGCOLOR="#87C6C9">&nbsp;</TD>
        <TD BGCOLOR="#999999">
            <TABLE WIDTH="400" BORDER="0" CELLSPACING="1"
CELLPADDING="0" HEIGHT="300">
                <TR BGCOLOR="FFFFFF" ALIGN="CENTER">
                    <TD><?php GMapDrawMap() ?></TD>
                </TR>
            </TABLE>
        </TD>
        <TD BGCOLOR="#87C6C9">&nbsp;</TD>
    </TR>
    <TR>
        <TD BGCOLOR="#87C6C9">&nbsp;</TD>
        <TD BGCOLOR="FFFFFF" ALIGN="CENTER"><?php
GMapDrawScaleBar() ?></TD>
        <TD BGCOLOR="#87C6C9">&nbsp;</TD>
    </TR>
    <TR>
        <TD><IMG SRC="images/corner_BL_DB.jpg" WIDTH="4"
HEIGHT="4"></TD>
        <TD BGCOLOR="#87C6C9"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
        <TD><IMG SRC="images/corner_BR_DB.jpg" WIDTH="4"
HEIGHT="4"></TD>
    </TR>
</TABLE>

</TD>
<TD VALIGN="TOP">
    <TABLE BORDER="0" CELLSPACING="0" CELLPADDING="0">
        <TR>
            <TD><IMG SRC="images/corner_TL_LB.jpg" WIDTH="4"
HEIGHT="4"></TD>
            <TD BGCOLOR="#E2EFF5"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
            <TD><IMG SRC="images/corner_TR_LB.jpg" WIDTH="4"
HEIGHT="4"></TD>
        </TR>
        <TR ALIGN="CENTER">
            <TD BGCOLOR="#E2EFF5">&nbsp;</TD>
    </TABLE>

```

```

        <TD BGCOLOR="#E2EFF5"><FONT FACE="Arial, Helvetica, sans-
serif" SIZE="2"><B>Plano de Referencia</B></FONT></TD>
        <TD BGCOLOR="#E2EFF5">&nbsp;</TD>
    </TR>
    <TR>
        <TD BGCOLOR="#87C6C9"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
        <TD BGCOLOR="#87C6C9"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
        <TD BGCOLOR="#87C6C9"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
    </TR>
    <TR>
        <TD BGCOLOR="#87C6C9">&nbsp;</TD>
        <TD BGCOLOR="#EEEEEE"><?php GMapDrawKeyMap() ?></TD>
        <TD BGCOLOR="#87C6C9">&nbsp;</TD>
    </TR>
    <TR>
        <TD><IMG SRC="images/corner_BL_DB.jpg" WIDTH="4"
HEIGHT="4"></TD>
        <TD BGCOLOR="#87C6C9"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
        <TD><IMG SRC="images/corner_BR_DB.jpg" WIDTH="4"
HEIGHT="4"></TD>
    </TR>
</TABLE>

```

```

<FONT FACE="Arial, Helvetica, sans-serif" SIZE="2"> <BR>
<SELECT NAME="MapSize" onchange="MapSizeSelected()">

```

```

    <OPTION>Resolucion del mapa</OPTION>
    <OPTION>-----</OPTION>
    <OPTION VALUE="350,300"
        <?php if (($gpoMap->width == 350) && ($gpoMap->height == 300))
            echo "SELECTED";?>> 350 x 300</OPTION>
    <OPTION VALUE="600,450"
        <?php if (($gpoMap->width == 600) && ($gpoMap->height ==
450))
            echo "SELECTED";?>> 600 x 450</OPTION>
    <OPTION VALUE="800,600"
        <?php if (($gpoMap->width == 800) && ($gpoMap->height ==
600))
            echo "SELECTED";?>> 800 x 600</OPTION>
</SELECT>
<br><br>

```

```

<TABLE BORDER="0" CELLSPACING="0" CELLPADDING="0">
<tr>
  <TD bgcolor=#ffffff>
    <INPUT TYPE="radio" NAME="MAP_NAME"
VALUE="prosiguis1.map"
    <?php if (strval($_HTTP_FORM_VARS["MAP_NAME"]) ==
"prosiguis1.map") echo "CHECKED"; ?> >
    Observar piso 1.</TD>
</tr>

<tr>
  <TD bgcolor=#ffffff>
    <INPUT TYPE="radio" NAME="MAP_NAME"
VALUE="prosiguis2.map"
    <?php if (strval($_HTTP_FORM_VARS["MAP_NAME"]) ==
"prosiguis2.map") echo "CHECKED"; ?> >
    Observar piso 2.</TD>
</tr>

<tr>
  <TD bgcolor=#ffffff>
    <INPUT TYPE="radio" NAME="MAP_NAME"
VALUE="prosiguis3.map"
    <?php if (strval($_HTTP_FORM_VARS["MAP_NAME"]) ==
"prosiguis3.map") echo "CHECKED"; ?> >
    Observar piso 3.</TD>
</tr>

