

CAPÍTULO 5

DESARROLLO DEL SISTEMA

5.1 IMPLEMENTACIÓN DE BASE DE DATOS

La implementación de la base de datos se realizó usando el manejador de Bases de datos Microsoft SQL Server 2008 como se mencionó en el capítulo 3.

Para llevar a cabo la implementación física de una base de datos definimos su nombre, su tamaño y los archivos y grupos de archivos usados para almacenarla.

Se usan tres tipos de archivos para almacenar una base de datos: archivos principales, archivos secundarios y registro de transacciones. Toda base de datos tiene al menos 2 archivos: un archivo de datos principal y un registro de transacciones como mínimo.^[6]

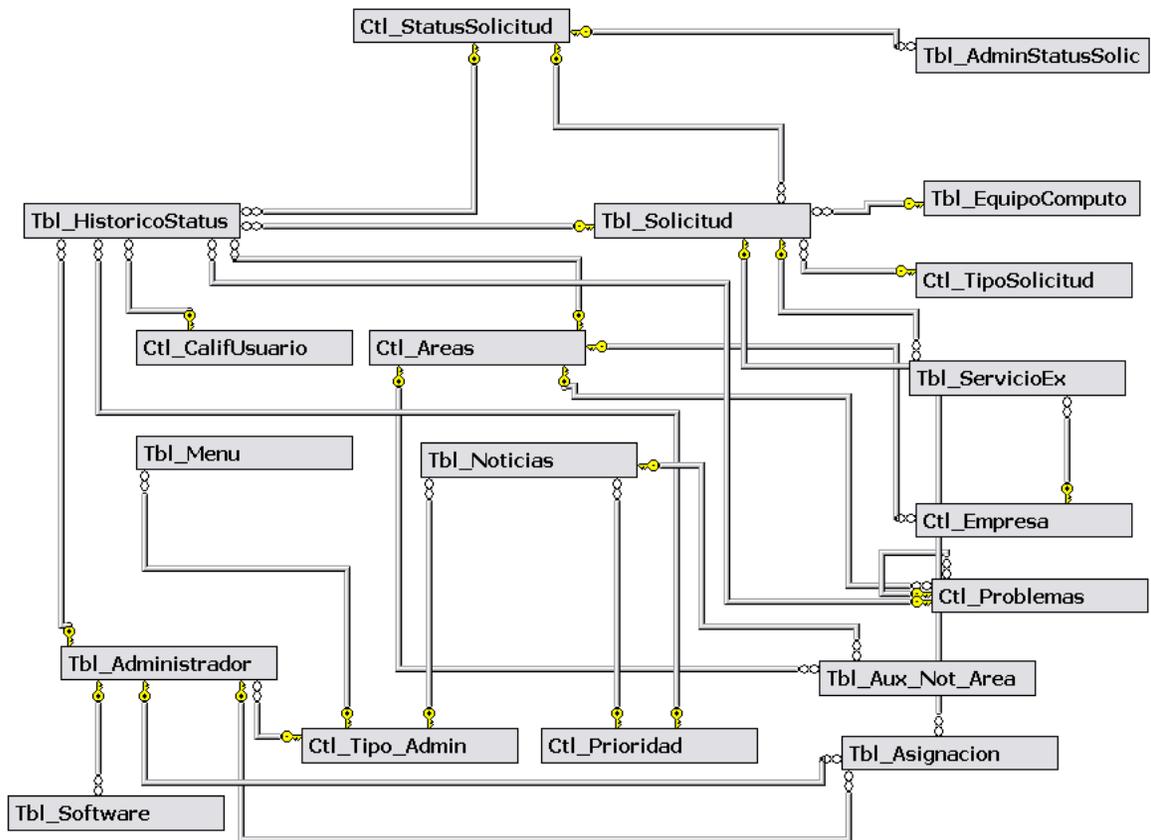
- Archivos principales: contienen la información de inicio de la base de datos. Se utilizan también para almacenar datos. Cada base de datos tiene un archivo principal.
- Archivos secundarios: contienen todos los datos que no caben en el archivo de datos principal, éstos son opcionales si el archivo principal es lo suficientemente grande como para contener todos los datos.
- Registro de transacciones: contienen la información de registro que se utiliza para recuperar la base de datos. Cada base de datos debe tener al menos un archivo de registro de transacciones, aunque puede tener más de uno.

