

CAPÍTULO 4

DISEÑO DEL SISTEMA

4.1 DISEÑO DE LA BASE DE DATOS

Conceptos básicos

Las Bases de Datos (BD) surgen por la necesidad de gestionar cada vez mayores cantidades de información, al referirnos a información nos referimos no sólo a los datos, sino también a las relaciones entre ellos y las restricciones que deben cumplir.

Una BD es una colección organizada de datos, relativa a un problema concreto, que puede ser compartida por un conjunto de usuarios/aplicaciones. ^[5]

DBA (del inglés Database Administrator, ‘Administrador de la Base de Datos’). Administra el uso de la BD y garantiza que funcione apropiadamente. Dentro de sus deberes se encuentra la recuperación de desastres (respaldos y pruebas de respaldos), análisis de desempeño y optimización, y asistencia en el diseño de la base de datos. ^[5]

Diseñador de Base de Datos. Diseña la estructura de la BD, llamado también arquitecto de BD. Actualmente, como las organizaciones se esfuerzan por optimizar sus recursos, el trabajo del diseñador de bases de datos se ha ampliado hasta abarcar más responsabilidades.

Modelo

Los modelos de datos son un grupo de herramientas conceptuales para describir los datos, sus relaciones, su semántica y sus limitantes, no son cosas físicas, son abstracciones que permiten la implementación de un sistema eficiente de BD, siendo este, el primer paso para su diseño; por lo general se refieren a algoritmos, y conceptos matemáticos.

La evolución de los sistemas de BD ha sido impulsada por la búsqueda de nuevas formas de modelar datos del mundo real. Haciendo un resumen de los modelos más reconocidos mostramos la figura 4.1

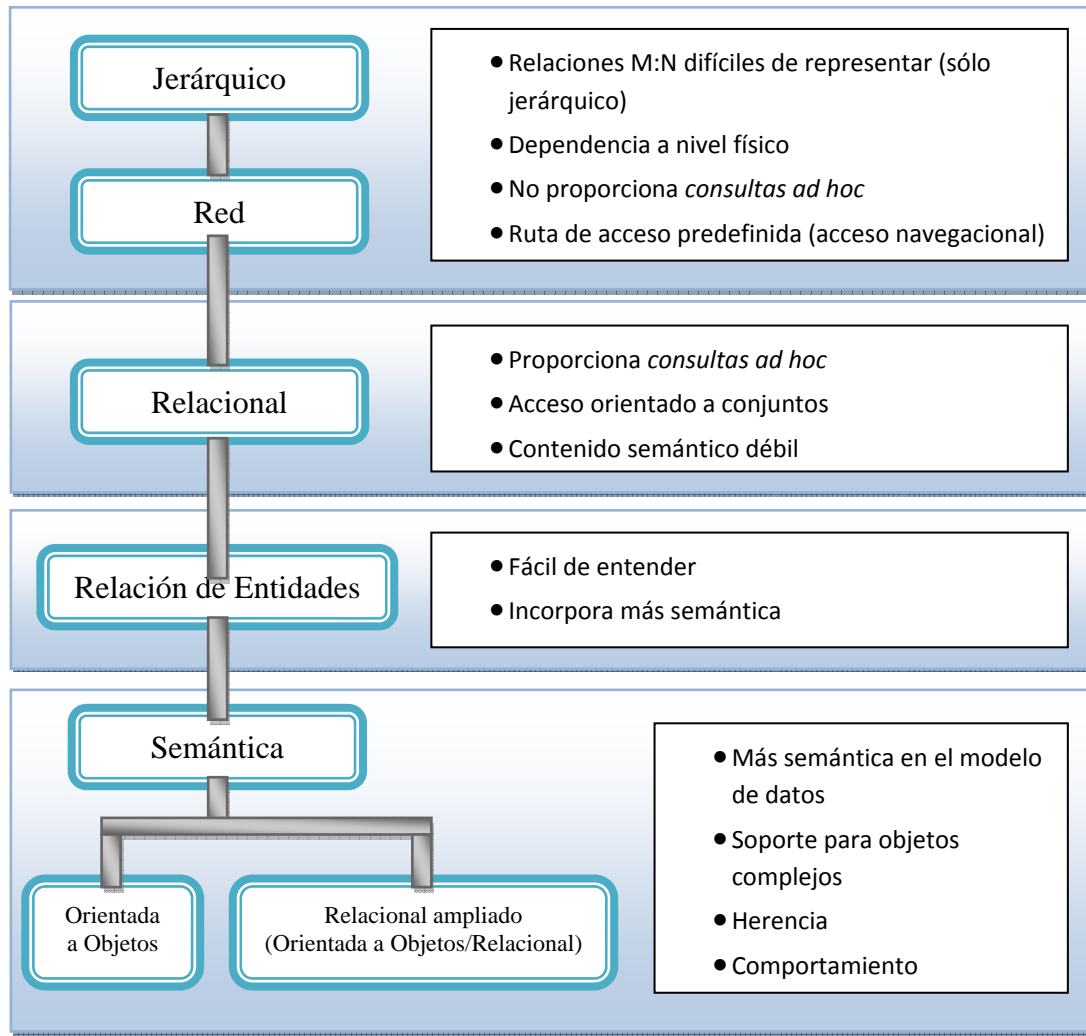


Figura 4.1 Diagrama de los modelos más reconocidos.

El modelo de red comparado con el jerárquico facilitó la representación de relaciones de muchos a muchos, el modelo relacional representa los datos de una forma más simple que los anteriores y un lenguaje de consulta relativamente fácil.

Analizando los modelos de datos, se pueden determinar algunas características que éstos deben tener para que sean ampliamente aceptados:

1. Un modelo de datos debe mostrar un cierto grado de simplicidad conceptual sin que se comprometa el contenido semántico* de la BD.
*La semántica tiene que ver con el comportamiento dinámico de los datos.
2. Un modelo de datos debe representar tanto como sea posible el mundo real.
3. La representación de las transformaciones en el mundo real (comportamiento) debe ajustarse a las características de consistencia e integridad de cualquier modelo de datos.

El modelo Entidad Relación (ER) se ha establecido como el principal modelo de datos para las aplicaciones de procesamiento de datos. Fue propuesto por Peter Chen en 1976 para la definición del esquema conceptual de una base de datos.

Términos del modelo ER:

Entidad

Representa un objeto con existencia física como: una persona, un animal, una casa, etc. (entidad concreta), o un objeto con existencia conceptual como: un puesto de trabajo, una asignatura de clases, un nombre, etc. (entidad abstracta).

Una entidad está descrita y se representa por sus características o atributos. Por ejemplo, la entidad Persona puede llevar consigo las características: Nombre, Apellido, Género, Estatura, Peso, Fecha de nacimiento, por nombrar algunas.

Hay dos tipos de entidades: fuertes y débiles. Una entidad débil es una entidad cuya existencia depende de la existencia de otra entidad. Una entidad fuerte es una entidad que no es débil.

Atributos

Los atributos son las propiedades que describen a cada entidad en un conjunto de entidades.

Un conjunto de entidades dentro de una entidad, tiene valores específicos asignados para cada uno de sus atributos, de esta forma, es posible su identificación unívoca. Dos o más entidades diferentes pueden tener los mismos valores para algunos de sus atributos, pero nunca para todos.

En particular, los atributos identificativos son aquellos que permiten diferenciar a una instancia de la entidad de otra distinta. Por ejemplo, el atributo identificativo que distingue a un alumno de otro es su número de cuenta.

Relación

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función.

Las entidades que están involucradas en una determinada relación se denominan entidades participantes. El número de participantes en una relación es lo que se denomina grado de la relación. Por lo tanto, una relación en la que participan dos entidades es una relación binaria; si son tres las entidades participantes, la relación es ternaria; etc.

Una relación recursiva es una relación donde la misma entidad participa más de una vez en la relación con distintos papeles. El nombre de estos papeles es importante para determinar la función de cada participación.

La cardinalidad con la que una entidad participa en una relación, especifica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad.

Dependencia de Existencia

Ésta implica que la existencia de una entidad depende de la existencia de otra. Si la existencia de la entidad a depende de la existencia de entidad b , se dice entonces que a es dependiente por existencia de b . Ésto implica que si se elimina b , también se debe eliminar a . Es entonces como podemos decir que a es una entidad subordinada y b es una entidad dominante. ^[13]

Llaves

Las entidades y las relaciones deben diferenciarse unas de otras en términos de sus atributos, para hacer la distinción se asigna una superllave a cada conjunto de entidades.

Superllave

Es un conjunto de uno o más atributos, que juntos permiten identificar en forma única a una entidad dentro del conjunto de entidades.

Llave Candidato

Es una superllave en la que ningunos de sus subconjuntos es una superllave de la relación. El atributo o conjunto de atributos K de la relación R si y sólo si satisface las siguientes propiedades:

- Unicidad: nunca hay dos entidades en la relación R con el mismo valor de K .
- Irreducibilidad: ningún subconjunto de K tiene la propiedad de unicidad, es decir, no se pueden eliminar componentes de K sin destruir la unicidad.

Llave Primaria

Es una llave candidato que el diseñador de la base de datos elije para identificar una entidad dentro de un conjunto de identidades. Es posible que una entidad no cuente con suficientes atributos para definir una llave primaria, a éstas también las podemos llamar entidades débiles, en consecuencia a las entidades con llave primaria son llamadas entidades fuertes.

Diseño

El diseño de las bases de datos se realiza por lo general en tres fases:

1. Diseño conceptual
Representación abstracta y de alto nivel de la realidad.
2. Diseño lógico
Convierte esta presentación en especificaciones que pueden implantarse en un sistema de cómputo.
3. Diseño físico
Determina las estructuras de almacenamiento físico y los métodos de consulta requeridos para un acceso eficaz a los contenidos de una base de datos.

En el diseño de una base de datos existen algunos puntos importantes a tomar en cuenta para lograr integridad y no redundancia de datos como son los siguientes:

- Correcta definición de objetos (Tablas y Catálogos)
- Establecer convenciones en el nombrado de la base de datos y sus objetos, estandarizarlo y apegarse a él.
- Utilizar el tipo de dato adecuado.
- Todas las tablas y catálogos deberán contener llaves primarias.
- Definir las relaciones entre tablas.
- Asegurarnos de normalizar los datos cuando menos hasta la tercer forma normal, así no comprometeremos el rendimiento.
- Definir aquellos tipos de datos cuya longitud y precisión se ajuste mejor al rango de valores requeridos validos.

Normalización

El objetivo del diseño de las bases de datos relacionales es la generación de un conjunto de esquemas relacionales que permite almacenar la información sin redundancias innecesarias, pero que también nos permita recuperar fácilmente esa información. Un enfoque es el diseño de esquemas que se hallen en una forma normal adecuada. ^[13]

La **primera forma normal (1FN)**, impone un requisito muy elemental a las relaciones; a diferencia de las demás formas normales, no exige información adicional como las dependencias funcionales.

Un dominio es atómico si se considera que los elementos de dominio son unidades indivisibles. Se dice que el esquema de una relación R está en la primera forma normal si los dominios de todos los atributos de R son atómicos.

Un conjunto de nombres es un ejemplo de valor no atómico. Por ejemplo, si el esquema de la relación *empleado* incluyera el atributo *hijos*, los elementos de cuyo dominio son conjuntos de nombres, el esquema no se hallaría en la 1FN.

Un esquema de relación R está en la **segunda forma normal (2FN)** si cada atributo A de R cumple uno de los criterios siguientes:

- Aparece en una clave candidata.
- No es parcialmente dependiente de una clave candidata.

Una dependencia funcional $x \rightarrow y$ se denomina dependencia parcial si hay un subconjunto adecuado z de x tal que $z \rightarrow y$. Se dice que y es parcialmente dependiente de x .

Un esquema de relación R está en **tercera forma normal (3FN)** respecto a un conjunto F de dependencias funcionales si, para todas las dependencias funcionales de F^+ de la forma $x \rightarrow y$, se cumple al menos una de las siguientes condiciones:

- $X \rightarrow y$ es una dependencia funcional trivial.
- X es una superclave de R .
- Cada atributo A de $y-x$ está contenido en alguna clave candidata de R .

Se dice que algunas dependencias funcionales son triviales porque las satisfacen todas las relaciones. Por ejemplo, $A \rightarrow A$ las satisfacen todas las relaciones que impliquen el atributo A .

