

**Informe final* del Proyecto Q018
Revisión técnica del modelo BIÓTICA**

Responsable: M. en C. Luis Fernando Castro Careaga
Institución: Universidad Autónoma Metropolitana-Iztapalapa
Dirección: Av Michoacán y La Purísima s/n, Vicentina, México, DF, 09340 , México
Correo electrónico: lucas@xanum.uam.mx
Teléfono/Fax: N/D
Fecha de inicio: Junio 15, 1998
Fecha de término: Septiembre15, 1999
Principales resultados: Informe final
Forma de citar el informe final y otros resultados:** Castro Careaga, L. F., 1999. Revisión técnica del modelo BIÓTICA. Universidad Autónoma Metropolitana-Iztapalapa. **Informe final SNIB-CONABIO proyecto No. Q018.** México D. F.

Resumen:

En todos los sistemas de información, la estructuración de la información es fundamental para determinar la estructura del sistema, que a su vez determina la facilidad de la construcción de los sistemas y su mantenimiento. El documento emitido por el proyecto describe las observaciones realizadas a la base de datos del sistema Biótica. Las observaciones parten de los diagramas de entidades y asociaciones en ER-Win que documentan el sistema y su correspondiente diccionario de datos.

-
- * El presente documento no necesariamente contiene los principales resultados del proyecto correspondiente o la descripción de los mismos. Los proyectos apoyados por la CONABIO así como información adicional sobre ellos, pueden consultarse en www.conabio.gob.mx
 - ** El usuario tiene la obligación, de conformidad con el artículo 57 de la LFDA, de citar a los autores de obras individuales, así como a los compiladores. De manera que deberán citarse todos los responsables de los proyectos, que proveyeron datos, así como a la CONABIO como depositaria, compiladora y proveedora de la información. En su caso, el usuario deberá obtener del proveedor la información complementaria sobre la autoría específica de los datos.

Observaciones sobre la Base de Datos del Sistema de BIÓTICA

Elaborada para:



Por:

Laboratorio de Ingeniería de
Software
Área de Computación y Sistemas
Departamento de Ing. Eléctrica
UAM-Iztapalapa

Contenido

<i>1. Presentación</i>	<i>1</i>
<i>2. Aspectos de Diseño</i>	<i>2</i>
Propuesta	2
Referencia del Hijo hacia el Padre.	2
Inclusión de la jerarquía en el identificador del registro.	6
Conclusiones Respecto a los Aspectos de Diseño	9
<i>3. Aspectos de Forma</i>	<i>10</i>
Presentación de los Diagramas	10
Propuesta	10
<i>4. Aspectos de Optimización</i>	<i>11</i>
Propuesta	11
<i>Anexo 1 Diagramas de la Base de Datos</i>	<i>12</i>
Diagrama de Lugares	13
Diagrama de Nombres	13
Diagrama de Grupos y Personas	13
Diagrama de Colecciones e Instituciones	13
Diagrama de Ejemplares	13
Diagrama de Bibliografía	13
<i>Anexo 2 Observaciones a la Documentación de la Base de Datos de Biótica</i>	<i>19</i>
Comentarios específicos al Documento	19
Comentarios generales	20

1. Presentación

En todos los sistemas de información, la estructuración de la información es fundamental para determinar la estructura del sistema, que a su vez determina la facilidad de la construcción de los sistemas y su mantenimiento.

El presente documento describe las observaciones realizadas a la base de datos del sistema Biótica. Las observaciones parten de los diagramas de entidades y asociaciones en ER-Win que documentan el sistema y su correspondiente diccionario de datos, considerándose que no existe documentación adicional.

Las observaciones constan de dos partes: la primera en donde se plantea la observación como tal y la segunda en donde se propone una solución a la situación detectada.

El documento está estructurado de acuerdo a los tipos de observaciones encontrados:

- Diseño
- Forma
- Optimización

En la parte final se muestran las propuestas de ajustes en la diagramación.

2. Aspectos de Diseño

En esta sección se describen los aspectos que tienen que ver con la forma en que se concibió el modelo.

La estructura de la base de datos en general puede considerarse simple, de tal forma que las observaciones de diseño se centran exclusivamente en la implantación de estructuras complejas de la información.

Existe gran número de autoreferencias en el modelo (Nombre, Ejemplar, Bibliografía). Estas principalmente plantean una jerarquización de la información, las cuales dependen de la forma de implantación para facilitar o complicar la consulta de información.

Propuesta

Hay distintos tipos de implantación en tablas de estructuras jerárquicas, las cuales presentan ventajas y desventajas que deben evaluarse dependiendo de la aplicación.

Las formas de implantación se describen en las siguientes subsecciones.

Referencia del Hijo hacia el Padre.

El registro tiene un campo que hace referencia hacia la llave del registro que corresponde al padre. Para el registro que debe colocarse como raíz, el valor de referencia hacia el padre tendrá un valor nulo.

Árbol (Campo_Llave, Campo-Referencia-al-Padre, Otros-Campos)

Las ventajas de este esquema es la de tener un número de niveles flexible de tal forma que puedan incluirse nuevos niveles cuando sea necesario.

Sus desventajas son que dado un registro, no se sabe directamente sus ancestros o descendientes indirectos. Esta situación plantea consultas complejas para poder resolver este tipo de requerimiento.