Para crear la base de datos “CMP_SISOL” tomamos en cuenta lo siguiente:

- 1 Disposición de permisos para CRATE DATABASE, CREATE ANY DATABASE O ALTER ANY DATABASE.
- 2 En ciertos casos, algunos permisos se establecen en los archivos de datos y de registro de cada base de datos. Los permisos evitan que los archivos se modifiquen accidentalmente si residen en un directorio sin restricción de permisos.
- 3 Todos los objetos definidos por el usuario de la base de datos **model** se copian en todas las bases de datos recién creadas. Se puede agregar a la base de datos model todos los objetos (tablas, vistas, procedimientos almacenados, tipos de datos, etc.) que se desee incluir en todas las bases de datos nuevas.

SQL server provee métodos que se pueden utilizar para crear bases de datos: el comando Transact-SQL CREATE DATA BASE, el árbol de la consola Enterprise Manager, y el asistente Create Database, al cual se puede acceder a través de SQL Server Enterprise Manager.

Transact-SQL es el lenguaje que se utiliza para administrar instancias del motor de base de datos de SQL Server, para crear (CREATE) y administrar objetos de base de datos, y para insertar (INSERT INTO), recuperar (SELECT), modificar (ALTER) y eliminar (DELETE) datos. Transact-SQL es una extensión del lenguaje definido en los estándares de SQL publicados por International Standards Organization (ISO) y American National Standards Institute (ANSI). Lenguaje que se utilizó para la implementación de la base de datos en el Sistema de Solicitudes, tomando en cuenta la estructura de las tablas especificadas en el diccionario de datos^[6] (Ver Anexo 1).



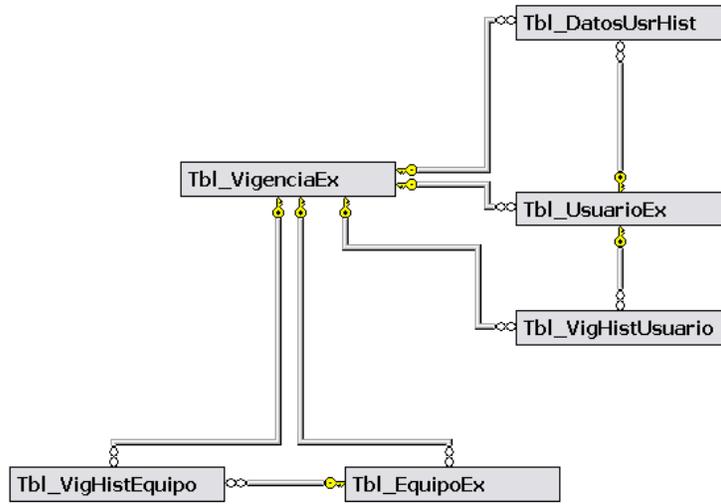
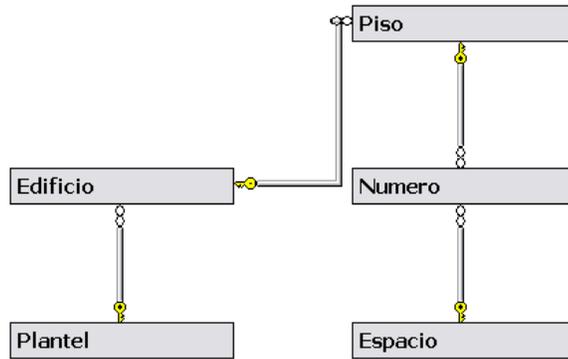


Figura 5.1 Diagrama de implementación en SQL Server 2008

5.2 PROGRAMACIÓN E INTERFAZ

Como se mencionó en los capítulos anteriores, la herramienta de desarrollo para este sistema es Microsoft Visual Studio 2008 y C#, el cual maneja una programación orientada a objetos, que se explica a continuación.

Un objeto es una estructura con datos y métodos que manipulan los datos. En general, una clase es una representación abstracta de algo, mientras que un objeto es un ejemplo utilizable de ese algo representado por la clase. La única excepción a esta regla la constituyen los miembros de clases compartidas, que se pueden usar tanto en caso de una variable de clase como de una variable de objeto declaradas como el tipo de la clase.

Las clases constan de campos, propiedades, métodos y eventos. Los campos y propiedades representan la información que contiene un objeto. Los campos son similares a las variables en cuanto que se pueden leer o definir directamente. Por ejemplo, si se tiene un objeto llamado "Automóvil", se puede almacenar su color en un campo llamado "Color".