<tr>
  <TD bgcolor=#ffffff align="center"> <INPUT TYPE="submit"
VALUE="Observar" </TD>
</tr>
</table>
</FONT><BR>

```

```

<?php if (!isHtmlMode()) echo "<!--" ?>
<TABLE BORDER="0" CELLSPACING="0" CELLPADDING="0">
<TR>
  <TD>
    <INPUT TYPE="radio" NAME="CMD" VALUE="ZOOM_IN"
    <?php if ($gszCommand == "ZOOM_IN") echo "CHECKED"; ?> >
  </TD>
  <TD><IMG SRC="images/icon_zoomin.gif" WIDTH="25"
HEIGHT="25"></TD>
</TR>
<TR>

```

```

        <TD>
            <INPUT TYPE="radio" NAME="CMD" VALUE="ZOOM_OUT"
            <?php if ($gszCommand == "ZOOM_OUT") echo "CHECKED"; ?> >
        </TD>
    <TD><IMG SRC="images/icon_zoomout.gif" WIDTH="25"
HEIGHT="25"></TD>
</TR>
<TR>
    <TD>
        <INPUT TYPE="radio" NAME="CMD" VALUE="RECENTER"
        <?php if ($gszCommand == "RECENTER") echo "CHECKED"; ?> >
    </TD>
    <TD><IMG SRC="images/icon_recentre.gif" WIDTH="25"
HEIGHT="25"></TD>
</TR>
<TR>
    <TD>
        <INPUT TYPE="radio" NAME="CMD" VALUE="QUERY_POINT"
        <?php if ($gszCommand == "QUERY_POINT") echo "CHECKED";
?>>
    </TD>
    <TD><IMG SRC="images/icon_info.gif" WIDTH="25"
HEIGHT="25"></TD>
</TR>
</TABLE>
<?php if (!IsHtmlMode()) echo "-->" ?>
</TR>
<TR>
    <TD VALIGN="TOP" ALIGN="CENTER">
        <TABLE CELLPADDING="2" CELLSPACING="0">
            <TR>
                <TD><INPUT TYPE="image" BORDER="0" <?php if (IsHtmlMode())
printf("NAME=\"JavaOn\" SRC=\"images/java_on.gif\""); else
printf("NAME=\"JavaOff\" SRC=\"images/java_off.gif\"");?> WIDTH="24"
HEIGHT="25"></TD>
                <TD><FONT FACE="Arial, Helvetica, sans-serif" SIZE="2"><?php if
(IsHtmlMode()) printf("Click para activar Java"); else printf("Click para
desactivar Java");?></FONT>
                <INPUT TYPE=HIDDEN NAME=PREVIOUS_MODE VALUE=<?php
echo IsHtmlMode(); ?>></TD>
            </TR>
        </TABLE>
    </TD>
<TD ALIGN="CENTER">

```

```

<TABLE BORDER="0" CELLSPACING="0" CELLPADDING="0">
  <TR>
    <TD><IMG SRC="images/corner_TL_LB.jpg" WIDTH="4"
HEIGHT="4"></TD>
    <TD BGCOLOR="#E2EFF5"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
    <TD><IMG SRC="images/corner_TR_LB.jpg" WIDTH="4"
HEIGHT="4"></TD>
  </TR>
  <TR ALIGN="CENTER">
    <TD BGCOLOR="#E2EFF5">&nbsp;</TD>
    <TD BGCOLOR="#E2EFF5"><FONT FACE="Arial, Helvetica, sans-
serif" SIZE="2"><B>Informacion General</B></FONT></TD>
    <TD BGCOLOR="#E2EFF5">&nbsp;</TD>
  </TR>
  <TR>
    <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="1"></TD>
    <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="1"></TD>
    <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="1"></TD>
  </TR>
  <TR>
    <TD BGCOLOR="#87C6C9"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
    <TD BGCOLOR="#87C6C9"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
    <TD BGCOLOR="#87C6C9"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
  </TR>
  <TR>
    <TD BGCOLOR="#87C6C9">&nbsp;</TD>
    <TD WIDTH=<?php echo $gpoMap->width; ?><?php
GMapDumpQueryResults(); ?></TD>
    <TD BGCOLOR="#87C6C9">&nbsp;</TD>
  </TR>
  <TR>
    <TD><IMG SRC="images/corner_BL_DB.jpg" WIDTH="4"
HEIGHT="4"></TD>
    <TD BGCOLOR="#87C6C9"><IMG SRC="images/pixel.gif"
WIDTH="1" HEIGHT="4"></TD>
    <TD><IMG SRC="images/corner_BR_DB.jpg" WIDTH="4"
HEIGHT="4"></TD>
  </TR>

```