Su aplicación más recomendable es cuando es necesario un número variable de niveles y navegación de sucesor y antecesores inmediatos.

Este caso se aplica correctamente con la entidad Nombre y las entidades Sinónimia Nomenclatural y Basónimo.

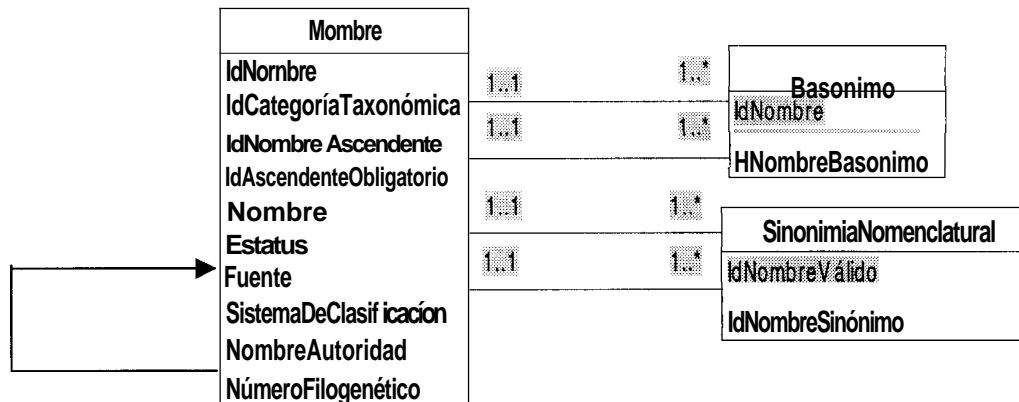


Figura 1. Jerarquía utilizando Entidades

La estructura de la tabla Nombre está compuesta por:

Campos	Formato del Campo
IdNombre	Numérico (Campo Llave de Referencia)
IdCategoríaTaxonomica	Numérico
IdNombreAscendente	Numérico (Campo de Referencia al Padre)
IdAscendObligatorio	Numérico
Nombre	Texto
Status	Numérico
Fuente	Texto
SistemaDeClasificacion	Texto
NombreAutoridad	Texto
NumeroFilogenetico	Texto

Donde el campo *IdNombre* realiza la función de Referencia hacia los registros en general, el campo *IdNombreAscendente* es el campo de referencia hacia el padre sucesor y demás campos son datos complementarios.

A continuación se muestran las consultas para la obtención de información de acuerdo al esquema actual de la tabla «Nombre» y el esquema a proponer.

Consultas

A) Consulta para obtener un Nodo y todos sus hijos.

A.1) Consulta para obtener un Nodo y sus hijos inmediatos.

```
SELECT Nombre.IdNombreAscendente, Nombre.IdNombre, Nombre.Nombre,
Nombre.Fuente, Nombre.NombreAutoridad
FROM Nombre, Nombre AS Aux1, Nombre AS Aux2
WHERE Aux1.IdNombre=ID-NOMBRE AND
Aux2.IdNombreAscendente=Aux1.IdNombre AND
Nombre.IdNombre=Aux2.IdNombre;
```

A.2) Consulta para obtener un Nodo y sus hijos dos niveles mas.

```
SELECT Nombre.IdNombreAscendente, Nombre.IdNombre, Nombre.Nombre,
Nombre.Fuente, Nombre.NombreAutoridad
FROM Nombre, Nombre AS Aux1, Nombre AS Aux2, Nombre AS Aux3
WHERE Aux1.IdNombre=ID-NOMBRE AND
```

```
Aux2.IdNombreAscendente=Aux 1.IdNombre AND
Aux3.IdNombreAscendente=Aux2.IdNombre AND
Nombre.IdNombre=Aux3.IdNombre;
```

Ventajas

Es la obtención rápida de los registros sucesores (Hijo) inmediatos mostrando un procesamiento muy rápido en la obtención de la información. La elaboración de la consulta al nivel inmediato siguiente es relativamente simple.

Desventajas

Para poder obtener todos los sucesores posteriores de un registro dado, la consulta se complica y sólo muestra los sucesores del nivel deseado, sin mostrar los sucesores de los niveles intermedios. Ya que el número de niveles del árbol de nombres es variable dependiendo de las ramas que se están consultando, una consulta genérica tiene el problema de que, para árboles cortos la consulta no proporciona resultados pues no se cumplen las condiciones para los join's de los últimos niveles. La forma de resolverlo implica programación con utilización de ciclos que manejen condiciones en lenguajes distintos a SQL.

B) Consulta para obtener un Nodo y todos sus antecesores.