Las propiedades se recuperan y definen igual que los campos, pero se implementan mediante los procedimientos de las propiedades Get y Set, que proporcionan más control sobre el modo en que los valores se definen o se devuelven.

Los métodos representan acciones que puede realizar un objeto. Por ejemplo, un objeto "Automóvil" puede tener definidos los métodos "MotorArranque," "Tracción" y "Parada". Los métodos se definen al agregar procedimientos (subrutinas o funciones) a la clase.

Los eventos son notificaciones que un objeto recibe de otros objetos u otras aplicaciones o que transmite a ellos. Los eventos permiten a los objetos realizar acciones cada vez que se da un caso concreto. Un ejemplo de evento para la clase "Automóvil" podría ser "Comprobar_motor".

Los campos, las propiedades, los métodos y los eventos forman sólo la mitad de la programación orientada a objetos, el resto requiere que los objetos admitan tres cualidades: encapsulación, herencia y polimorfismo.

La encapsulación implica el tratamiento de un grupo de propiedades, métodos y otros miembros como una única unidad u objeto. Los objetos pueden controlar el modo en que se cambian las propiedades y se ejecutan los métodos. Por ejemplo, un objeto puede validar valores antes de permitir cambios en propiedades. La encapsulación facilita también el cambio de implementación posterior al permitir ocultar los detalles de implementación de los objetos (ocultación de la información).

Herencia describe la capacidad de crear clases nuevas a partir de una clase existente. La nueva clase hereda todas las propiedades, métodos y eventos de la clase base, y se puede personalizar con propiedades y métodos adicionales. Por ejemplo, puede crear una nueva clase llamada "Camión" basada en la clase "Automóvil". La clase "Camión" hereda la propiedad "Color" de la clase "Automóvil" y puede tener propiedades adicionales, como "TracciónCuatroRuedas".

El polimorfismo implica la posibilidad de tener varias clases que se pueden usar de forma intercambiable, incluso si cada clase implementa las mismas propiedades o métodos de formas distintas. El polimorfismo es esencial para la programación orientada a objetos, ya que permite usar elementos con los mismos nombres, sin importar qué tipo de objeto esté en uso en ese momento. Por ejemplo, si se tiene una clase base "Automóvil", el polimorfismo permite al desarrollador definir distintos métodos "MotorArranque" para cualquier número de clases derivadas. El método

"MotorArranque" de una clase derivada llamada "AutomóvilDiesel" puede ser completamente distinto del método con el mismo nombre en la clase base.

Para la programación del sistema se utilizó una arquitectura de 3 capas que consta de capa de datos, capa de negocios y capa de presentación que permite su respuesta de una manera rápida y eficaz. Así mismo, permite futuras modificaciones de código para la ampliación del sistema en caso de ser necesario.

La figura 5.2 ilustra la arquitectura de programación en capas. Las Figuras 5.3, 5.4 y 5.5 muestran la implementación de las 3 capas, respectivamente.