```
</TABLE>

</TD>
</TR>
<TR>
  <TD COLSPAN="3">&nbsp;</TD>
</TR>
<TR>
  <TD
      COLSPAN="3"
      BGCOLOR="#E2EFF5"><IMG
SRC="images/pixel.gif" WIDTH="1" HEIGHT="2"></TD>
</TR>
<TR>
  <TD COLSPAN="3">&nbsp;</TD>
</TR>
<TR>
  <TD COLSPAN="3">
  </TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>
```

ARCHIVO DE EXTENSION .PHP3

Los archivos de extensión **.php3** contienen una combinación de código **php/mapsript** y código **java**, los cuales al ser llamados o incluidos por los archivos **.phtml** realizan la gran mayoría de tareas requeridas para la visualización de datos.

A continuación se presenta el archivo **.php3** utilizado en la aplicación:

```

<HTML>
<?php

//$HTTP_FORM_VARS contiene las variables GET o POST.
if (sizeof($HTTP_POST_VARS) > 0)
    $HTTP_FORM_VARS = $HTTP_POST_VARS;
else if (sizeof($HTTP_GET_VARS) > 0)
    $HTTP_FORM_VARS = $HTTP_GET_VARS;
else
    $HTTP_FORM_VARS = array("");

//Revisar que tipo de soporte de imagen está soportado.
if (strpos( ms_GetVersion(), "OUTPUT=GIF" ) > 0 )
{
    $gAppletImgFmt = MS_GIF;
    $gImagesFmt = MS_GIF;
}
else
{
    $gAppletImgFmt = MS_JPEG;
    $gImagesFmt = MS_PNG;
}

//Función que permite establecer el status on/off de las capas de acuerdo a
parámetros HTTP dados.
function GMap75CheckClick()
{
    GLOBAL    $HTTP_FORM_VARS;
    GLOBAL    $gpoMap, $gbShowQueryResults, $gszZoomBoxExt;
    GLOBAL    $dfMapExtMinX;
    GLOBAL    $dfMapExtMinY;
    GLOBAL    $dfMapExtMaxX;
    GLOBAL    $dfMapExtMaxY;

    GLOBAL    $dfMaxExtMinX;
    GLOBAL    $dfMaxExtMinY;
    GLOBAL    $dfMaxExtMaxX;
    GLOBAL    $dfMaxExtMaxY;

    reset( $HTTP_FORM_VARS );
//Busca las capas prendidas o apagadas
if (sizeof($HTTP_FORM_VARS) >= 2)
{
    $poLayer = $gpoMap->getLayerByName("punto especial");
}
}

```

```

if ($HTTP_FORM_VARS["punto_especial"])
    $poLayer->set("status", 1);
else
    $poLayer->set("status", 0);

$poLayer = $gpoMap->getLayerByName("punto electrico");
if ($HTTP_FORM_VARS["punto_electrico"])
    $poLayer->set("status", 1);
else
    $poLayer->set("status", 0);

    $poLayer = $gpoMap->getLayerByName("punto hidraulico");
if ($HTTP_FORM_VARS["punto_hidraulico"])
    $poLayer->set("status", 1);
else
    $poLayer->set("status", 0);

    $poLayer = $gpoMap->getLayerByName("punto sanitario");
if ($HTTP_FORM_VARS["punto_sanitario"])
    $poLayer->set("status", 1);
else
    $poLayer->set("status", 0);

    $poLayer = $gpoMap->getLayerByName("division_tabique_estructura");
if ($HTTP_FORM_VARS["division_tabique_estructura"])
    $poLayer->set("status", 1);
else
    $poLayer->set("status", 0);

    $poLayer = $gpoMap->getLayerByName("area");
if ($HTTP_FORM_VARS["area"])
    $poLayer->set("status", 1);
else
    $poLayer->set("status", 0);

    $poLayer = $gpoMap->getLayerByName("piso");
if ($HTTP_FORM_VARS["piso"])
    $poLayer->set("status", 1);
else
    $poLayer->set("status", 0);

    $poLayer = $gpoMap->getLayerByName("edificio");
if ($HTTP_FORM_VARS["edificio"])
    $poLayer->set("status", 1);

```

```

else
    $poLayer->set("status", 0);

    $poLayer = $gpoMap->getLayerByName("lote");
if ($HTTP_FORM_VARS["lote"])
    $poLayer->set("status", 1);
else
    $poLayer->set("status", 0);

$poLayer = $gpoMap->getLayerByName("sede");
if ($HTTP_FORM_VARS["sede"])
    $poLayer->set("status", 1);
else
    $poLayer->set("status", 0);

$poLayer = $gpoMap->getLayerByName("posicion luminaria");
if ($HTTP_FORM_VARS["posicion_luminaria"])
    $poLayer->set("status", 1);
else
    $poLayer->set("status", 0);