Un esquema de relación R está en la **cuarta forma normal (4FN)** con respecto a un conjunto F de dependencias funcionales y multivaloradas si, para todas las dependencias multivaloradas de F^+ de la forma $x \twoheadrightarrow y$, se cumple, como mínimo, una de las condiciones siguientes

- $x \twoheadrightarrow y$ es una dependencia multivalorada trivial.
- X es una superclave del esquema R .

Las dependencias funcionales impiden que ciertas tuplas estén en una relación. Si $A \rightarrow B$, entonces no puede haber dos tuplas con el mismo valor de A y diferentes valores de B . Las dependencias multivaloradas, por otro lado, no impiden la existencia de esas tuplas. En lugar de eso, exigen que estén presentes en la relación otras tuplas de una cierta forma.

4.1.1 Modelo entidad relación

Con base en los conceptos anteriores, a continuación se muestran las entidades que conforman la base de datos que finalmente darán forma al modelo ER (Figura 4.n):

Ctl_Areas: Contiene las áreas correspondientes a la CSC

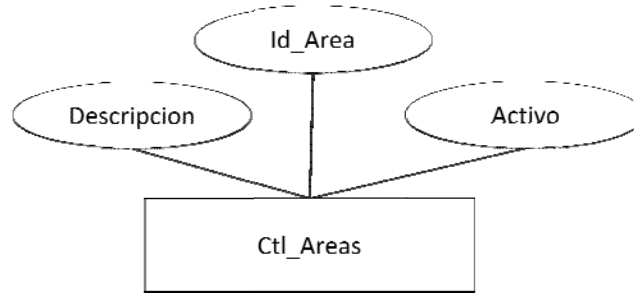


Figura 4.2 Entidad “Ctl_Areas”

Ctl_CalifUsuario: Contiene las calificaciones que el usuario puede asignar a la atención de su solicitud.

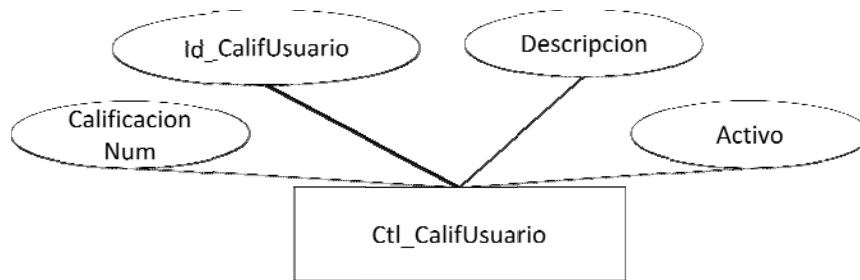


Figura 4.3 Entidad “Ctl_CalifUsuario”

Ctl_Empresa: Contiene las empresas externas que pueden realizar servicios de alguna área de la CSC.

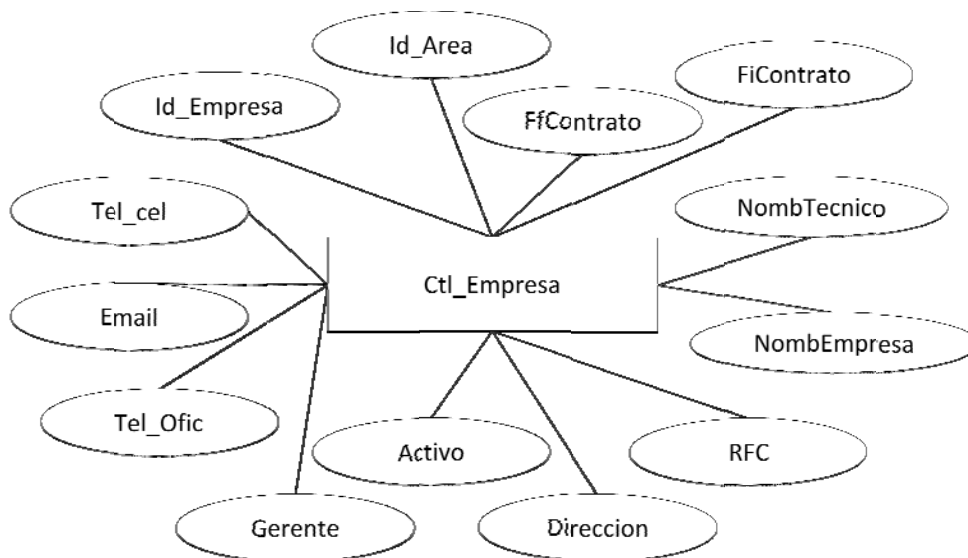


Figura 4.4 Entidad “Ctl_Empresa”

Ctl_Prioridad: Contiene las prioridades que se pueden asignar a una solicitud.

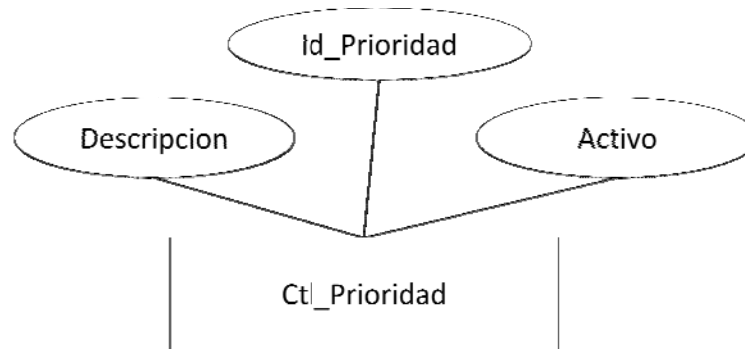


Figura 4.5 Entidad "Ctl_Prioridad"

Ctl_Problemas: Contiene los problemas correspondientes a las diferentes áreas de la CSC.

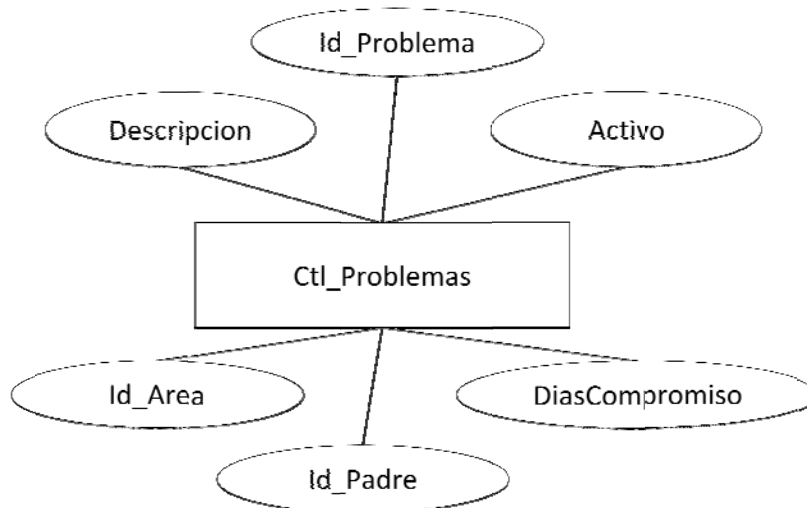


Figura 4.6 Entidad "Ctl_Prioridad"

Ctl_StatusSolicitud: Contiene los status que se pueden asignar a una solicitud.

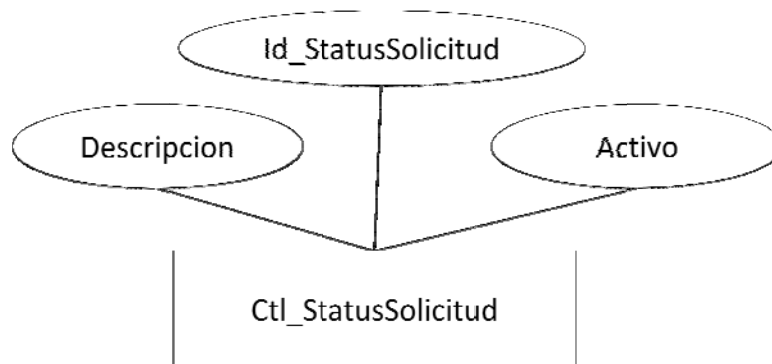


Figura 4.7 Entidad "Ctl_StatusSolicitud"

Ctl_Tipo_Admin: Contiene los tipos de administradores del sistema.

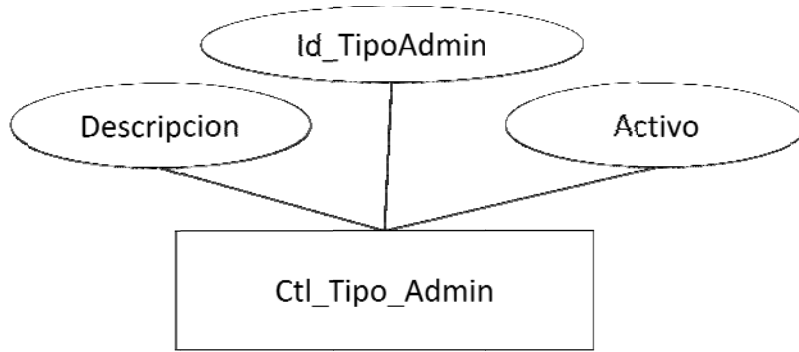


Figura 4.8 Entidad "Ctl_Tipo_Admin"

Ctl_TipoSolicitud: Contiene el tipo de solicitud que se puede realizar en el sistema.

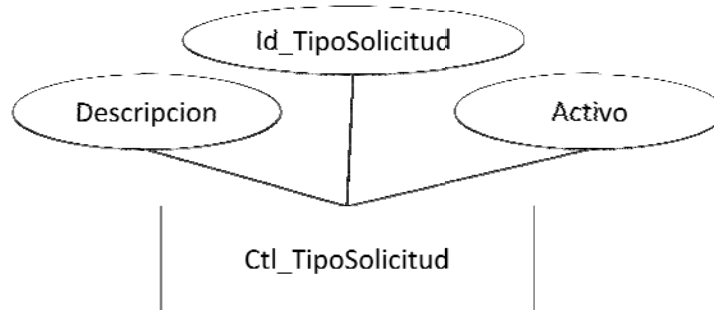


Figura 4.9 Entidad "Ctl_TipoSolicitud"

Edificio: Contiene los edificios existentes en el II.

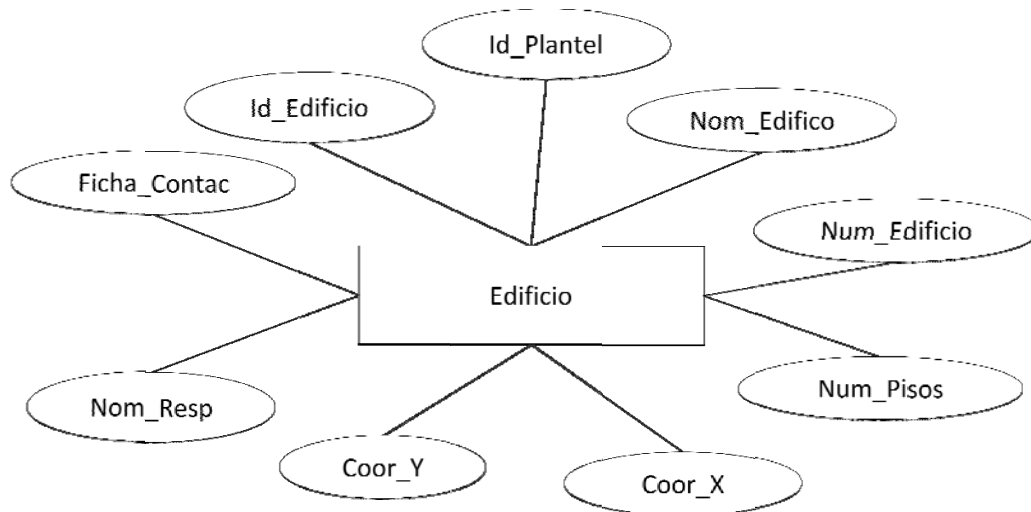


Figura 4.10 Entidad "Edificio"

Espacio: Contiene los espacios existentes en Ciudad Universitaria.