B.1) Consulta para obtener un Nodo y su Padre inmediato.

```
SELECT Nombre.IdNombre, Nombre.IdNombreAscendente, Nombre.Nombre
       Nombre.Fuente, Nombre.NombreAutoridad
FROM Nombre, Nombre AS Aux 1, Nombre AS Aux2
WHERE Aux1.IdNombre=ID_NOMBRE AND
       Aux2.IdNombre=Aux1.IdNombreAscendente AND
       Nombre.IdNombre=Aux2.IdNombre;
```

B.2) Consulta para obtener un Nodo y sus Padres dos niveles más.

```
SELECT Nombre.IdNombre, Nombre.IdNombreAscendente, Nombre.Nombre,
       Nombre.Fuente, Nombre.NombreAutoridad
FROM Nombre, Nombre AS Aux 1, Nombre AS Aux2, Nombre AS Aux3
WHERE Aux1.IdNombre=ID_NOMBRE AND
       Aux2.IdNombre=Aux1.IdNombreAscendente AND
       Aux3.IdNombre=Aux2.IdNombreAscendente AND
       Nombre.IdNombre=Aux3.IdNombre;
```

Ventajas

Es la obtención rápida del registros antecesor (Padre) inmediato mostrando un procesamiento muy rápido en la obtención de la información.

En estas consultas es la obtención rápida del registro antecesor (Padre) inmediato mostrando un procesamiento muy rápido en la obtención de la información. La elaboración de la consulta al nivel inmediato anterior es relativamente simple.

Desventajas

Para poder obtener todos los predecesores anteriores de un registro dado, la consulta se complica y sólo muestra el predecesor del nivel deseado, sin mostrar los predecesores de los niveles intermedios. Ya que el número de niveles del árbol de nombres es variable dependiendo de las ramas que se están consultando, una consulta genérica tiene el problema de que para árboles cortos la consulta no

proporciona resultados pues no se cumplen las condiciones para los join's de los últimos niveles. La forma de resolverlo implica programación con utilización de ciclos que manejen condiciones en lenguajes distintos a SQL.

C) Consulta para agregar un Nodo.

```
INSERT INTO Nombre (IdNombre, IdCategoríaTaxonómica, IdNombreAscendente,
                    IdAscendObligatorio, Nombre, Estatus, Fuente, SistemaDeClasificación,
                    NombreAutoridad, NúmeroFilogenético )
SELECT Teclee-IdNombre, IdCategoríaTaxonómica, IdNombreAscendente,
        IdAscendObligatorio, Nombre, Estatus, Fuente, SistemaDeClasificación,
        NombreAutoridad, NúmeroFilogenético;
```

Ventajas

Esta consulta es muy sencilla ya que consiste sólo en introducir toda la información referente al registro y asegurar que el registro antecesor exista.

Desventajas

La revisión de existencia en principio puede hacerse mediante una regla de integridad, pero el registro raíz del árbol no la cumple pues es el único que no tiene antecesor. Por lo mismo la revisión de integridad debe hacerse con más código programando un Trigger.

D) Consulta para eliminar un Nodo y sus hijos.

D.1) Consulta para eliminar un Nodo y sus hijos inmediatos.

```
DELETE *
FROM Nombre, Nombre AS Aux1, Nombre AS Aux2
WHERE Aux1.IdNombre=IDNOMBRE AND
        Aux2.IdNombreAscendente=Aux1.IdNombre AND
        Nombre.IdNombre=Aux2.IdNombre;
```

D.2) Consulta para eliminar un Nodo y sus hijos dos niveles mas.

```
DELETE *
FROM Nombre, Nombre AS Aux1, Nombre AS Aux2, Nombre AS Aux3
WHERE Aux1.IdNombre=ID-NOMBRE AND
        Aux2.IdNombreAscendente=Aux1.IdNombre AND
        Aux3.IdNombreAscendente=Aux2.IdNombreAND
        Nombre.IdNombre=Aux3.IdNombre;
```

Ventajas

Es la eliminación rápida de los registros sucesores (Hijo) inmediatos mostrando un procesamiento muy rápido en la eliminación de la información. La elaboración de la consulta al nivel inmediato siguiente es relativamente simple.

Desventajas

Para poder eliminar todos los sucesores posteriores de un registro dado, la consulta se complica y sólo elimina a los sucesores del nivel deseado, sin eliminar a los sucesores de los niveles intermedios. Ya que el número de niveles del árbol de nombres es variable dependiendo de las ramas que se están consultando, una consulta genérica tiene el problema de que para árboles cortos la consulta no proporciona resultados pues no se cumplen las condiciones para los join's de los

últimos niveles. La forma de resolverlo implica programación con utilización de ciclos que manejen condiciones en lenguajes distintos a SQL.

Para asegurar que el árbol no pierda integridad, debe hacerse la revisión de que no queden sucesores de los registros eliminados.

Inclusión de la jerarquía en el identificador del registro.

El registro tiene varios campos para identificarlo. Uno para cada nivel dentro de la jerarquía, el primer registro en la jerarquía tiene un valor sólo en el primer campo de identificación y teniendo valores cero en todos sus demás campos. En los registros del segundo nivel los registros tendrán en su primer campo igual que el de su padre, y tendrán un valor de identificación en el segundo campo, con los campos restantes con ceros y así sucesivamente.

Árbol(Nivel_1, Nivel-2, Nivel-3, ..., Nivel-n, Otros-Campos)

Las ventajas de este esquema es que pueden identificarse todos los ancestros del registro y también sus descendientes no inmediatos. Esto simplifica ese tipo de consulta.

La desventaja es que la jerarquía tiene un número fijo de niveles, de tal forma que no es posible aumentar los niveles.

Su utilización es idónea cuando desde el análisis se conoce la profundidad de la jerarquía y se requiere concentración desde un nivel cualquiera a todos sus sucesores.