$poLayer = $gpoMap->getLayerByName("puerta");
if ($HTTP_FORM_VARS["puerta"])
    $poLayer->set("status", 1);
else
    $poLayer->set("status", 0);

$poLayer = $gpoMap->getLayerByName("escalones");
if ($HTTP_FORM_VARS["escalones"])
    $poLayer->set("status", 1);
else
    $poLayer->set("status", 0);
}
// Los extents de lo que en realidad se ve, generados con GMapRedraw()
if ($HTTP_FORM_VARS["minx"])
{
    $dfMinX = doubleval($HTTP_FORM_VARS["minx"]);
    $dfMinY = doubleval($HTTP_FORM_VARS["miny"]);
    $dfMaxX = doubleval($HTTP_FORM_VARS["maxx"]);
    $dfMaxY = doubleval($HTTP_FORM_VARS["maxy"]);
}
else
{
    $dfMinX = $gpoMap->extent->minx;
    $dfMinY = $gpoMap->extent->miny;
}

```

```

    $dfMaxX = $gpoMap->extent->maxx;
    $dfMaxY = $gpoMap->extent->maxy;
}

// Busca cambios en la altura y ancho de la imagen principal.
if ($HTTP_FORM_VARS["MapSize"])
{
    if (ereg("([0-9]+),([0-9]+)",
        $HTTP_FORM_VARS["MapSize"], $MapSizeExtents))
    {
        $dfWidthPix = intval($MapSizeExtents[1]);
        $dfHeightPix = intval($MapSizeExtents[2]);

        $gpoMap->set("width",$dfWidthPix);
        $gpoMap->set("height",$dfHeightPix);

        SetMapExtents($dfMinX, $dfMinY, $dfMaxX, $dfMaxY);
    }
}

// Revisar si se ha hecho click sobre el mapa
if ($HTTP_FORM_VARS["KEYMAP_x"])
{
    // Inicia los extents georeferenciados del mapa principal
    $dfKeyMapXMin = $gpoMap->reference->extent->minx;
    $dfKeyMapYMin = $gpoMap->reference->extent->miny;
    $dfKeyMapXMax = $gpoMap->reference->extent->maxx;
    $dfKeyMapYMax = $gpoMap->reference->extent->maxy;

    $nClickPixX = intval($HTTP_FORM_VARS[KEYMAP_x]);
    $nClickPixY = intval($HTTP_FORM_VARS[KEYMAP_y]);

    $dfWidthPix = doubleval($HTTP_FORM_VARS[KEYMAPXSIZE]);
    $dfHeightPix = doubleval($HTTP_FORM_VARS[KEYMAPYSIZE]);

    $nClickGeoX = GMapPix2Geo($nClickPixX, 0, $dfWidthPix,
$dfKeyMapXMin,
    $dfKeyMapXMax, 0);
    $nClickGeoY = GMapPix2Geo($nClickPixY, 0, $dfHeightPix,
$dfKeyMapYMin,
    $dfKeyMapYMax, 1);

    $dfDeltaX = $dfMaxX - $dfMinX;
    $dfDeltaY = $dfMaxY - $dfMinY;
    $dfMiddleX = $nClickGeoX;

```

```

    $dfMiddleY = $nClickGeoY;

    $dfNewMinX = $dfMiddleX - ($dfDeltaX/2);
    $dfNewMinY = $dfMiddleY - ($dfDeltaY/2);
    $dfNewMaxX = $dfMiddleX + ($dfDeltaX/2);
    $dfNewMaxY = $dfMiddleY + ($dfDeltaY/2);

// Sin salirse de los bordes.
    if ($dfNewMinX < $dfMaxExtMinX)
    {
        $dfNewMinX = $dfMaxExtMinX;
        $dfNewMaxX = $dfNewMinX + ($dfDeltaX);
    }

    if ($dfNewMaxX > $dfMaxExtMaxX)
    {
        $dfNewMaxX = $dfMaxExtMaxX;
        $dfNewMinX = $dfNewMaxX - ($dfDeltaX);
    }
    if ($dfNewMinY < $dfMaxExtMinY)
    {
        $dfNewMinY = $dfMaxExtMinY;
        $dfNewMaxY = $dfNewMinY + ($dfDeltaY);
    }
    if ($dfNewMaxY > $dfMaxExtMaxY)
    {
        $dfNewMaxY = $dfMaxExtMaxY;
        $dfNewMinY = $dfNewMaxY - ($dfDeltaY);
    }

    SetMapExtents($dfNewMinX,      $dfNewMinY,      $dfNewMaxX,
    $dfNewMaxY);

}

// Ancho y alto actualmente seleccionado
if ($HTTP_FORM_VARS["imagewidth"])
{
    $dfWidthPix = intval($HTTP_FORM_VARS["imagewidth"]);
    $dfHeightPix = intval($HTTP_FORM_VARS["imageheight"]);
}
else
{
    $dfWidthPix = $gpoMap->width;
    $dfHeightPix = $gpoMap->height;
}

```