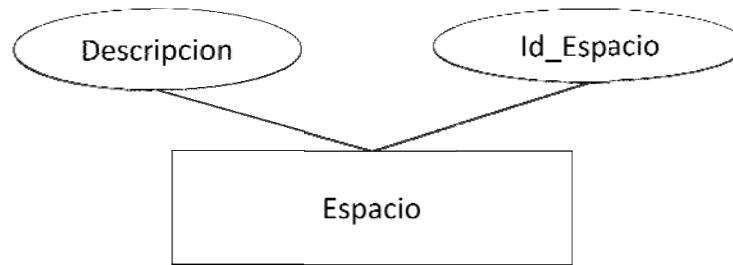


Figura 4.11 Entidad "Espacio"

Número: Contiene los números de los espacios.

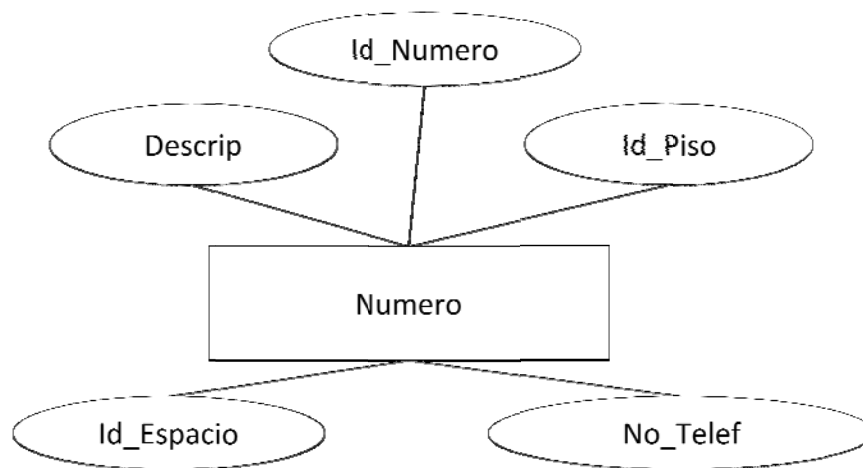


Figura 4.12 Entidad "Numero"

Piso: Contiene los pisos de los números de edificios.

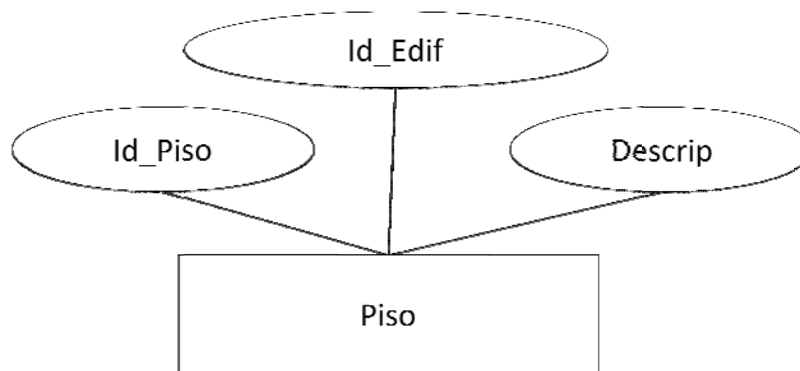


Figura 4.13 Entidad "Piso"

Plantel: Contiene los planteles existentes en Ciudad Universitaria.

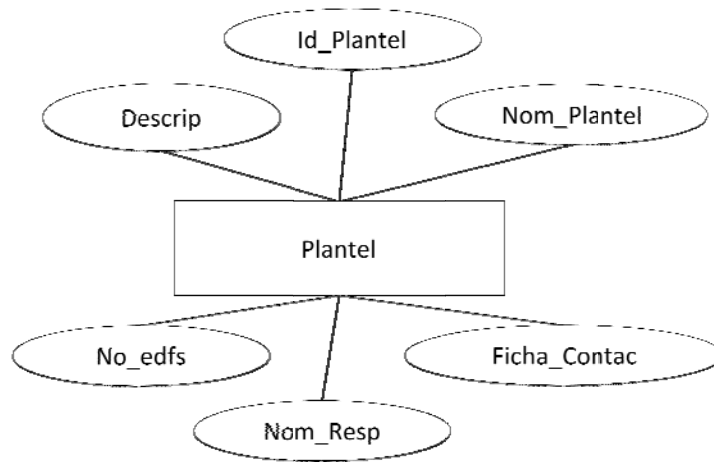


Figura 4.14 Entidad “Plantel”

Tbl_Administrador: Contiene los administradores del sistema.

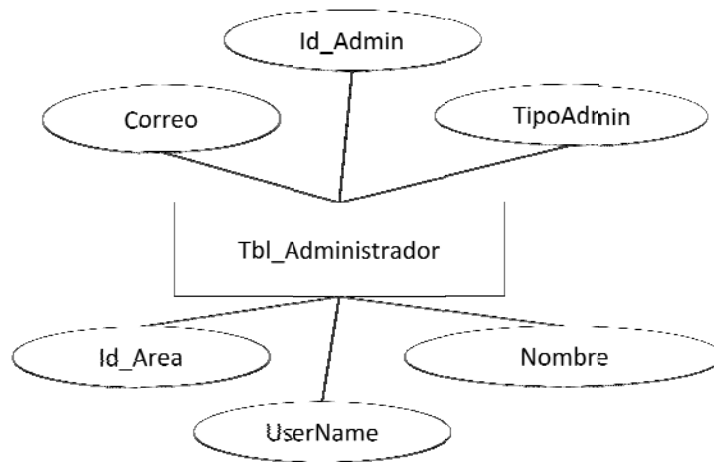


Figura 4.15 Entidad “Tbl_Administrador”

Tbl_AdminiStatusSolic: Contiene los status que los administradores pueden asignar a una solicitud.

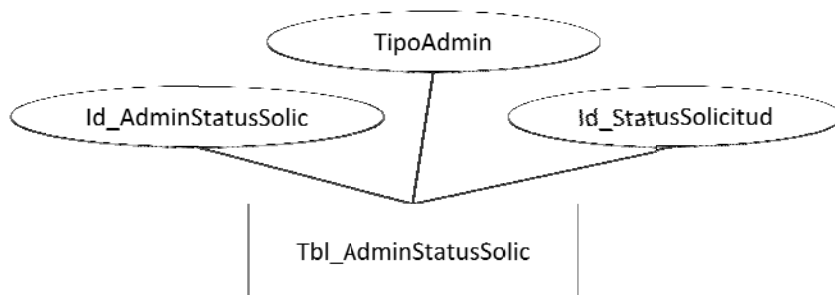


Figura 4.16 Entidad “Tbl_AdminiStatusSolic”

Tbl_Asignacion: Contiene los status que los administradores pueden asignar a una solicitud.

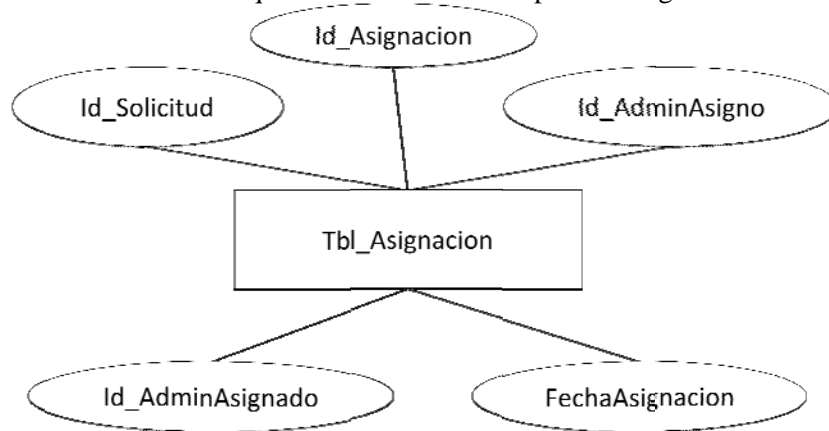


Figura 4.17 Entidad “Tbl_Asignacion”

Tbl_Aux_Not_Area: Relaciona las noticias por área.

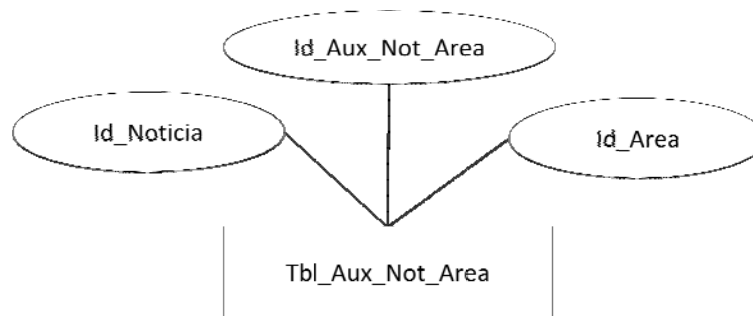


Figura 4.18 Entidad “Tbl_Aux_Not_Area”

Tbl_DatosUsrHist: Contiene los datos históricos de los usuarios.

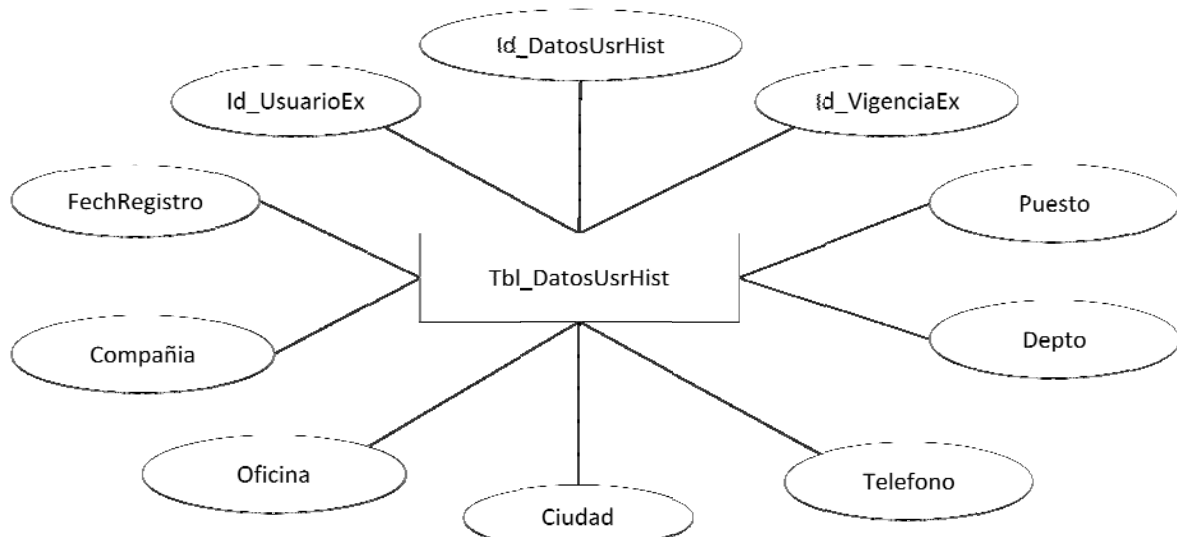


Figura 4.19 Entidad “Tbl_DatosUsrHist”

Tbl_EquipoComputo: Contiene los datos del equipo de computo registrado en el sistema de inventarios.

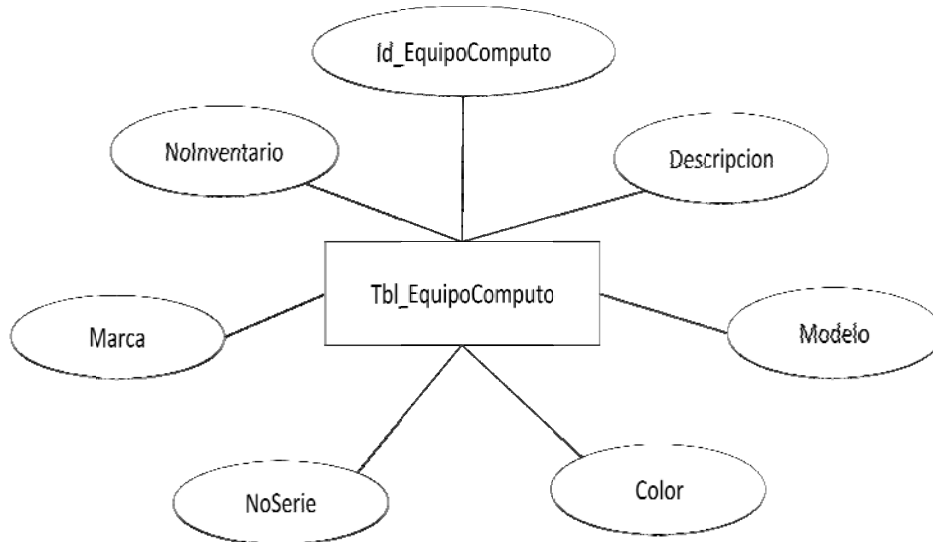


Figura 4.20 Entidad "Tbl_EquipoComputo"

Tbl_EquipoEx: Contiene los datos del equipo externo.