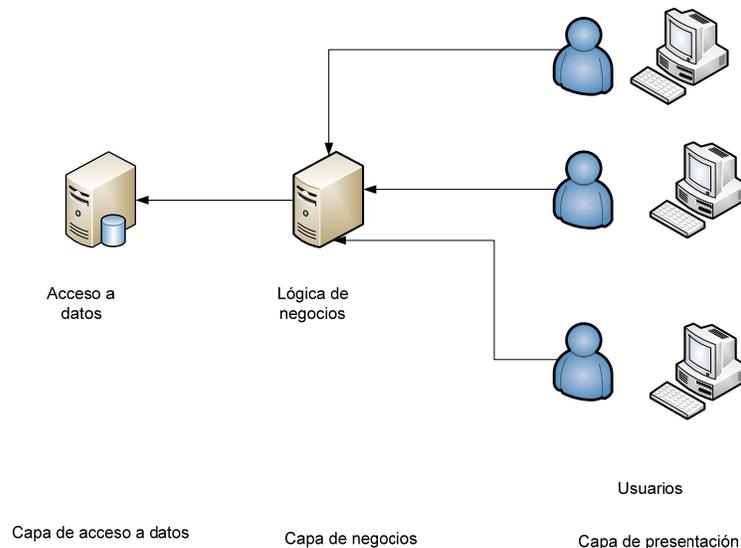


Figura 5.2 Programación en capas

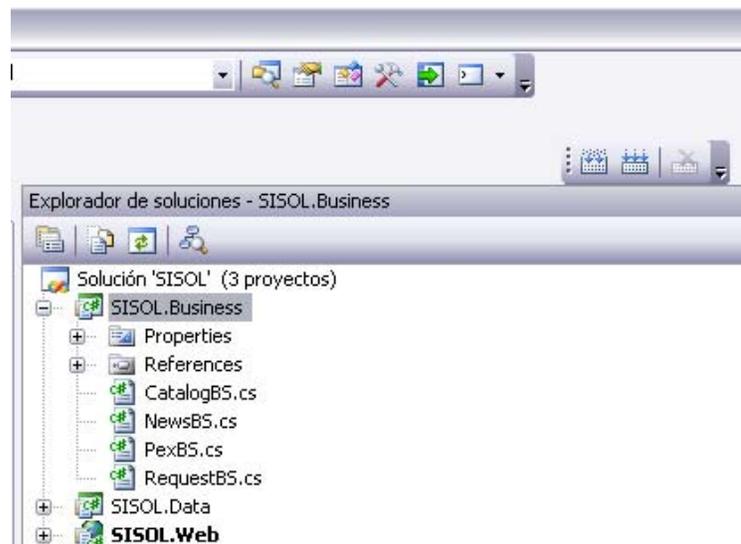


Figura 5.3 Implementación de la capa de Negocio

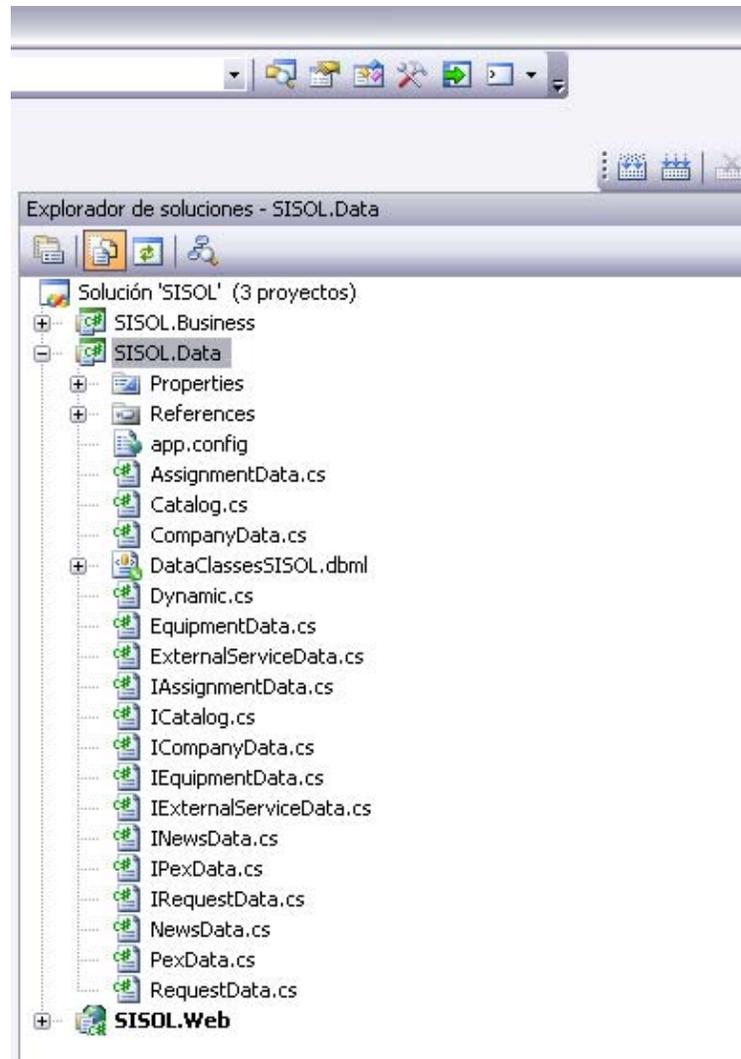


Figura 5.4 Implementación de la capa de Datos

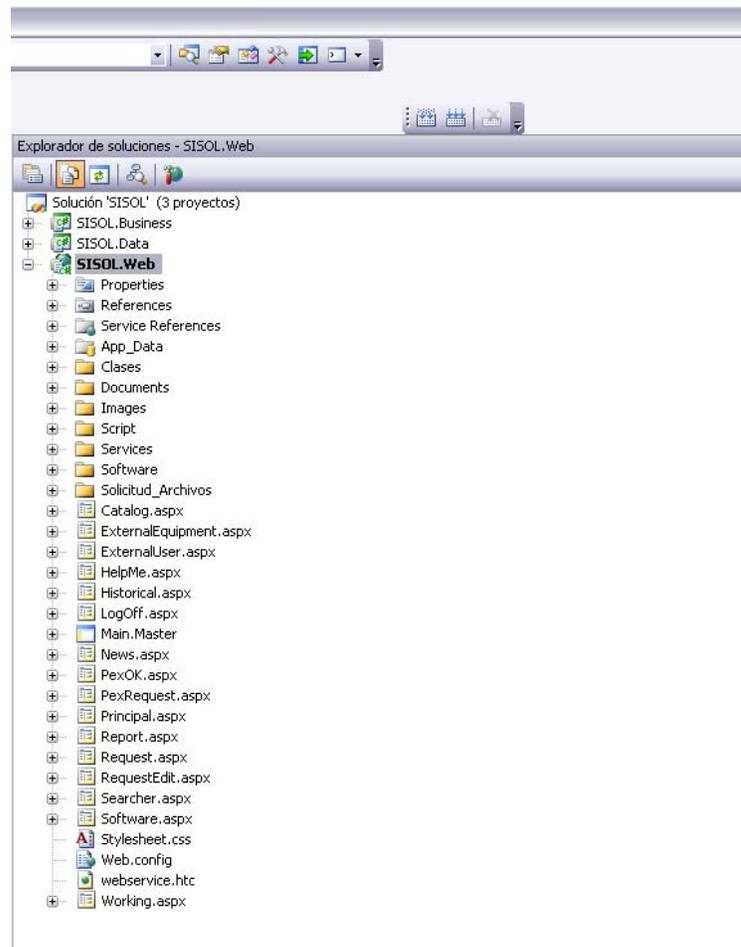


Figura 5.5 Implementación de la capa de Presentación

En la capa de presentación se diseñaron las páginas que integran el sistema, haciendo uso de páginas maestras o principales que proporcionan una funcionalidad para no repetir código, texto y elementos de control. En los siguientes puntos se muestran algunas pantallas principales implementadas en los módulos sistema.

5.2.1 Recepción, consulta y confirmación de solicitudes recibidas

- **Realizar solicitudes de servicio y solicitudes a nombre de otra persona**

En la siguiente pantalla (Figura 5.6) podemos observar la página principal para realizar una solicitud de servicio con la opción de poder realizarla a nombre de otra persona.



Figura 5.6 Realizar solicitud

- **Consulta de noticias o tips importantes para los usuarios y consulta de solicitudes realizadas**

En la siguiente pantalla (Figura 5.7) es posible consultar las solicitudes realizadas por el usuario que ingresó al sistema así como también consultar las noticias registradas y activas.



Figura 5.7 Consulta de noticias, tips y solicitudes

5.2.2 Administración

- **Estados de solicitudes, Prioridades**

En la siguiente pantalla (Figura 5.8) de edición de una solicitud es posible actualizar el estado y la prioridad de la misma.



Figura 5.8 Actualizar estado de solicitudes

- **Consulta de solicitudes**

En la siguiente pantalla (Figura 5.9) es posible consultar las solicitudes asignadas al administrador que ingresó al sistema.