```

}

// Probar si el boton redibujar ha sido seleccionado, si lo ha sido, redibujar
con los mismo extents.
if ($HTTP_FORM_VARS["redraw_x"])
{
    SetMapExtents($dfMinX, $dfMinY, $dfMaxX, $dfMaxY);
}

// Probar si el zoom hacia afuera ha sido seleccionado.
else if ($HTTP_FORM_VARS["CMD"] == "ZOOM_OUT" &&
        !($HTTP_FORM_VARS["mainmap_x"]
$HTTP_FORM_VARS["INPUT_COORD"])
        {
            $oPixelPos = ms_newpointobj();
            $oGeorefExt = ms_newrectobj();
            $oGeorefMaxExt = ms_newrectobj();

            $oPixelPos->setxy($dfWidthPix/2.0, $dfHeightPix/2.0);
            $oGeorefExt->setextent($dfMinX, $dfMinY, $dfMaxX, $dfMaxY);
            $oGeorefMaxExt->setextent($dfMaxExtMinX, $dfMaxExtMinY,
                $dfMaxExtMaxX, $dfMaxExtMaxY);

            if (!$gpoMap->zoompoint(-2, $oPixelPos, $dfWidthPix, $dfHeightPix,
                $oGeorefExt, $oGeorefMaxExt))
            {
                $gpoMap->setExtent($dfMapExtMinX, $dfMapExtMinY,
                    $dfMapExtMaxX, $dfMapExtMaxY);
            }
        }
// Buscar zoom/consultar a través del applet o clic en la imagen.
else
{
    $dfDeltaX = $dfMaxX - $dfMinX;
    $dfDeltaY = $dfMaxY - $dfMinY;

// Extraer posición del clic y convertirla a coordenadas georeferenciadas.
if ($HTTP_FORM_VARS["mainmap_x"]
$HTTP_FORM_VARS["INPUT_TYPE"])
{
    $bRectangleInput = 0;
    if ($HTTP_FORM_VARS["INPUT_TYPE"])
    {
        $szCoord = strval($HTTP_FORM_VARS["INPUT_COORD"]);
        $szCoordType = strval($HTTP_FORM_VARS["INPUT_TYPE"]);
    }
}
}

```

```

        if (strcmp($szCoordType,"auto_point") == 0)
        {
            $szCoordArray =explode(",", $szCoord);
            $nClickPixX = $szCoordArray[0];
            $nClickPixY = $szCoordArray[1];
        }
// El applet Rosa se utiliza y el rectangulo de entrada en sintaxis (X1, Y1;X2,
Y2)
// Si el par de puntos es igual, tratar como un click puntual.
        else if (strcmp($szCoordType,"auto_rect") == 0)
        {
            $bRectangleInput = 1;

            $szFirstSetXY = strtok($szCoord, ";");
            $szSecondSetXY = strtok("\n");

            $szFirstCoord = explode(",", $szFirstSetXY);
            $szSecondCoord = explode(",", $szSecondSetXY);

            $oPixelRect = ms_newrectobj();

            $oPixelRect->setextent($szFirstCoord[0], $szFirstCoord[1],
                                $szSecondCoord[0],
                                $szSecondCoord[1]);

            if ($oPixelRect->minx == $oPixelRect->maxx &&
                $oPixelRect->miny == $oPixelRect->maxy)
            {
                $bRectangleInput = 0;
                $nClickPixX = $oPixelRect->minx;
                $nClickPixY = $oPixelRect->miny;
            }
            else
            {
                if ($oPixelRect->minx > $oPixelRect->maxx)
                {
                    // Usar *1 para evitar que $dfTmp se convierta en referencia
                    // para $oPixelRect->minx con PHP4.
                    $dfTmp = $oPixelRect->minx*1;
                    $oPixelRect->set("minx",$oPixelRect->maxx);
                    $oPixelRect->set("maxx",$dfTmp);
                }
                if ($oPixelRect->miny < $oPixelRect->maxy)
                {

```

```

        $dfTmp = $oPixelRect->miny*1;
        $oPixelRect->set("miny", $oPixelRect->maxy);
        $oPixelRect->set("maxy", $dfTmp);
    }
}

// El extent georeferenciado como comentario HTML
// Util para definir vistas.
$gszZoomBoxExt = sprintf("<!-- BOX= (%f, %f)-(%f, %f) -->",
    GMapPix2Geo($oPixelRect->minx, 0,
        $dfWidthPix, $dfMinX,
        $dfMaxX, 0),
    GMapPix2Geo($oPixelRect->miny, 0,
        $dfHeightPix, $dfMinY,
        $dfMaxY, 1),
    GMapPix2Geo($oPixelRect->maxx, 0,
        $dfWidthPix, $dfMinX,
        $dfMaxX, 0),
    GMapPix2Geo($oPixelRect->maxy, 0,
        $dfHeightPix, $dfMinY,
        $dfMaxY, 1) );
}
}
else
{
    $nClickPixX = intval($HTTP_FORM_VARS[mainmap_x]);
    $nClickPixY = intval($HTTP_FORM_VARS[mainmap_y]);
}