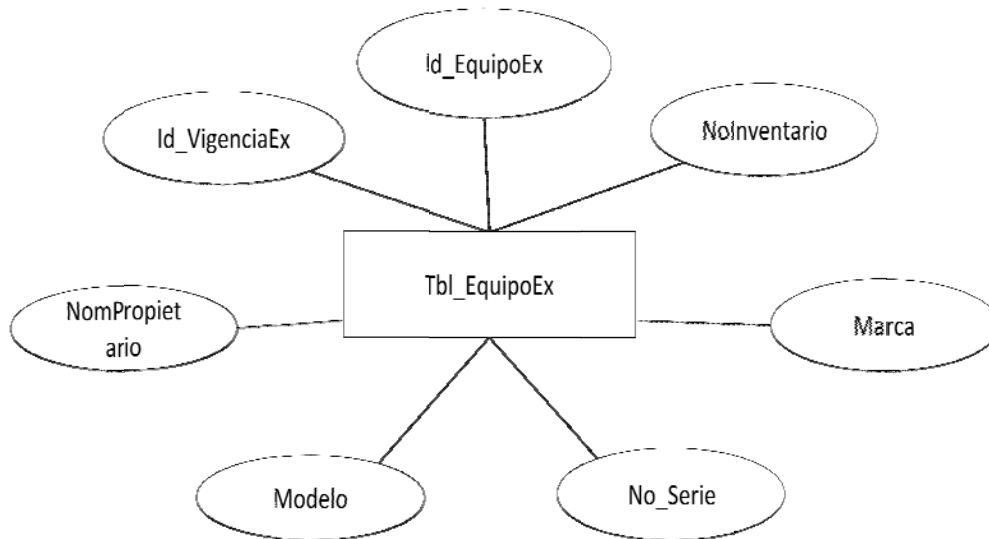


Figura 4.21 Entidad "Tbl_EquipoEx"

Tbl_HistoricoStatus: Contiene los datos históricos de los status de una solicitud.

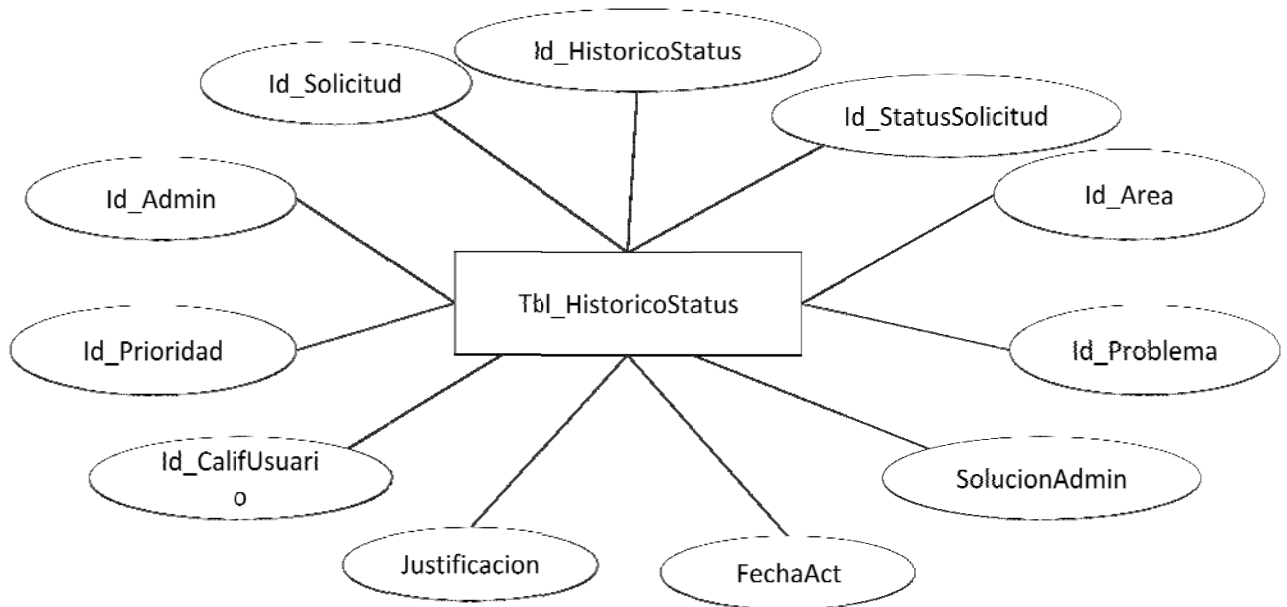


Figura 4.22 Entidad “Tbl_HistoricoStatus”

Tbl_Menu: Contiene las opciones de menú por tipo de usuario.

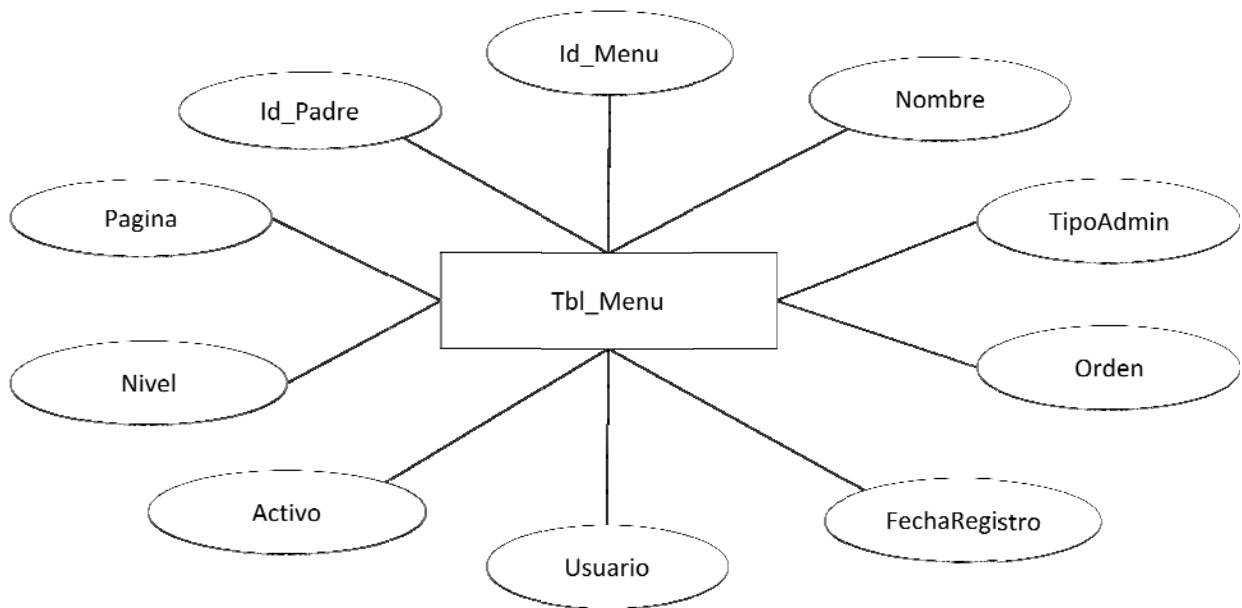


Figura 4.23 Entidad “Tbl_Menu”

Tbl_Noticias: Contiene las noticias para los usuarios.

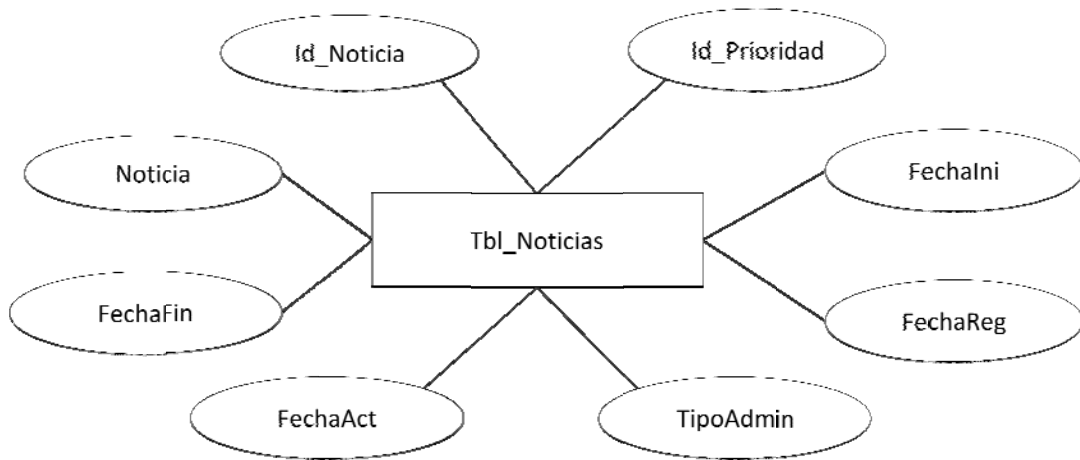


Figura 4.24 Entidad "Tbl_Noticias"

Tbl_ServicioEx: Contiene las solicitudes que son enviadas a servicio externo.

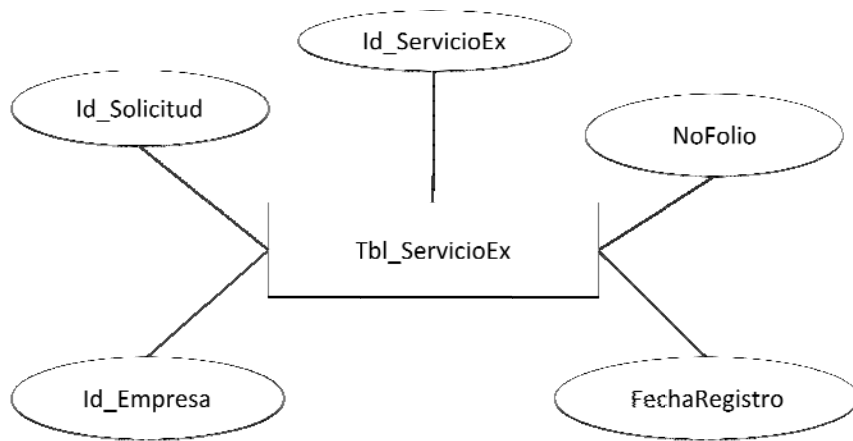


Figura 4.25 Entidad "Tbl_ServicioEx"

Tbl_Software: Contiene el software de referencia para los usuarios.

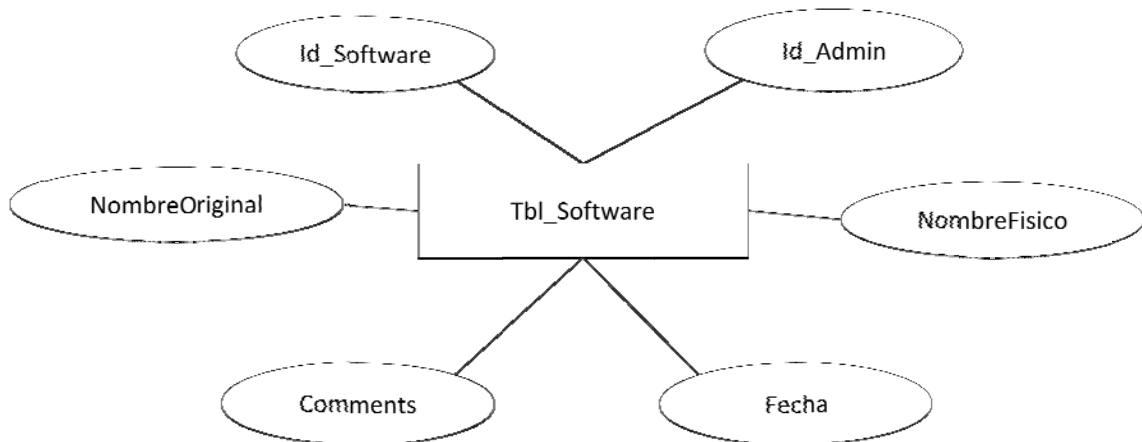


Figura 4.26 Entidad "Tbl_Software"

Tbl_Solicitud: Contiene las solicitudes de servicio registradas en el sistema.