En el caso de la Entidad Nombre se puede generar una nueva entidad que presente este comportamiento, ya que el número de Categorías Taxonómicas es relativamente fijo. En la Figura 2 se muestran las dos situaciones. En la parte a) se tiene la asociación original, en la b) se tiene la solución propuesta, haciendo el comentario que sólo incluye algunas categorías taxonómicas (5), pudiendo agregar o reordenar la lista propuesta.

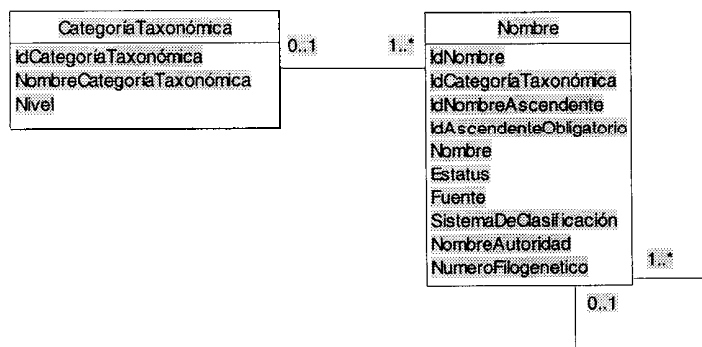


Figura 2 a). Situación Original.

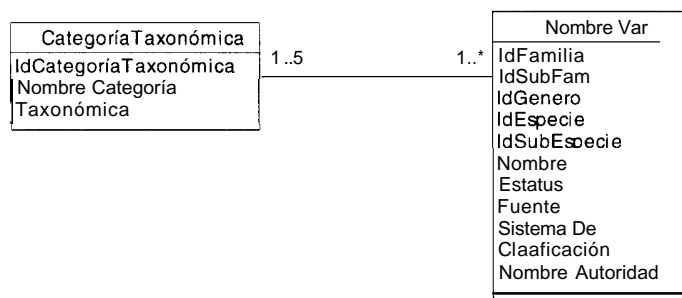


Figura 2 b). Situación Propuesta.

La estructura de la tabla (Nombre2 "Propuesto") está compuesta por:

Campos	Formato del Campo
Referencia	Numérico (Campo Llave de Referencia)
Nombre	Texto
N0	Numérico (Referencia de registros que se encuentran en nivel 0)
N1	Numérico (Referencia de registros que se encuentran en nivel 1)
.	.
.	.
.	.
N9	Numérico (Referencia de registros que se encuentran en nivel 9)
Nivel	Numérico (Nivel en que se encuentra el registro)

Donde el campo Referencia realiza la función de identificación de los registros en general, los campos NO, N1, ..., N9 son los niveles de los registros en ellos se encuentra las referencias de los registros que se encuentran en el nivel. y demás campos son datos complementarios.

Consultas

A) Consulta para obtener un Nodo y todos sus hijos.

```

SELECT      Nombre2.Referencia, Nombre2.Nombre, Nombre2.Nivel
FROM        Nombre2
WHERE       Nombre2.N0 = Padre-Nivel-0 AND
           Nombre2.N1 = Padre-Nivel-1 AND
           Nombre2.N2 = Padre-Nivel-2 AND
           Nombre2.N3 = Padre-Nivel-3;

```

Ventajas

Esta consulta es simple y se puede acceder rápido a todos sus sucesores, ya que, los hijos tendrán como prefijo la identificación del registro deseado. Esto se logra teniendo en cuenta en que nivel se encuentra el registro y con la ayuda de programación externa controlar las peticiones de los niveles o referencias.

Desventajas:

Es necesario saber en qué nivel se encuentra el registro.

B) Consulta para obtener un Nodo y todos sus antecesores.

```
SELECT      A2.Nombre, A3.Nombre, A4.Nombre, AS.Nombre, A6.Nombre,
            A7.Nombre, A8.Nombre, A9.Nombre, A10.Nombre, A11.Nombre,
            A1.Nivel
FROM        Nombre2 AS A1, Nombre2 AS A2, Nombre2 AS A3, Nombre2 AS A4,
            Nombre2 AS A5, Nombre2 AS A6, Nombre2 AS A7, Nombre2 AS A8,
            Nombre2 AS A9, Nombre2 AS A10, Nombre2 AS A11
WHERE       A1.Referencia= Referencia AND
            A2.Referencia=A1.N0 AND A3.Referencia=A1.N1 AND
            A4.Referencia=A1.N2 AND A5.Referencia=A1.N3 AND
            A6.Referencia=A1.N4 AND A7.Referencia=A1.N5 AND
            A8.Referencia= A1.N6 AND A9.Referencia=A1.N7 AND
            A10.Referencia=A1.N8 AND A11.Referencia=A1.N9;
```

Ventajas

Se puede acceder rápido a todos sus predecesores. Sólo es necesario conocer la valor del campo referencia del registro en cuestión. La consulta es más compleja pero no es necesaria realizar programación externa.

Desventajas

La consulta es un poco complicada en su escritura y no es eficiente en su procesamiento, pues implica hacer join's para todos los niveles del árbol de nombres.