$oPixelPos = ms_newpointobj();
$oGeorefExt = ms_newrectobj();
$oGeorefMaxExt = ms_newrectobj();

$oPixelPos->setxy($nClickPixX, $nClickPixY);
$oGeorefExt->setextent($dfMinX, $dfMinY, $dfMaxX, $dfMaxY);
$oGeorefMaxExt->setextent($dfMaxExtMinX, $dfMaxExtMinY,
    $dfMaxExtMaxX, $dfMaxExtMaxY);

if ($HTTP_FORM_VARS["CMD"] == "ZOOM_IN")
{
    if ($bRectangleInput)
    {
        $gpoMap->zoomrectangle($oPixelRect, $dfWidthPix,
            $dfHeightPix, $oGeorefExt);
    }
}

```

```

else
    $gpoMap->zoompoint(2, $oPixelPos, $dfWidthPix,
                      $dfHeightPix, $oGeorefExt, $oGeorefMaxExt);
}
if ($HTTP_FORM_VARS["CMD"] == "ZOOM_OUT")
{
    if (!$gpoMap->zoompoint(-2, $oPixelPos, $dfWidthPix,
                          $dfHeightPix, $oGeorefExt,
                          $oGeorefMaxExt))
    {
        $gpoMap->setExtent($dfMapExtMinX, $dfMapExtMinY,
                          $dfMapExtMaxX, $dfMapExtMaxY);
    }
}
if ($HTTP_FORM_VARS["CMD"] == "RECENTER")
{
    $gpoMap->zoompoint(1, $oPixelPos, $dfWidthPix,
                      $dfHeightPix, $oGeorefExt,
                      $oGeorefMaxExt);
}
else if ($HTTP_FORM_VARS["CMD"] == "QUERY_POINT")
{
    // Consulte las capas seleccionadas en puntos o rectángulos.
    // DumpQueryResults() se usará para llamar los resultados.
    if ($bRectangleInput)
    {
        $oGeorefRect = ms_newrectobj();
        $oGeorefRect->set("minx",      GMapPix2Geo($oPixelRect-
>minx, 0,
                                                $dfWidthPix, $dfMinX,
                                                $dfMaxX, 0));
        $oGeorefRect->set("maxx",      GMapPix2Geo($oPixelRect-
>maxx, 0,
                                                $dfWidthPix, $dfMinX,
                                                $dfMaxX, 0));
        $oGeorefRect->set("miny",      GMapPix2Geo($oPixelRect-
>miny, 0,
                                                $dfHeightPix, $dfMinY,
                                                $dfMaxY, 1));
        $oGeorefRect->set("maxy",      GMapPix2Geo($oPixelRect-
>maxy, 0,
                                                $dfHeightPix, $dfMinY,
                                                $dfMaxY, 1));
        // Usar '@' para evitar avisos si la consulta no es hallada.
        @$gpoMap->queryByRect($oGeorefRect);
    }
}

```



```

$img = $gpoMap->draw();
$url = $img->saveWebImage($gAppletImgFmt, 0, 0, -1);

echo "\n".$gszZoomBoxExt."\n";
printf("<INPUT TYPE=HIDDEN NAME=minx VALUE=\\"%f\\"", $gpoMap->extent->minx);
printf("<INPUT TYPE=HIDDEN NAME=miny VALUE=\\"%f\\"", $gpoMap->extent->miny);
printf("<INPUT TYPE=HIDDEN NAME=maxx VALUE=\\"%f\\"", $gpoMap->extent->maxx);
printf("<INPUT TYPE=HIDDEN NAME=maxy VALUE=\\"%f\\"", $gpoMap->extent->maxy);

printf("<INPUT TYPE=HIDDEN NAME=imagewidth VALUE=\\"%d\\"", $gpoMap->width);
printf("<INPUT TYPE=HIDDEN NAME=imageheight VALUE=\\"%d\\"", $gpoMap->height);