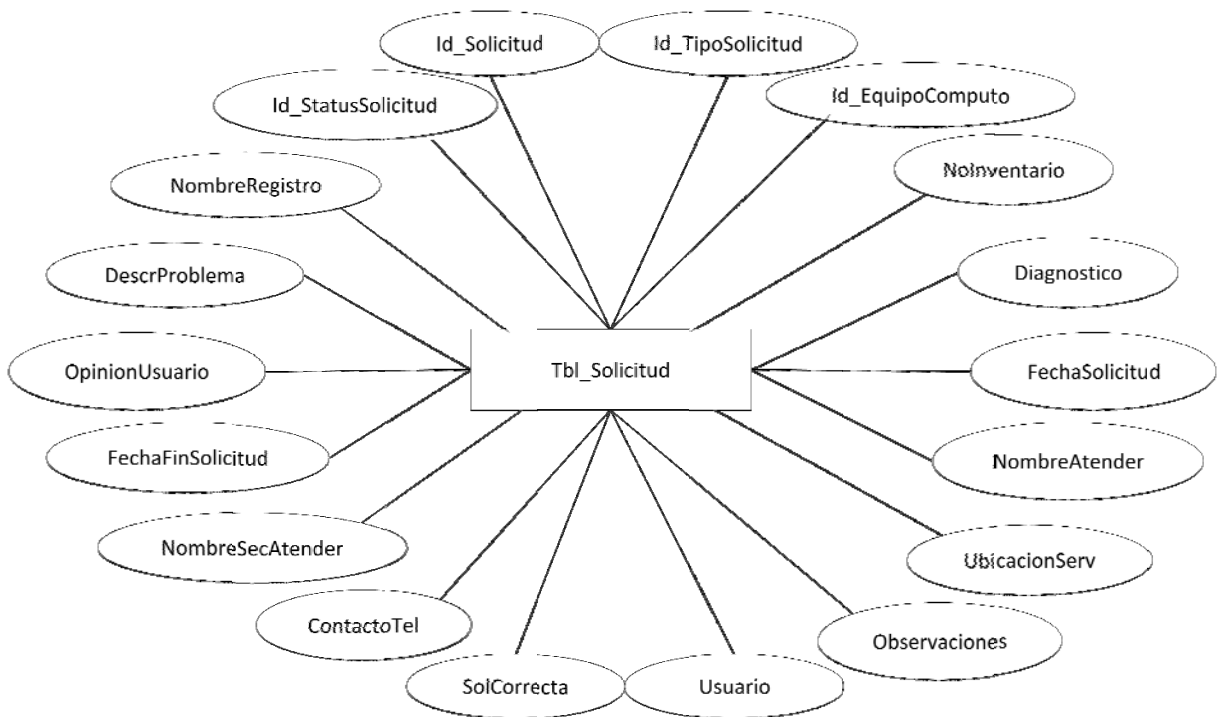


Figura 4.27 Entidad “Tbl_Solicitud”

Tbl_UsuarioEx: Contiene el registro de usuarios externos.

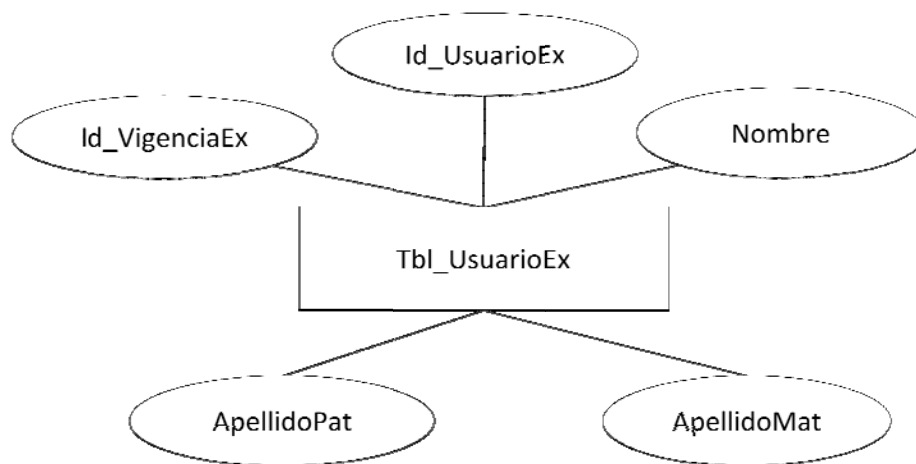


Figura 4.28 Entidad “Tbl_UsuarioEx”

Tbl_VigenciaEx: Contiene el registro de vigencia externa.

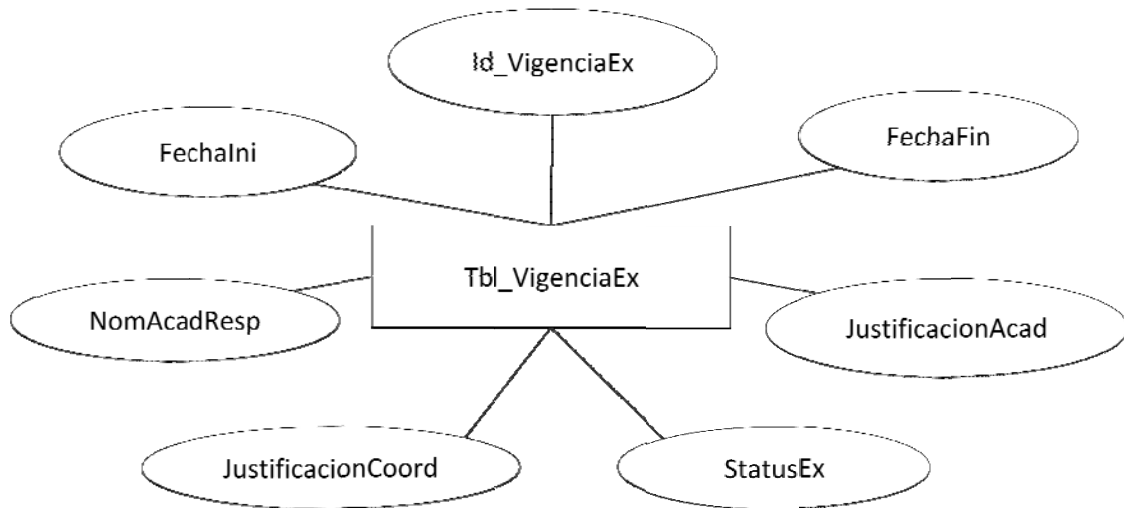


Figura 4.29 Entidad “Tbl_VigenciaEx”

Tbl_VigHistEquipo: Contiene el registro histórico de vigencias de equipos externos.

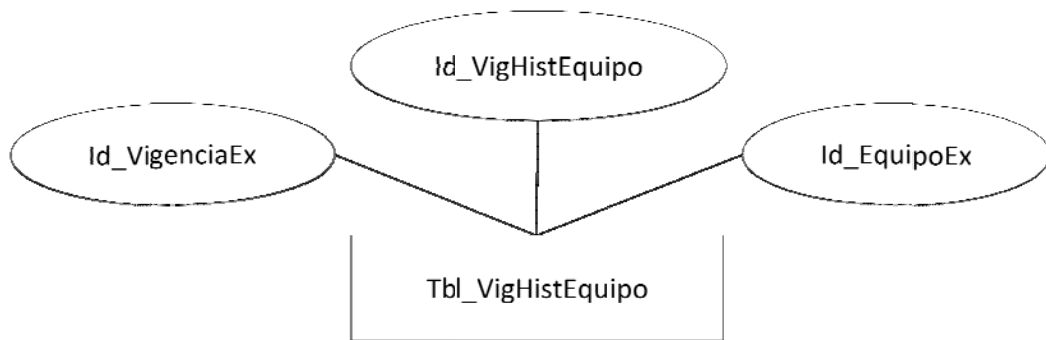


Figura 4.30 Entidad “Tbl_VigHistEquipo”

Tbl_VigHistUsuario: Contiene el registro histórico de vigencias de usuarios externos.

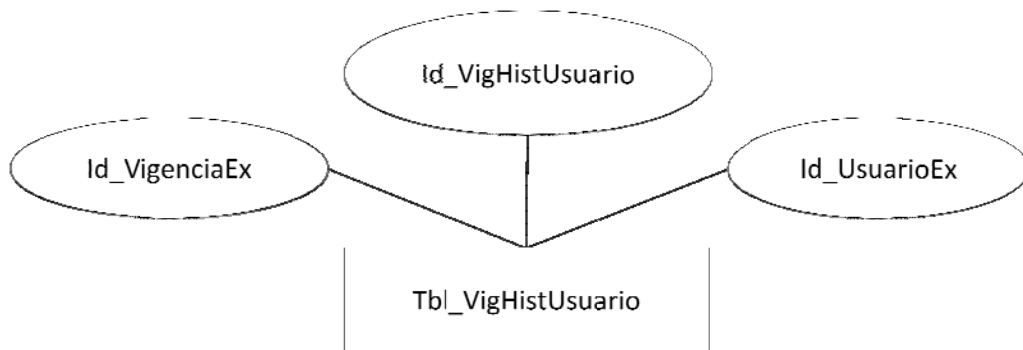


Figura 4.31 Entidad “Tbl_VigHistUsuario”

4.2 DISEÑO DE CASOS DE USO

Un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto. ^[1]

Los casos de uso fueron introducidos por Jacobson en 1992, sin embargo, la idea de especificar un sistema a partir de su interacción con el entorno es original de Mc Menamin y Palmer, dos precursores del análisis estructurado. Gran parte de los más reconocidos especialistas en métodos orientados a objetos coincidieron en considerar a los casos de uso como una excelente forma de especificar el comportamiento externo de un sistema. De esta forma, la notación de los casos de uso fue incorporada al lenguaje estándar de modelado UML propuesto por Ivar Jacobson, James Rumbaugh y Grady Booch, tres de los precursores de las metodologías de análisis y diseño orientado a objetos, y avalado por las principales empresas que desarrollan software en el mundo. UML va en camino de convertirse en un estándar para modelado de sistemas de software de amplia difusión.

Los requisitos funcionales se expresan como casos de uso en un modelo de casos de uso. Este modelo ayuda al cliente, a los usuarios y a los desarrolladores a llegar a un acuerdo sobre cómo utilizar el sistema. La mayoría de los sistemas tienen muchos tipos de usuarios. Cada tipo de usuarios se representa mediante un actor. Los actores utilizan el sistema al interactuar con los casos de uso. Todos los actores y casos de uso del sistema forman un modelo de casos de uso.

Los actores se comunican con el sistema mediante el envío y recepción de mensajes hacia y desde el sistema. Para el Sistema de Solicitudes, se identifican los siguientes actores:

- Usuario: Miembros de la comunidad del Instituto que pueden realizar una solicitud de servicio.
- Administrador 1: Jefes de área y coordinador de la CSC.
- Administrador 2: Personal de área de la CSC.