C) Consulta para agregar un Nodo.

```
INSERT INTO Nombre2 ( Referencia, Nombre, N0, N1, N2, N3, N4, N5, N6, N7, N8,
                    N9, Nivel )
SELECT Teclee-la-Referencia, Nombre, Padre0, Padre1, Padre2, Padre3, Padre4,
       Padre5, Padre6, Padre7, Padre8, Padre9, Nivel;
```

Ventajas

Esta consulta es muy sencilla ya que sólo es necesario introducir la información referente al registro y tener certeza de existencia en la información de los campos de niveles o referencias, ya que la información nos indicará quién su antecesor.

Desventajas

Es necesario escribir una regla de integridad un tanto compleja o bien construir un Trigger para poder hacer las revisiones de integridad referencial hacia los padres.

D) Consulta para eliminar un Nodo y sus hijos.

```
SELECT      *
FROM        Nombre2
WHERE       Nombre2.N0= Deme_el_Padre 0 AND
            Nombre2.N1= Deme_el_Padre 1 AND
            Nombre2.N2= Deme_el_Padre 2 AND
            Nombre2.N3= Deme_el_Padre 3;
```

Ventajas

Esta consulta es simple y se puede eliminar más rápido a todos sus sucesores. Esto se logra tomando en cuenta en que nivel se encuentra el registro y con la ayuda de programación externa controlar las peticiones de los niveles o referencias.

Desventajas

Ninguna. Si no se desea eliminar un registro que tenga hijos, es necesario programar la regla de integridad a través de un Trigger

Conclusiones Respecto a los Aspectos de Diseño

En la primera opción la estructura de la tabla el registro tiene un campo que hace referencia hacia su padre, esto beneficia el acceso a predecesores o sucesores inmediatos, ya que se puede saber quién es su antecesor o quiénes son sus sucesores inmediatos, pero el problema radica en que se van complicando las consultas cuando se desea saber más allá de sus sucesores o antecesores, las consultas cada vez se van haciendo más complejas y menos fáciles de estructurar.

Las ventajas de este esquema es la de tener una estructura de n de niveles de tal forma que puedan incluirse nuevos niveles cuando sea necesario.

Por otro lado, en la segunda opción la estructura propuesta podría ser la mas aceptable por simpleza y eficiencia, ya que los registros tienen varios campos para identificar tanto a sus antecesores y sucesores. El primer registro en la jerarquía tiene un valor en el primer campo de identificación y teniendo valores cero en todos sus demás campos. En los registros del segundo nivel los registros tendrán en su primer campo igual que el de su padre, y tendrán un valor de identificación en el segundo campo, con los campos restantes con ceros y así sucesivamente. Esta estructura es recomendable, ya que puede conocerse de manera inmediata quiénes son sus sucesores o antecesores sin la necesidad de consultas complejas, el único inconveniente es la utilización de más espacio y datos redundantes, pero el desperdicio es mínimo con respecto al tamaño del registro (un 22% con respecto a un 8% del esquema original, haciendo la proporción del tamaño de la llave con respecto al tamaño de los datos). Es claro que en este esquema la jerarquía no podrá crecer en niveles más allá del número de campos que se dedicaron para ello. En el caso propuesto es nueve.

Por lo consiguiente se puede optar por la segunda opción, ya que las consultas son mucho más sencillas y el procesamiento de datos es aun más rápido. Esto compensa la utilización de datos redundantes y espacio físico en la estructura de la tabla. Además la estructura jerárquica no requiere un número variable en la profundidad del árbol de nombres.

Por otro lado cada nivel de jerarquía puede asociarse con la descripción correspondiente de acuerdo a la nomenclatura utilizada tales como reinos, especies, generos, familias, etc.

3. Aspectos de Forma

En esta sección se muestran todas las observaciones acerca de la presentación de la documentación relacionada con la Base de Datos, que serán importantes para la interpretación de la misma.

Presentación de los Diagramas

- Los diagramas no tienen ningún estándar para su elaboración lo que los hace difícil de leer para cualquier lector.
- El número de entidades que se presentan en el diagrama es excesivo (37 entidades), lo cual distrae al lector del diagrama, haciendo más complicada su interpretación, además de proponer más cruces de líneas de relación.

Propuesta

- Se plantea como estándar de diagramación la orientación de las entidades en función del lugar que ocupan en asociaciones de uno a varios, siendo las entidades que están en el extremo 1 colocadas en el lado izquierdo, quedando la del extremo varios en el lado derecho.
- El número de entidades en el diagrama no debe exceder más de 10 entidades. Esta situación tiene que ver con la percepción del lector de los diagramas. El número de objetos que un individuo puede retener en su memoria de corto plazo es de 7 ± 2 , teniendo.
- Cuando exista la situación de cruzar líneas para asociaciones, debe repetirse una de las entidades para evitar tales cruces. Mientras más simples las conexiones, es mejor.
- Ninguna asociación tiene nombre, lo cual dificulta la comprensión del diagrama para el lector que no está involucrado.
- No existen anotaciones que faciliten la comprensión del diagrama. Estas anotaciones pueden contener algunas de las Reglas de Negocio que contempla la aplicación dentro de la Base de Datos.