// Usar command para actualizar el applet Rosa.
if ( strlen($gszCommand) == 0)
{
    $szButtonName = "zoomin";
}
else
{
    if ($gszCommand == "ZOOM_IN")
        $szButtonName = "zoomin";
// No deje Zoom hacia afuera presionado para evitar un loop infinito.
// if ($gszCommand == "ZOOM_OUT")
//     $szButtonName = "zoomout";
if ($gszCommand == "RECENTER")
    $szButtonName = "recentre";
if ($gszCommand == "QUERY_POINT")
    $szButtonName = "pquery";
}

if (!$gblsHtmlMode) // Usar el applet.
{
    printf("\n");
    printf("<APPLET NAME=\\"RosaApplet\\" ARCHIVE=\\"rosa/rosa.jar\\"
CODE=\\"Rosa2000\\" WIDTH=\\"%d\\" HEIGHT=\\"%d\\" MAYSCRIPT>\n",
$gpoMap->width, $gpoMap->height);
    printf("<PARAM NAME=\\"TB_POSITION\\" VALUE=\\"centre\\">\n");
    printf("<PARAM NAME=\\"TB_ALIGN\\" VALUE=\\"top\\">");
    printf("<PARAM NAME=\\"IMG_URL\\" VALUE=\\"%s\\"", $url);
}

```

```

        printf("<PARAM NAME=\"INP_FORM_NAME\" VALUE=\"myform\">");
        printf("<PARAM                                NAME=\"TB_BUTTONS\"
VALUE=\"zoomin|zoomout|recentre|pquery\">\n");
        printf("<PARAM                                NAME=\"INP_TYPE_NAME\"
VALUE=\"INPUT_TYPE\">\n");
        printf("<PARAM                                NAME=\"INP_COORD_NAME\"
VALUE=\"INPUT_COORD\">\n");

```

```

        printf("<PARAM                                NAME=\"TB_SELECTED_BUTTON\"
VALUE=\"%s\">",$szButtonName);

```

```

        printf("<PARAM                                NAME=\"TB_BUT_zoomin_IMG\"
VALUE=\"./images/tool_zoomin_1.gif\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_zoomin_IMG_PR\"
VALUE=\"./images/tool_zoomin_2.gif\">\n");
        printf("<PARAM NAME=\"TB_BUT_zoomin_HINT\" VALUE=\"Zoom
para acercar\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_zoomin_INPUT\"
VALUE=\"auto_rect\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_zoomin_NAME\"
VALUE=\"CMD\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_zoomin_VALUE\"
VALUE=\"ZOOM_IN\">\n");

```

```

        printf("<PARAM                                NAME=\"TB_BUT_zoomout_IMG\"
VALUE=\"./images/tool_zoomout_1.gif\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_zoomout_IMG_PR\"
VALUE=\"./images/tool_zoomout_2.gif\">\n");
        printf("<PARAM NAME=\"TB_BUT_zoomout_HINT\" VALUE=\"Zoom
para minimizar\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_zoomout_INPUT\"
VALUE=\"submit\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_zoomout_NAME\"
VALUE=\"CMD\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_zoomout_VALUE\"
VALUE=\"ZOOM_OUT\">\n");

```

```

        printf("<PARAM                                NAME=\"TB_BUT_recentre_IMG\"
VALUE=\"./images/tool_recentre_1.gif\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_recentre_IMG_PR\"
VALUE=\"./images/tool_recentre_2.gif\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_recentre_HINT\"
VALUE=\"Recentrar el mapa\">\n");

```

```

        printf("<PARAM                                NAME=\"TB_BUT_recentre_INPUT\"
VALUE=\"auto_point\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_recentre_NAME\"
VALUE=\"CMD\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_recentre_VALUE\"
VALUE=\"RECENTER\">\n");

        printf("<PARAM                                NAME=\"TB_BUT_pquery_IMG\"
VALUE=\"./images/tool_info_1.gif\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_pquery_IMG_PR\"
VALUE=\"./images/tool_info_2.gif\">\n");
        printf("<PARAM NAME=\"TB_BUT_pquery_HINT\" VALUE=\"Consultar
informacion sobre un elemento\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_pquery_INPUT\"
VALUE=\"auto_rect\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_pquery_NAME\"
VALUE=\"CMD\">\n");
        printf("<PARAM                                NAME=\"TB_BUT_pquery_VALUE\"
VALUE=\"QUERY_POINT\">\n");

        printf("</APPLET>");
        printf("<INPUT TYPE=\"HIDDEN\" NAME=\"CMD\" VALUE=\"\">");
        printf("<INPUT        TYPE=\"HIDDEN\"        NAME=\"INPUT_TYPE\"
VALUE=\"\">");
        printf("<INPUT        TYPE=\"HIDDEN\"        NAME=\"INPUT_COORD\"
VALUE=\"\">");
    }
    else
    {
        echo"<INPUT        TYPE=image        SRC=$url        BORDER=0
WIDTH=$gpoMap->width HEIGHT=$gpoMap->height NAME=mainmap>";
        printf("<INPUT TYPE=\"HIDDEN\" NAME=\"CMD\" VALUE=\"%s\">",
$gszCommand);
    }
}

// Funcion que permite dibujar el mapa de referencia.
function GMapDrawKeyMap()
{
    GLOBAL    $gpoMap;
    GLOBAL $gAppletImgFmt, $gImagesFmt;