Los diagramas de caso de uso para cada uno de los actores del sistema se detallan del punto 4.2.1 al 4.2.3 de acuerdo a la notación básica de UML(Figura 4.32)

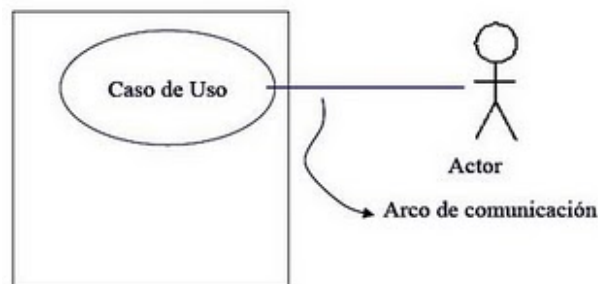


Figura 4.33 Notación básica de UML

4.2.1 Casos de uso para administrador 1

Para el administrador 1, las actividades identificadas de acuerdo a los requerimientos recabados, se muestran en el diagrama de casos de uso de la figura 4.34

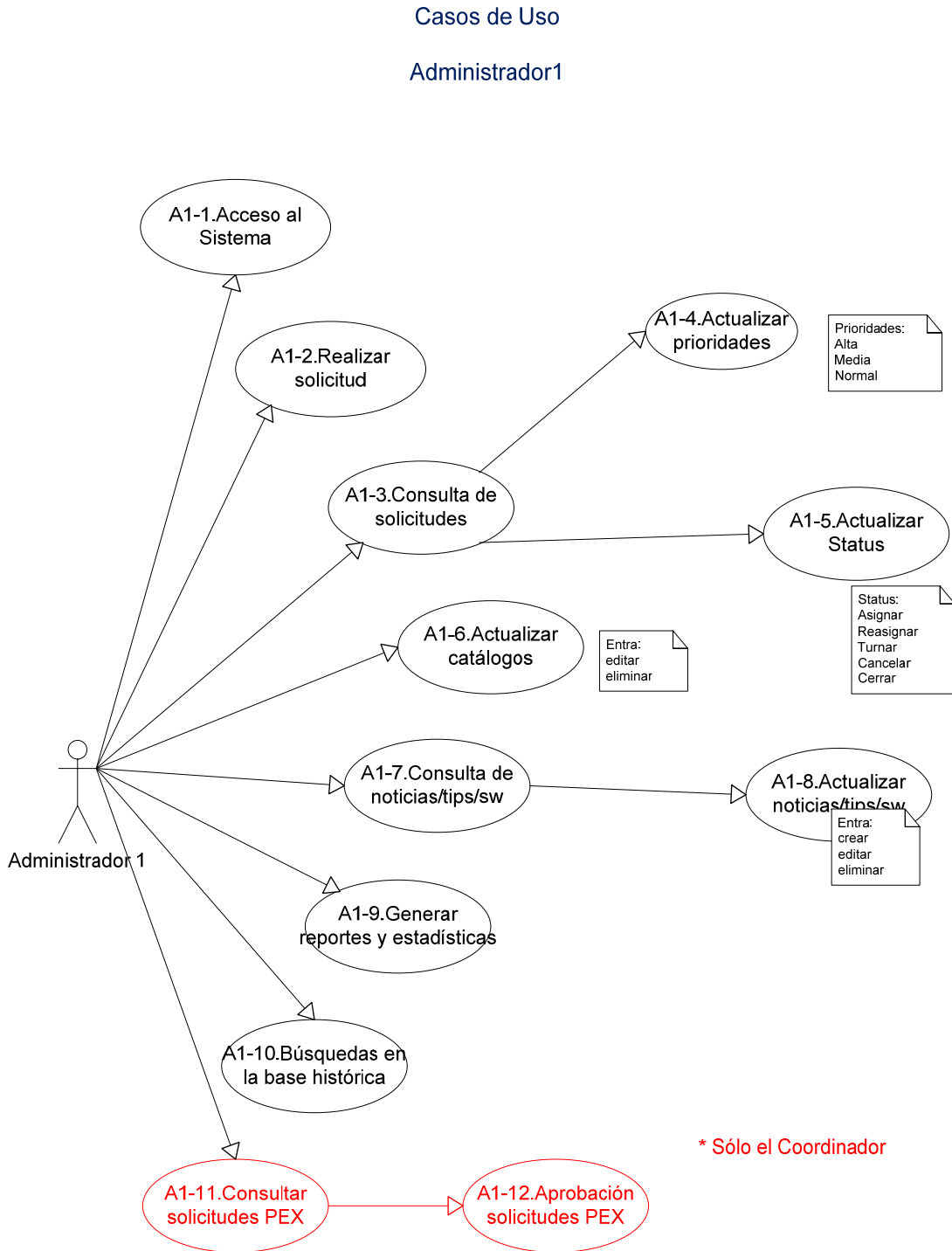


Figura 4.34 Casos de uso del Administrador 1

Un ejemplo del formato que se usó para la descripción de cada uno de los casos de uso del Administrador1 es el siguiente:

Caso de uso: Generar reportes y estadísticas.

Actor: Administrador

Descripción: Se pueden generar reportes y estadísticas de la información más relevante y requerida por el Coordinador de Sistemas de Cómputo.

Pre-requisitos:

- Deben existir solicitudes cerradas. Sólo tendrán acceso el Coordinador y jefes de área.
- Haber ejecutado el caso de uso Ingresar al sistema.

Flujo del caso de uso:

Actor		Sistema		
Paso	Acción	Paso	Acción	Id Excepción
1	Selecciona la opción “Reportes y estadísticas”.	2	Muestra los tipos de reportes y/o estadísticas.	
3	Selecciona el tipo de reporte y/o estadística.	4	Genera y muestra el reporte y/o estadísticas.	
5	Selecciona la opción “Generar en Word o PDF”	6	Genera el reporte en Word o PDF	

Excepciones del caso de uso:

Ninguna.

Post-condiciones:

Obtener información de la actividad realizada por parte del personal de la CSC.

El correspondiente diagrama de flujo para el caso de uso “Generar reportes y estadísticas” se muestra en la figura 4.35

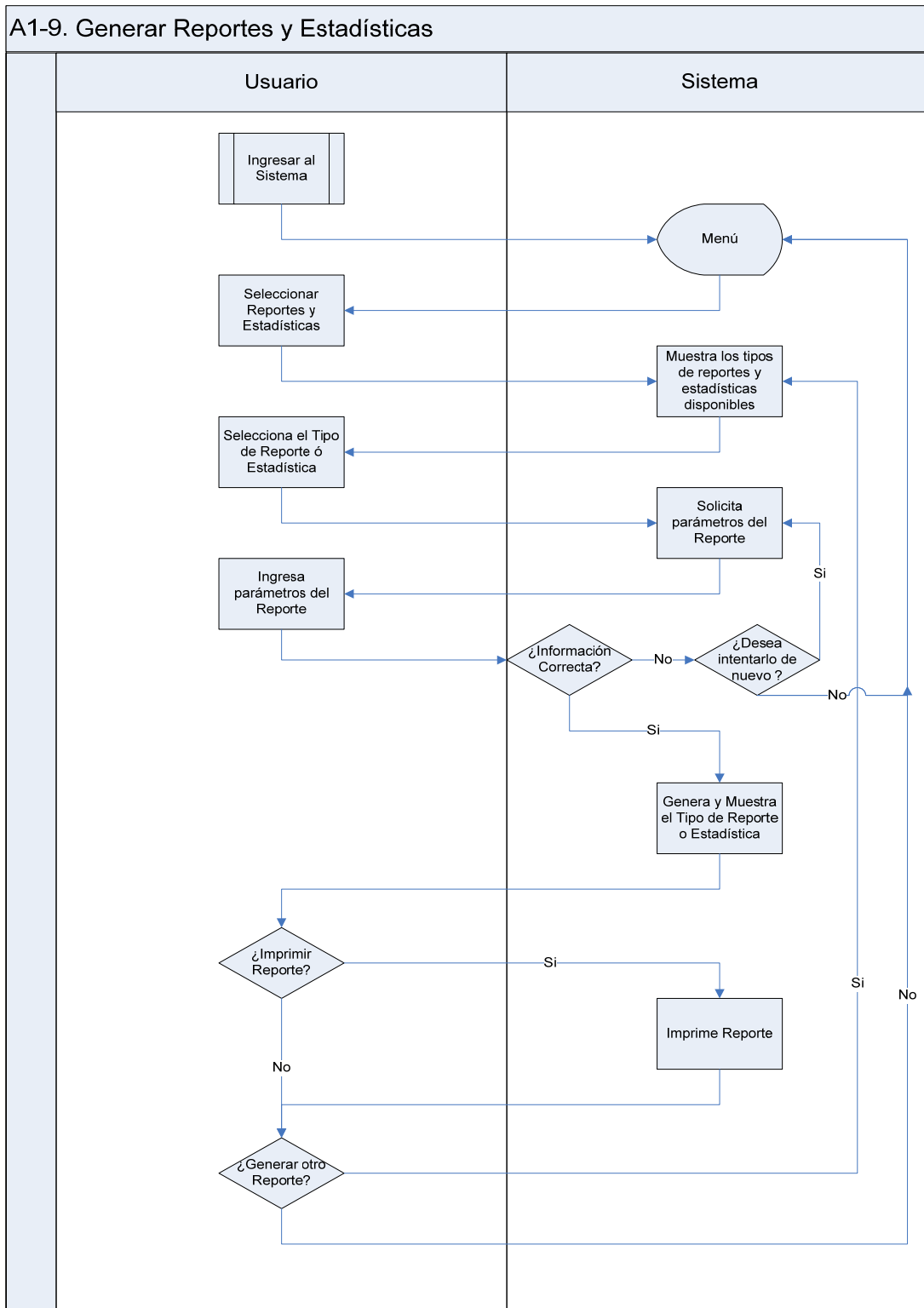


Figura 4.35 Diagrama de caso de uso “Generar reportes y estadísticas”

4.2.2 Casos de uso para administrador 2

Para el administrador 2, las actividades identificadas de acuerdo a los requerimientos recabados, se muestran en el diagrama de casos de uso de la figura 4.36

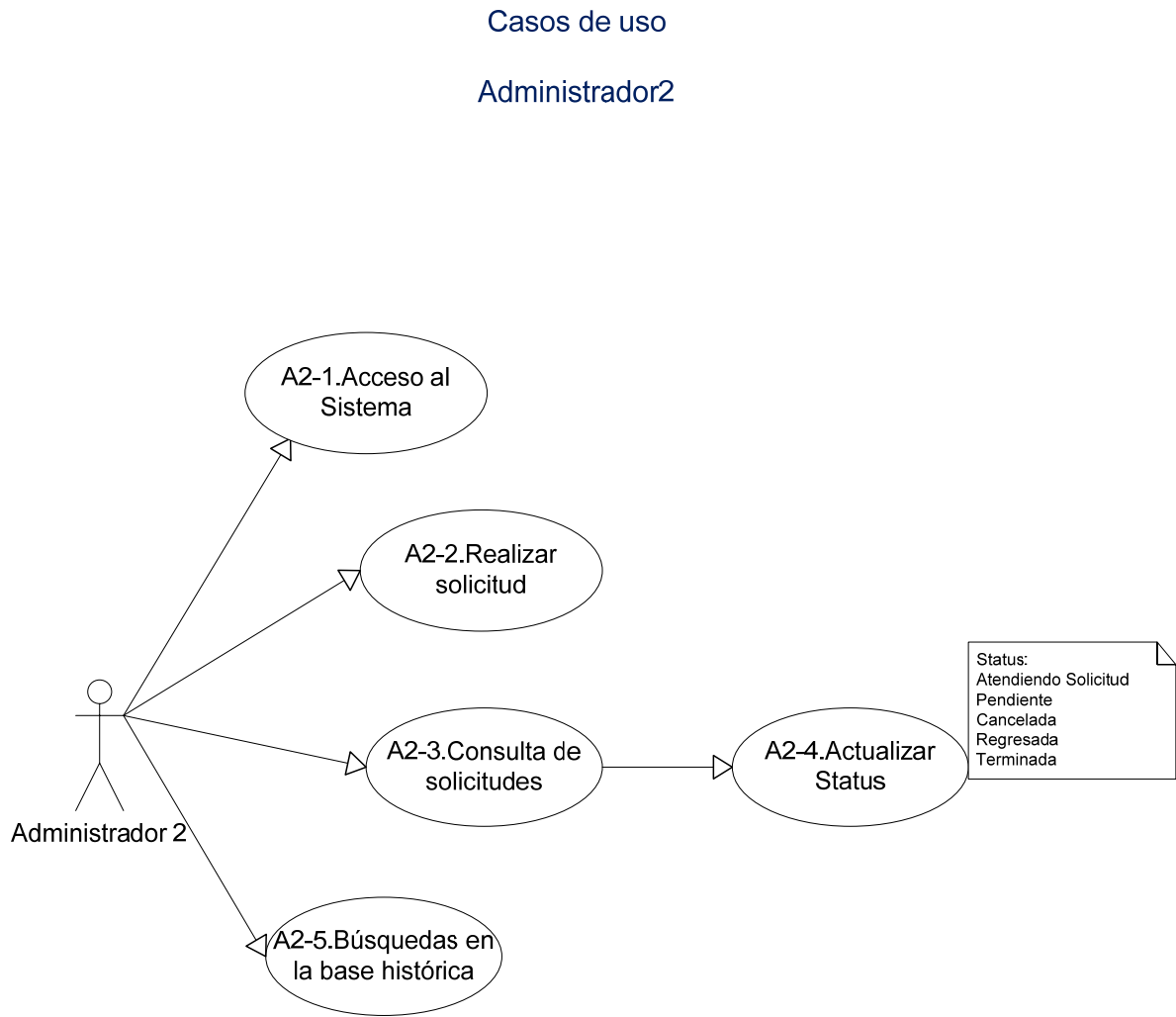


Figura 4.36 Casos de uso del Administrador 2

Un ejemplo del formato que se usó para la descripción de cada uno de los casos de uso del Administrador2 es el siguiente:

Caso de uso: Búsquedas en la base histórica.