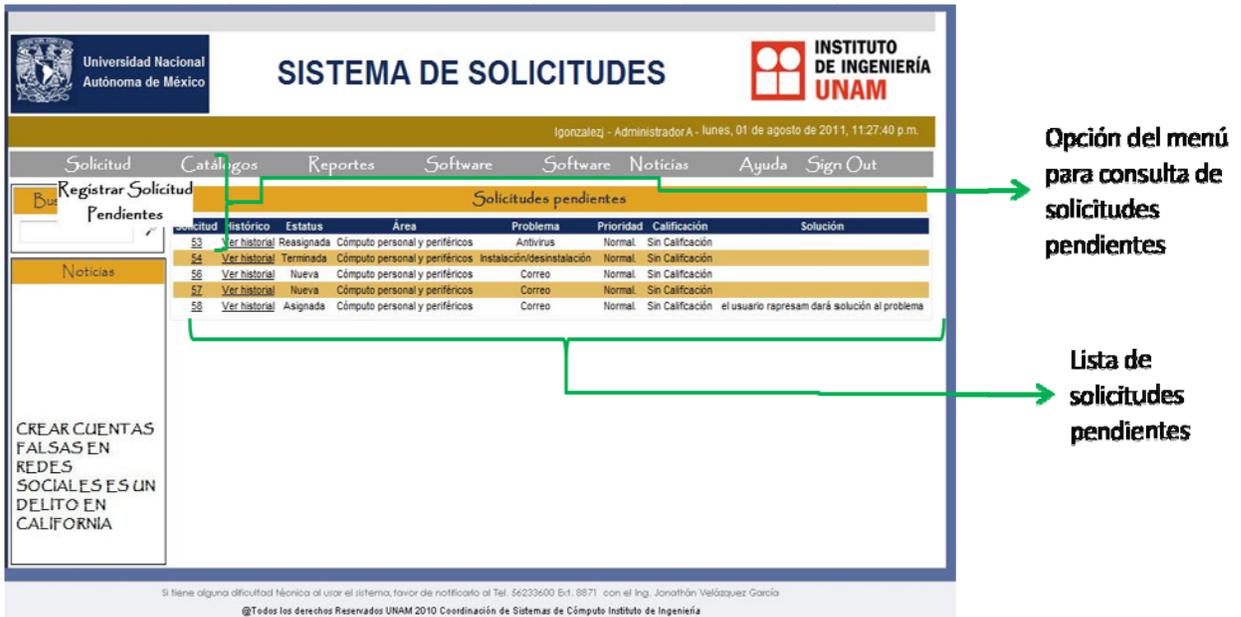


Figura 5.9 Consulta de solicitudes asignadas

- **Equipo y personal externo**

En la siguiente pantalla (Figura 5.10) es posible consultar las solicitudes de equipo y personal externo para aprobarlas o rechazarlas según sea el caso.

Si tiene alguna dificultad técnica al usar el sistema, favor de notificarlo al Tel. 56233600 Ext. 8871 con el Ing. Jonathan Velázquez García
 @Todos los derechos Reservados UNAM 2010 Coordinación de Sistemas de Cómputo Instituto de Ingeniería

Figura 5.10 Consulta de equipo y personal externo

- **Control de catálogos**

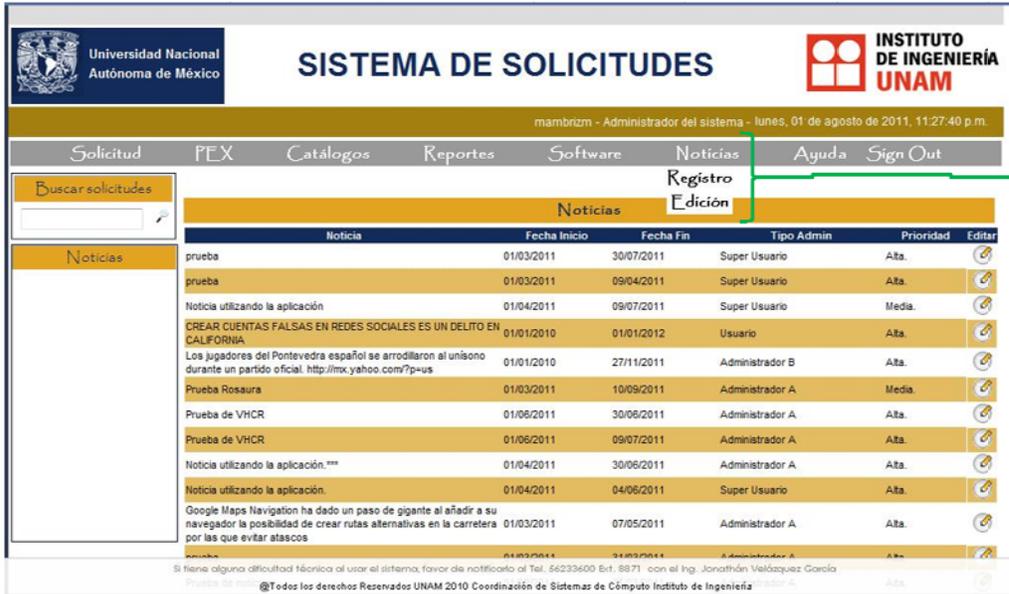
En la siguiente pantalla (Figura 5.11) es posible consultar, editar y agregar registros a los catálogos existentes en el sistema.