    $img = $gpoMap->drawReferenceMap();
    $url = $img->saveWebImage($gImagesFmt, 0, 0, -1);
}

```

```

    printf("<INPUT TYPE=HIDDEN NAME=KEYMAPXSIZE VALUE=\"%d\">",
$img->width);
    printf("<INPUT TYPE=HIDDEN NAME=KEYMAPYSIZE VALUE=\"%d\">",
$img->height);

    echo"<INPUT TYPE=image SRC=$url BORDER=0 NAME=KEYMAP>";
}

// Funcion que permite dibujar la barra de escala.
function GMapDrawScaleBar()
{
    GLOBAL $gpoMap;
    GLOBAL $gAppletImgFmt, $gImagesFmt;

    $img = $gpoMap->drawScaleBar();
    $url = $img->saveWebImage($gImagesFmt, 0, 0, -1);

    echo"<IMG SRC=$url BORDER=0 >\n";
}

// Funcion que permite convertir una posicion de pixel a georeferenciada.
function GMapPix2Geo($nPixPos, $dfPixMin, $dfPixMax, $dfGeoMin,
$dfGeoMax,
    $nInversePix)
{
    $dfWidthGeo = $dfGeoMax - $dfGeoMin;
    $dfWidthPix = $dfPixMax - $dfPixMin;

    $dfPixToGeo = $dfWidthGeo / $dfWidthPix;

    if (!$nInversePix)
        $dfDeltaPix = $nPixPos - $dfPixMin;
    else
        $dfDeltaPix = $dfPixMax - $nPixPos;

    $dfDeltaGeo = $dfDeltaPix * $dfPixToGeo;

    $dfPosGeo = $dfGeoMin + $dfDeltaGeo;

    return ($dfPosGeo);
}

// Funcion que permite ponerle extents al mapa. Se prueban tambi n las
escalas max y min del .map

```

```

function SetMapExtents($dfNewMinX, $dfNewMinY, $dfNewMaxX,
$dfNewMaxY)
{
    GLOBAL $gpoMap;

    $gpoMap->setExtent($dfNewMinX, $dfNewMinY, $dfNewMaxX,
$dfNewMaxY);

    $dfScale = $gpoMap->scale;

    if ($dfScale < $gpoMap->web->minscale ||
        $dfScale > $gpoMap->web->maxscale)
        return false;

    return true;
}

// Funcion que produce una tabla con los resultados de la consulta.
// Si no hay resultados simplemente imprime "&nbsp;";
function GMapDumpQueryResults()
{
    GLOBAL $gpoMap, $gbShowQueryResults;
    GLOBAL $HTTP_FORM_VARS;

    if (! $gbShowQueryResults )
    {
        printf("&nbsp;");
        return;
    }

    // Conexion a la base de datos.
    $numResultsTotal = 0;
    $conn = @pg_connect("user=postgres dbname=sig");

    if ($conn) {
        $DB_CONECTADO = 1;
    }else {
        $DB_CONECTADO = 0;
    }

    for($iLayer=0; $iLayer < $gpoMap->numlayers; $iLayer++)
    {
        $oLayer = $gpoMap->GetLayer($iLayer);
        $numResults = $oLayer->getNumResults();

        if ($numResults == 0)

```



```

        $numResultsTotal++;
    }

    // Toma la llave primaria de la consulta y trae los datos de la base de
datos
    // basado en esa llave.
    if ($DB_CONECTADO and count($valores) > 0) {
    $textvalores = implode(",", $valores);
        $campos = implode(",", $selFields);
        $llave = $selFields[0];
        if (strval($llave) == "matricula_inmob") $sql = "select $campos
from $selTable where $llave in ('$textvalores')";
        else $sql = "select $campos from $selTable where $llave
in ($textvalores)";
        $result = @pg_exec($conn, $sql);
        // Imprime los resultados de la consulta en la BD.
        if ($result) {
            for ($ir = 0; $ir < pg_num_rows($result); $ir++) {
                $fila = @pg_fetch_row($result, $ir);
                print "<tr>";
                    for ($jr=0; $jr<count($fila); $jr++) {
                        print
BGCOLOR=#FFFFFF>". $fila[$jr] . "</td>";
                    }
                print "</tr>";
            }
        }else {
        // Comentario en HTML Util para depurar codigo si la consulta
falla.
        print "<tr><td colspan=" . count($selFields) . "><!--Error en la
consulta SQL" . $sql . "--></td></tr>";
        }
    }
    else {
        print "<tr><td colspan=" . count($selFields) . ">No es posible
conectar a la base de datos</td></tr>";
    }
    $oLayer->close();
    printf("</TABLE>\n");
}
if ($numResultsTotal == 0)
    echo "No existe respuesta.";
    pg_close($conn);
}
?>

```

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.