Actor: Administrador

Descripción: El personal de cómputo podrá realizar búsquedas en la base histórica de todas las solicitudes concluidas que correspondan al área, por nombre de usuario, inventario equipo, etc.

Pre-requisitos:

- Debe existir por lo menos un registro de solicitudes concluidas.
- Haber ejecutado el caso de uso Ingresar al sistema.

Flujo del caso de uso:

Actor		Sistema		
Paso	Acción	Paso	Acción	Id Excepción
1	Selecciona la opción “Búsquedas históricas”.	2	Muestra el tipo de búsquedas que puede realizar.	
3	Selecciona el tipo de búsqueda.	4	Muestra las solicitudes encontradas.	

Excepciones del caso de uso: Ninguna.

Post-condiciones:

Consulta de solicitudes históricas.

El correspondiente diagrama de flujo para el caso de uso “Búsquedas en la base histórica” se muestra en la figura 4.37

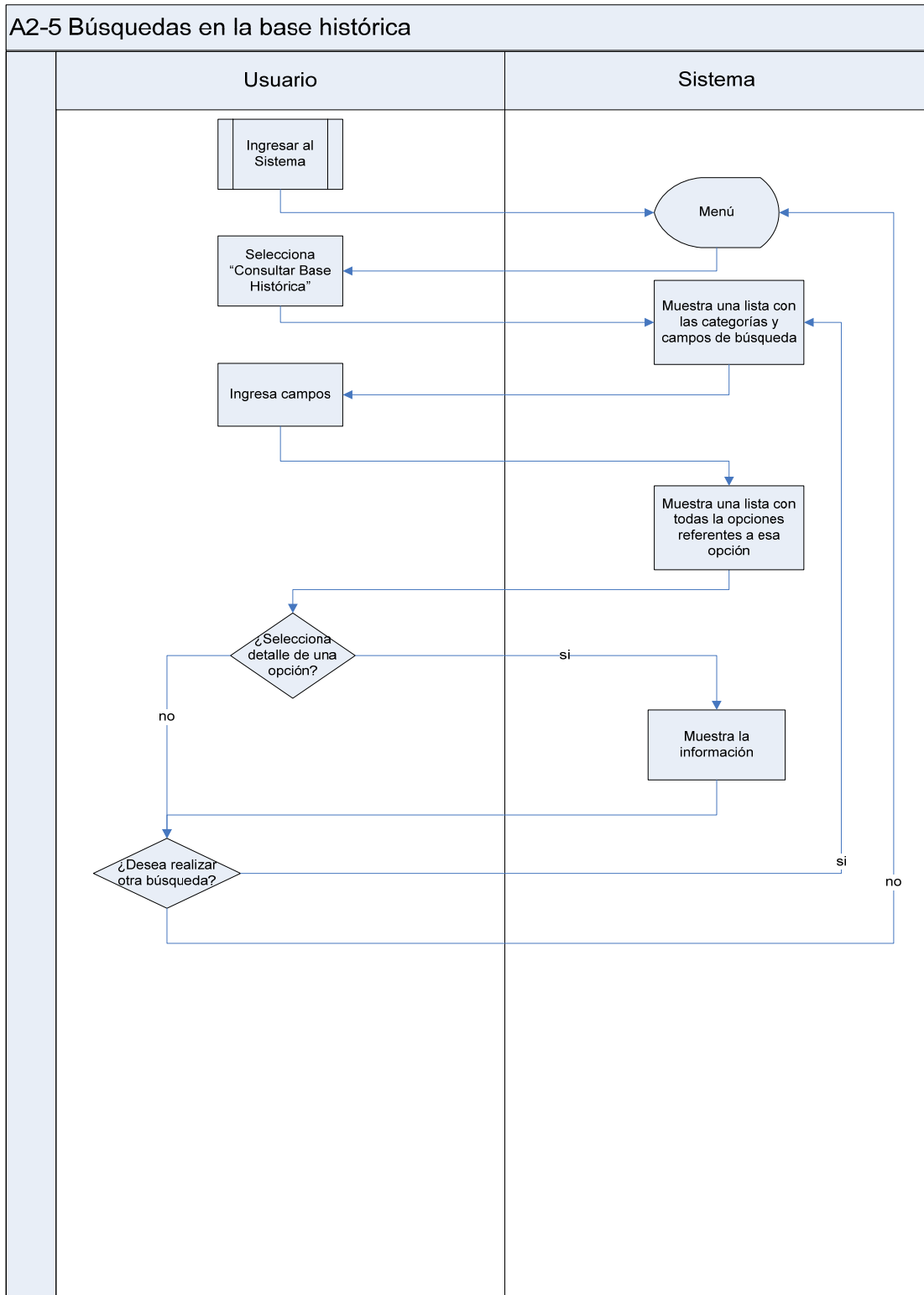
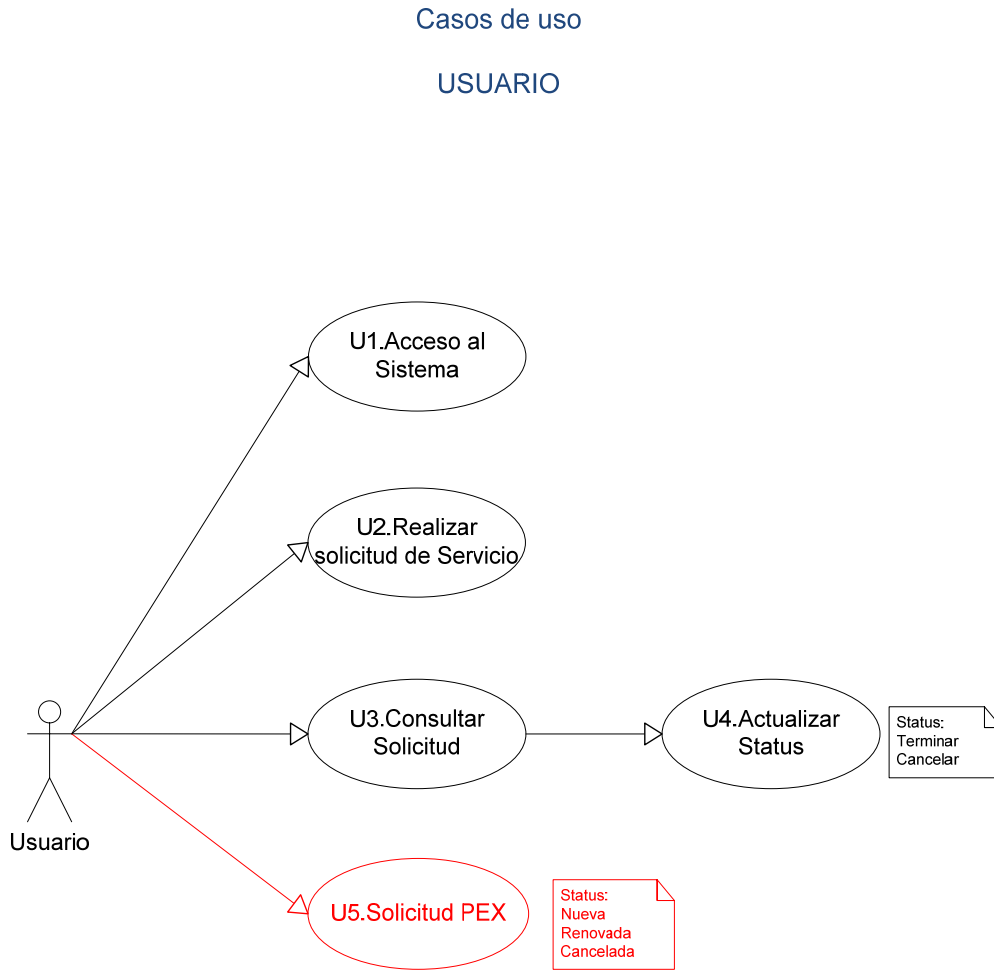


Figura 4.37 Diagrama de caso de uso “Búsquedas en la base histórica”

4.2.3 Casos de uso para usuarios

Para los usuarios, las actividades identificadas de acuerdo a los requerimientos recabados, se muestran en el diagrama de casos de uso de la figura 4.38



* Sólo Académicos y personal autorizado

Figura 4.38 Casos de uso del usuario

Un ejemplo del formato que se usó para la descripción de cada uno de los casos de uso del Usuario es el siguiente:

Caso de uso: Realizar solicitud

Actor: Usuario.

Descripción: El usuario realiza una solicitud con la descripción de sus necesidades.

Pre-requisitos:

- Ser usuario validado.
- El equipo a ser revisado debe estar registrado en el Sistema de Inventarios, de lo contrario deberá solicitar autorización para registrar el equipo externo.
- Haber ejecutado el caso de uso Ingresar al sistema.

Flujo del caso de uso:

Actor		Sistema		
Paso	Acción	Paso	Acción	Id Excepción
1	Selecciona la opción “Realizar solicitud” del menú.	2	Muestra en pantalla los datos personales del usuario.	
3	Proporciona el número de inventario del equipo y presiona el botón “Continuar”.	4	Valida el número de inventario proporcionado.	e1, e2
		5	Muestra en pantalla los datos del equipo.	
6	Captura la información necesaria para realizar la solicitud.			
7	Presiona el botón “Enviar solicitud”.	8	Valida la información capturada.	e3
		9	Guarda la información capturada.	
		10	Envía al usuario y administradores notificación vía correo electrónico con un número de registro para poder consultar el avance de la solicitud realizada.	
		11	Muestra el detalle de la solicitud.	
12	Selecciona la opción	13	Muestra las opciones de impresión.	

	“Imprimir comprobante”			
14	Imprime el comprobante.			

Excepciones del caso de uso:

Id	Descripción	Acción
e1	Información incorrecta.	Revisar y corregir información capturada.
e2	Número no existente en la base de datos.	Hacer petición de registro del equipo.
e3	Información incorrecta.	Revisar y corregir información capturada.

Post-condiciones:

Registro de la información de la solicitud realizada.

El correspondiente diagrama de flujo para el caso de uso “Realizar solicitud” se muestra en la figura 4.39

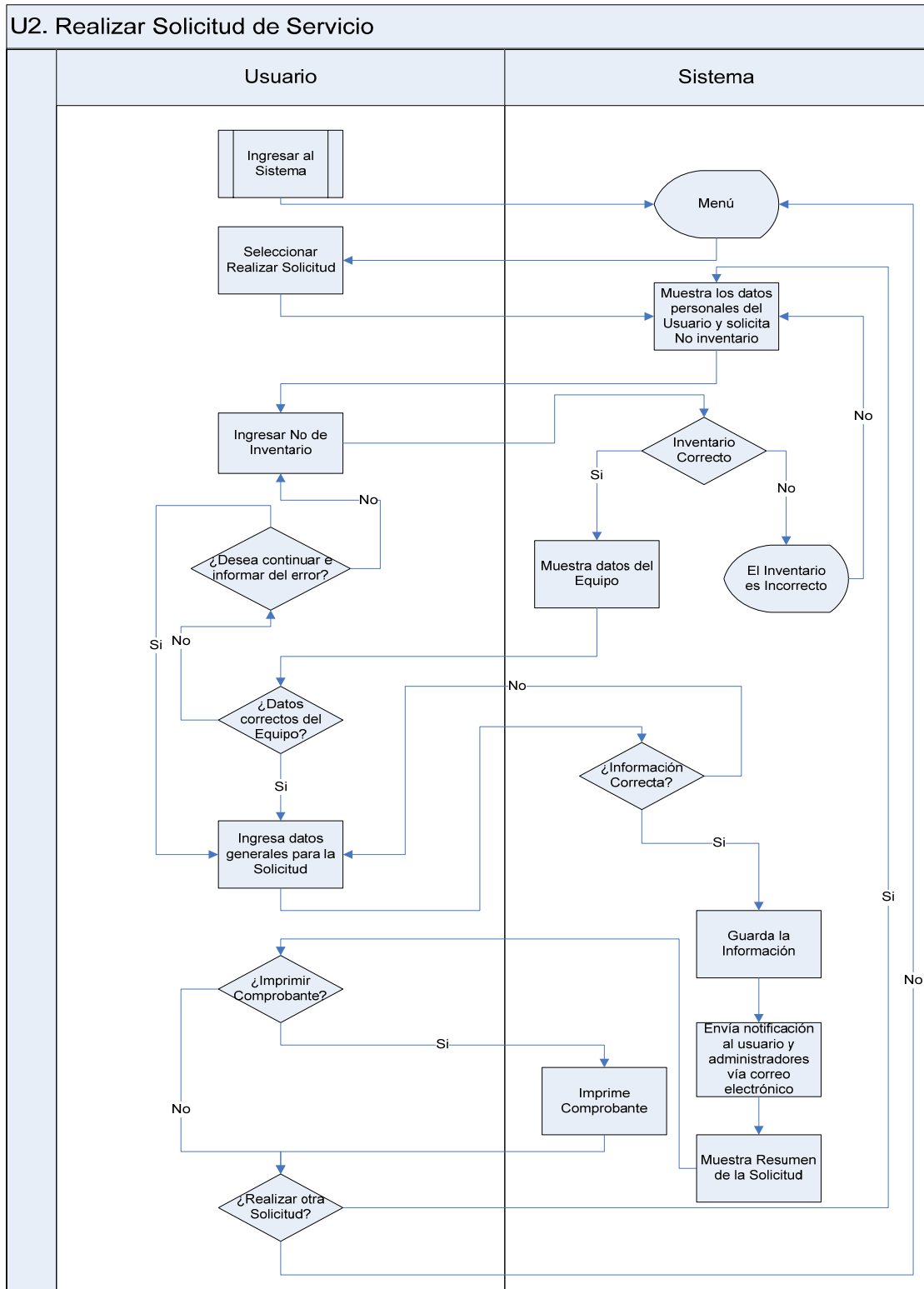


Figura 4.39 Diagrama de caso de uso “Realizar solicitud”