4. Aspectos de Optimización

En esta sección se describen aquellos aspectos que serán importantes para

- La entidad de Ejemplar tiene gran número de campos, además de ser la que tiene mayor número de registros. Esta situación obliga a que el número de registros por página sea muy reducido, aumentando el número de accesos a la tabla física. Esto implica tiempos de respuesta mayores para la consulta de la información de esta tabla.
- Existen campos que reflejan una marca en el tiempo (Año, Mes, Día), los cuales conviene se expresen en función de tipos fecha del manejador de base de datos utilizado.
- Algunos campos están utilizados por separado que no justifican tal separación. Ejemplos pueden ser los posición de un sitio, en los cuales se tienen por separado, incrementando la complejidad del modelo.

Propuesta

- Es necesario establecer un análisis de la frecuencia de utilización de ciertos campos de la entidad ejemplar, en el que se determinen los atributos que se utilizan con mayor frecuencia. Estos campos se dejarán en la entidad original, dejando los restantes en una entidad, la cual sea accedida en los casos que se requieran esos datos. La asociación entre estas dos entidades será de uno a uno.

Esta situación implica la modificación de los programas que utilizan la tabla Ejemplar, sin embargo, pueden utilizarse mecanismos como vistas o bien con **Triggers** para poder hacer esta actualización directa, reduciendo al mínimo el mantenimiento.

El tamaño del registro principal debe ser menor de 512 bytes, tratando en la medida de lo posible que este por los 100 bytes.

- En lo que respecta a los campos separados de Año, Mes y Día, es necesario explotar las funciones de manipulación de fechas, para poder integrarlos en un solo campo, optimizando espacio, el manejo en captura y su consulta en forma genérica y no especializada.
- Los campos que denoten cierta composición conviene integrarlos en uno solo y dejar a los programas de captura su manipulación para efectos de simplificar la estructura de la base de datos. Puede utilizarse un solo campo de posición que sea tipo cadena que integre todos los datos de la latitud y longitud así como altitud bajo un formato especificado para almacenamiento interno, aunque desde el punto de vista interfáz sean campos separados.

Anexo 1 Diagramas de la Base de Datos

En esta sección se muestran algunas propuestas de la separación del diagrama original en porciones más manejables. A continuación se muestra un diagrama de Paquetes conteniendo las distintas vistas o porciones de la Base de Datos, que da una percepción más estructurada de los distintos objetos estáticos de la aplicación.

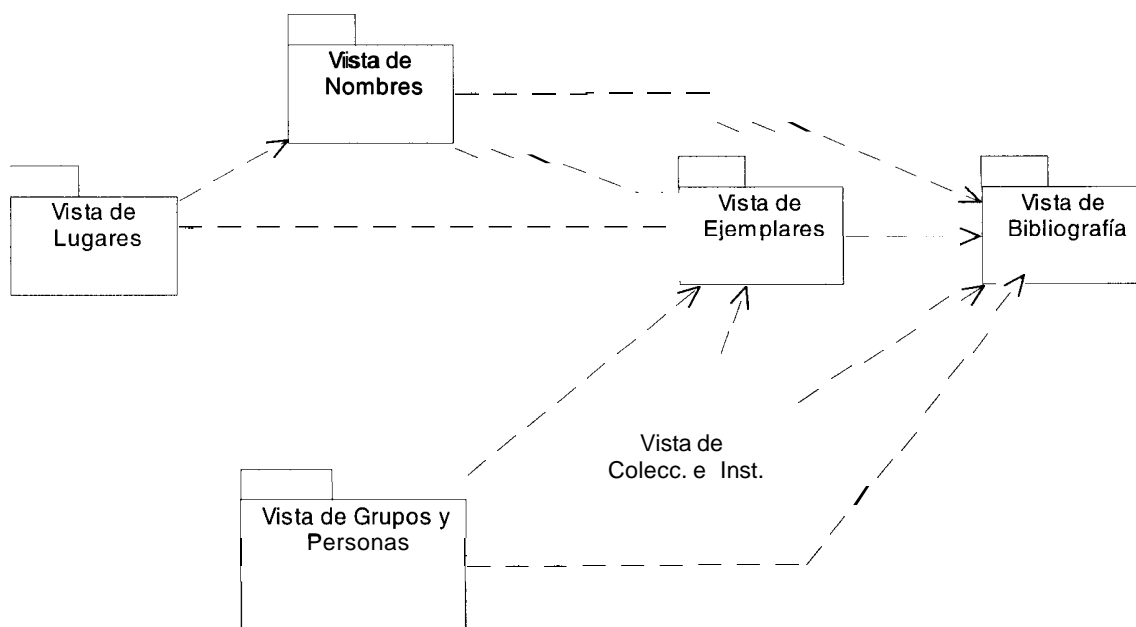


Figura 3. Diagrama de Bloques de la Base de Datos de Biótica.

Como puede observarse, existen distintos elementos que determinan a otros, indicándonos esta situación la secuencia de desarrollo de los programas de mantenimiento de los objetos de información.

Todos los diagramas presentan el estándar de diagramación aunque no incorporan las propuestas estructurales de diseño y optimización. Cuando en un diagrama se utiliza una Entidad que se definió en otro, se indica la referencia a que diagrama corresponde. Normalmente se indican todos los campos de todas las entidades, sin embargo, en algunos casos por restricciones de espacio se oculta el detalle de los campos.