Si tiene alguna dificultad técnica al usar el sistema, favor de notificarlo al Tel. 56233600 Ext. 8871 con el Ing. Jonathan Velázquez García
 @Todos los derechos Reservados UNAM 2010 Coordinación de Sistemas de Cómputo Instituto de Ingeniería

Figura 5.11 Consulta de catálogos

- **Noticias/Tips**

En la siguiente pantalla (Figura 5.12) es posible consultar, editar y agregar registros de noticias que son mostradas al usuario al ingresar al sistema.



Opción de menú para administración de noticias.

Figura 5.12 Consulta y edición de noticias/tips

- **Software**

En la siguiente pantalla (Figura 5.13) es posible consultar, agregar y eliminar el software de apoyo al usuario.



Opción de menú para consulta y carga de software

Figura 5.13 Consulta y edición de software

5.2.3 Estadísticas y reportes

- **Reportes**

En la siguiente pantalla (Figura 5.14) se deben ingresar área y fechas para la generación del reporte de solicitudes atendidas en el periodo indicado al área seleccionada.

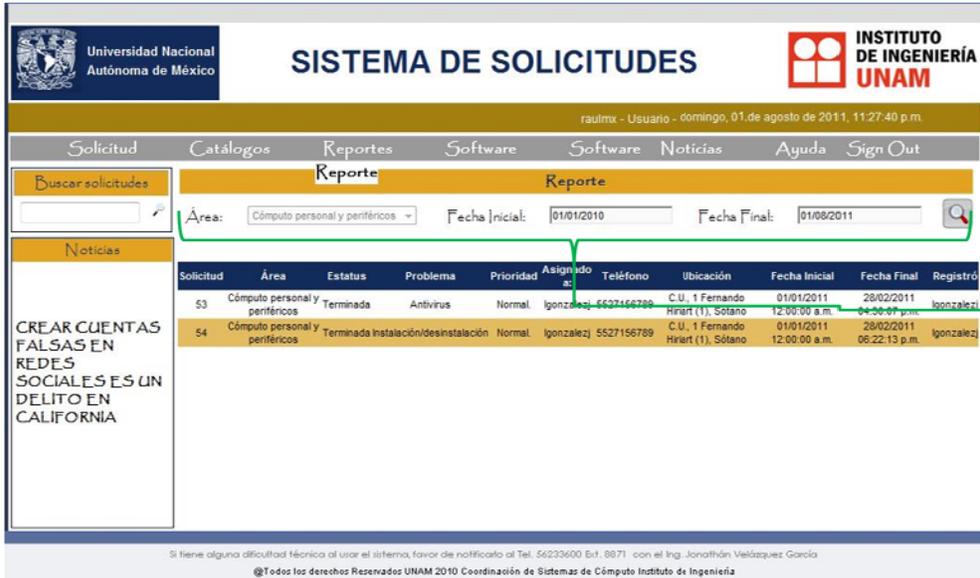


Figura 5.14 Generación de reportes

5.2.4 Registro de personal y equipo externo

- **Solicitud y control de equipo externo**

En la siguiente pantalla (Figura 5.15) es posible la captura de información para el registro de equipo externo.



Figura 5.15 Registro de equipo externo

- **Solicitud y control de personal externo**

En la siguiente pantalla (Figura 5.16) es posible la captura de información para el registro de usuario externo.

Universidad Nacional Autónoma de México

SIST

INSTITUTO DE INGENIERÍA UNAM

raulmx - Usuario - domingo, 01 de agosto de 2011, 11:27:40 p.m.

Solicitud PEX Ayuda Sign Out

Buscar solicitudes

Noticias

CREAR CUENTAS FALSAS EN REDES SOCIALES ES UN DELITO EN CALIFORNIA

Solicitud de Equipo tipo externo

Solicitud de Usuario

Académico Responsable: raulmx

Nombre:

Apellido Paterno:

Apellido Materno:

Periodo de vigencia

Fecha inicio:

Fecha fin:

Justificación Académico:

Capturar usuario externo

Si tiene alguna dificultad técnica al usar el sistema, favor de notificarlo al Tel. 56233600 Ext. 8971 con el Ing. Jonathan Velázquez García
© Todos los derechos Reservados UNAM 2010 Coordinación de Sistemas de Cómputo Instituto de Ingeniería

Figura 5.16 Registro de personal externo