4.2.4 Diseño de la interfaz

Interfaz es el conjunto de elementos gráficos y/o multimedia que permite la interacción del usuario con la computadora o más concretamente con nuestro software. Al realizar un buen diseño de interfaz de usuario, debemos realizar una investigación previa que nos proporcione información para asegurar el éxito de la misma, tal como:

- **Tipo de Usuario**
La interfaz será diseñada de acuerdo al tipo de usuario. Si es un usuario inexperto, la interfaz deberá ser muy amigable e intuitiva, llevando de la mano al usuario; por el contrario si el usuario es experto, la interfaz será más flexible con opciones de ajustes y parámetros de acuerdo a su conocimiento.
- **Tipo de trabajo**
Si el software será usado para una actividad repetitiva, la interfaz deberá ser ágil, dando prioridad a que todo se realice presionando teclas fáciles de recordar y de preferencia maximizando el uso del ratón.
- **Dispositivos de interfaz**
El tipo de dispositivo de interfaz empleado afecta también la interfaz de usuario para explotar las características propias del dispositivo.

Buscando una interfaz que le resulte familiar a los usuarios, se planteo una solución en la que la navegación por el sistema sea intuitiva y sencilla, cabe señalar que dependiendo del rol que se juegue en el sistema, el menú de opciones será un árbol diferente de navegación como ya lo pudimos notar en los diagramas de casos de uso del capítulo anterior.

También tomamos en cuenta algunos principios básicos que nos guiaron en el diseño:

- **Uniformidad**
Es fundamental no saturar nuestra interfaz de colores, debemos mantener en todas las pantallas los mismos colores, tipos de letras y tamaños.
- **Alineación**
La correcta alineación permite identificar el flujo de entrada de datos, que el usuario seguirá, y guiar la vista del usuario hacia donde queremos que vea.
- **Encuadre**
Es conveniente delimitar áreas que permitan identificar los distintos tipos de información a capturar, ya que esto le permite al usuario tener una lectura eficiente.

MAPA DE NAVIGACIÓN

Las siguientes figuras ilustran la navegación que puede realizar el usuario y los administradores en el sistema:

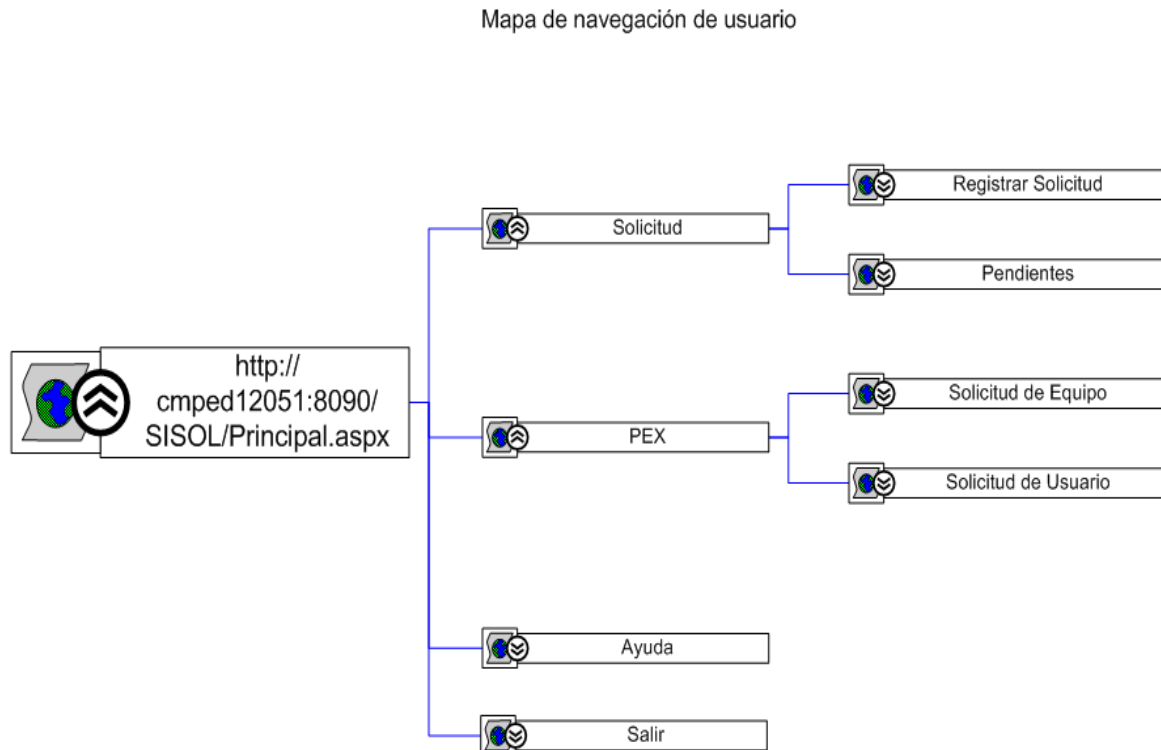


Figura 4.40 Mapa de navegación para usuarios

Mapa de navegación de administrador 1

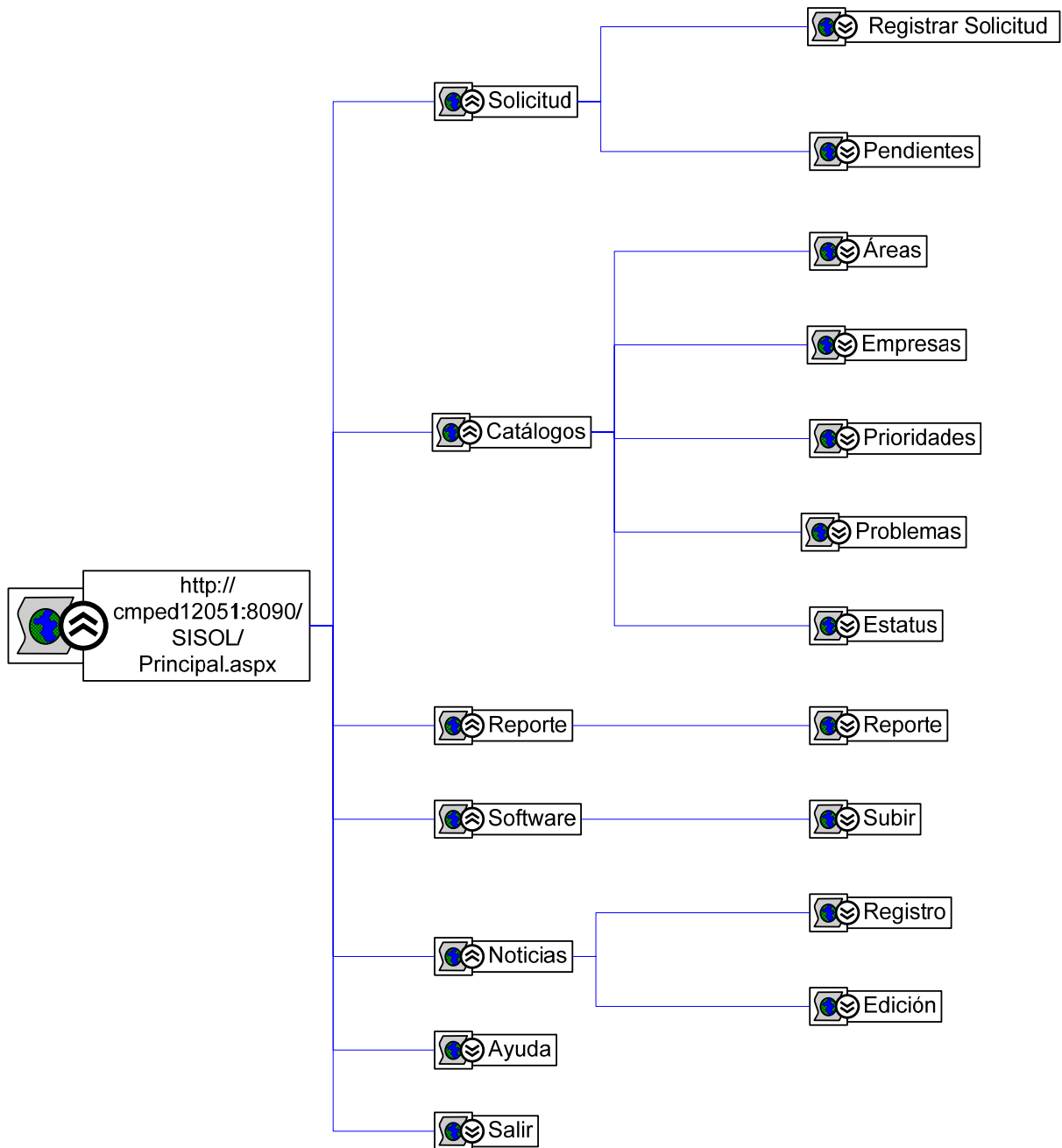


Figura 4.41 Mapa de navegación para administrador 1

Mapa de navegación de administrador 2

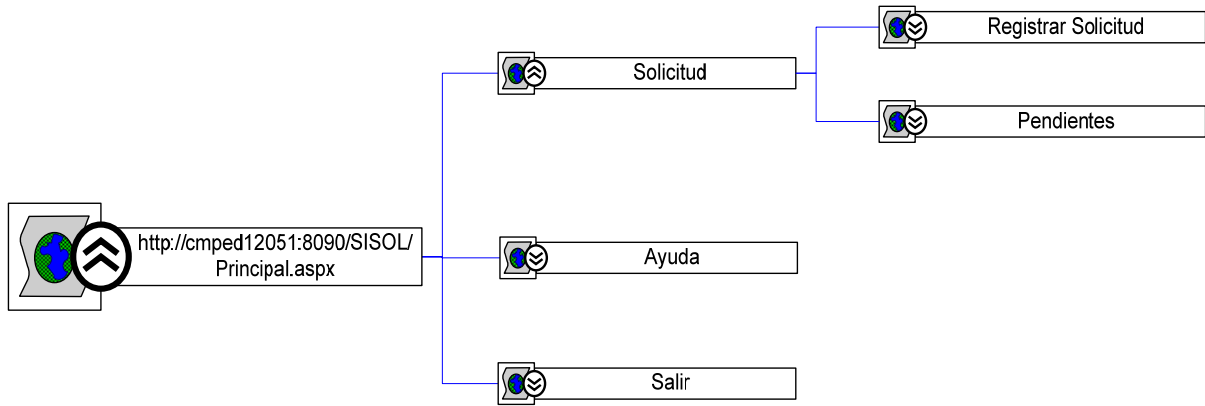


Figura 4.42 Mapa de navegación para administrador 2