Diagrama de Lugares

En este diagrama se muestran todos los elementos que corresponden a la información que servirá como referencia a un lugar en particular. Es de notarse que si esto se puede aislar existen elementos que pueden reutilizarse en otros proyectos distintos a Biótica.

Diagrama de Nombres

El diagrama de nombres presenta la clasificación taxonómica en la Base de Datos, tiene referencias directas e indirectas hacia sí misma. Los tiempos de consulta pueden ser grandes para poder extraer toda la información en una jerarquía por efectos de la navegación de acuerdo a la estructura actual.

Diagrama de Grupos y Personas

Establece los datos complementarios de personas con los ejemplares. Este diagrama muestra los elementos mínimos de la estructura de los grupos de investigación y las personas. Pueden ligarse los grupos a las instituciones, aunque en la versión actual no está tal asociación

Diagrama de Colecciones e Instituciones

Muestra la información de las colecciones y las instituciones que las poseen. Esta estructura permite ubicar la colección a la que pertenece el ejemplar. Se puede relacionar al grupo de trabajo.

Diagrama de Ejemplares

Constituye la información núcleo de Biótica y es donde se tendrá mayor número de registros. No toda la información tiene la misma frecuencia de uso. Falta la inclusión de los campos que corresponden a la asociación uno a muchos con respecto al ejemplar y la información relacionada con los grupos de trabajo relacionados en la asociación.

Diagrama de Bibliografía

En este diagrama se cubren los aspectos relacionados con la referencia bibliográfica. Algunas asociaciones deben analizarse para revisar si su utilización es significativa como para mantenerla dentro del esquema o

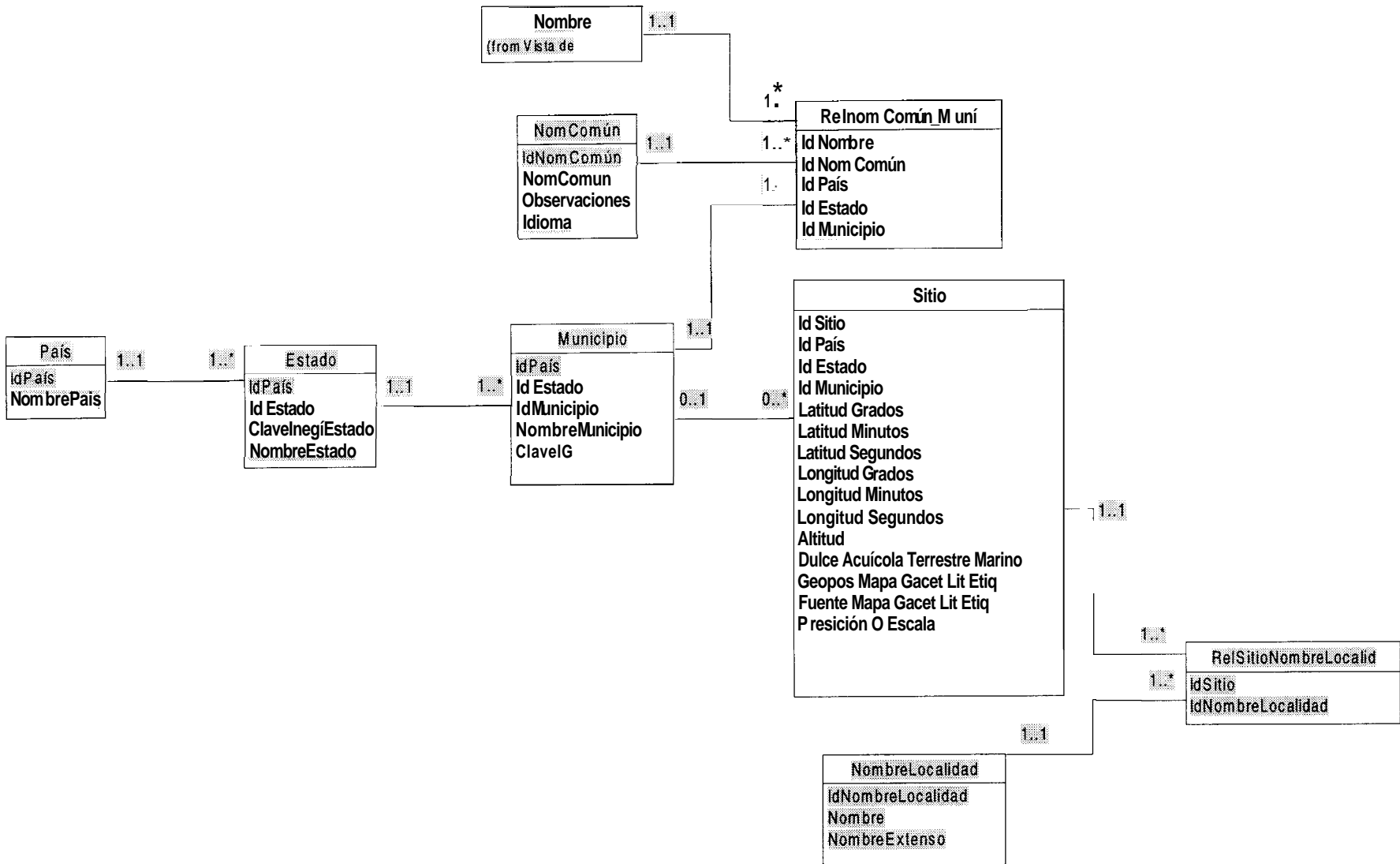


Figura 4. Diagrama de la Vista de Lugares.

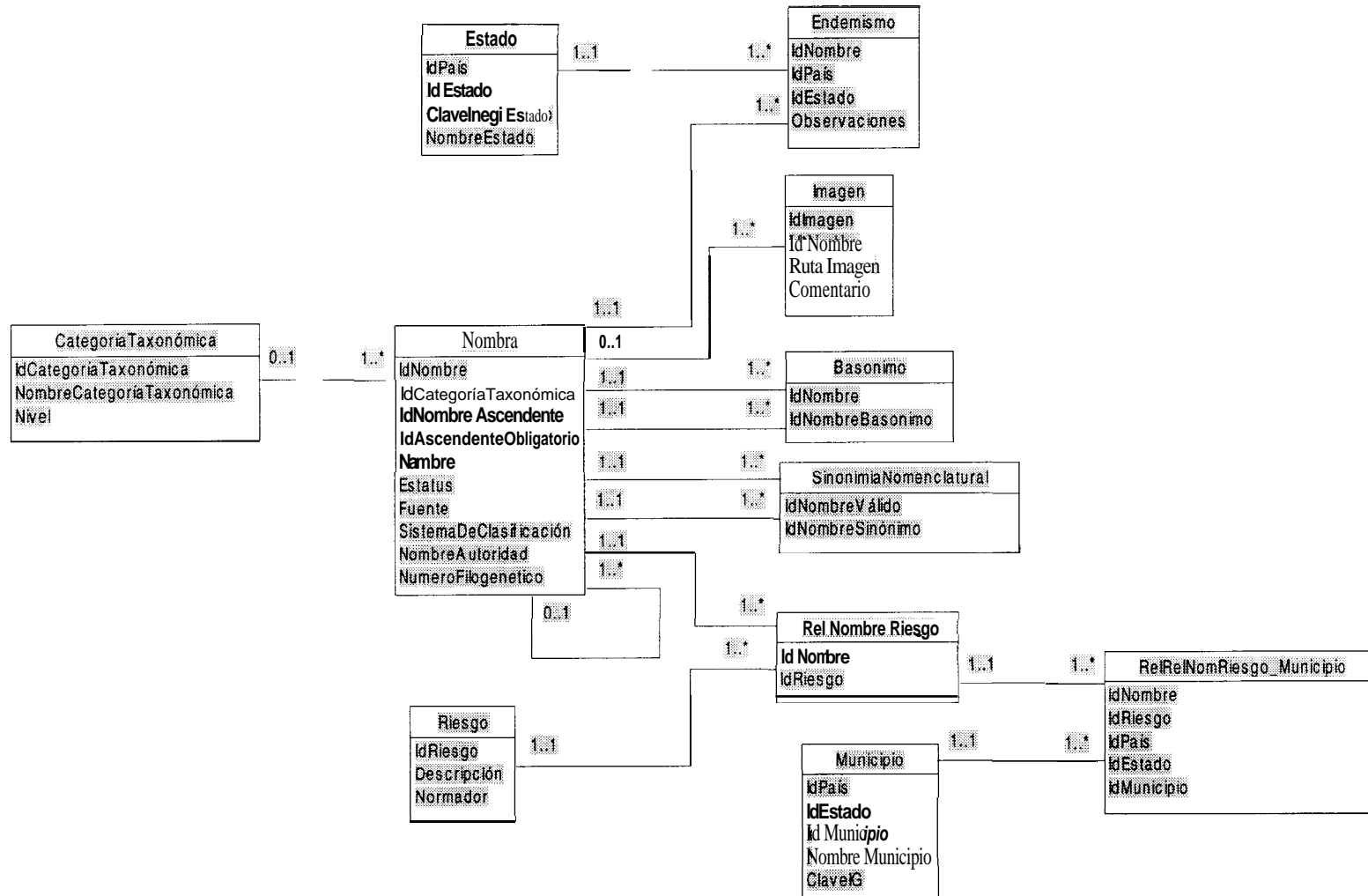


Figura 5. Diagrama de la Vista de Nombres.

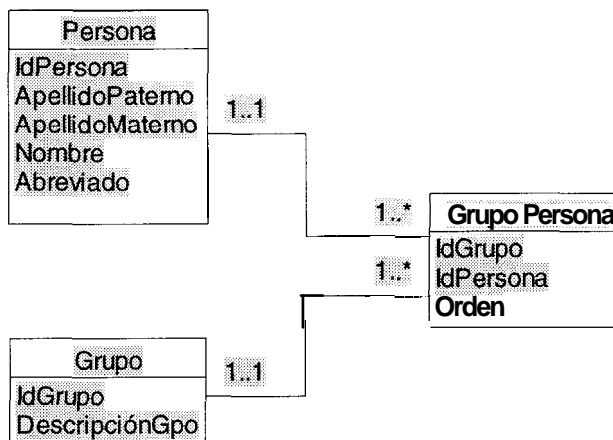


Figura 6. Diagrama de la Vista de Grupos y Personas.

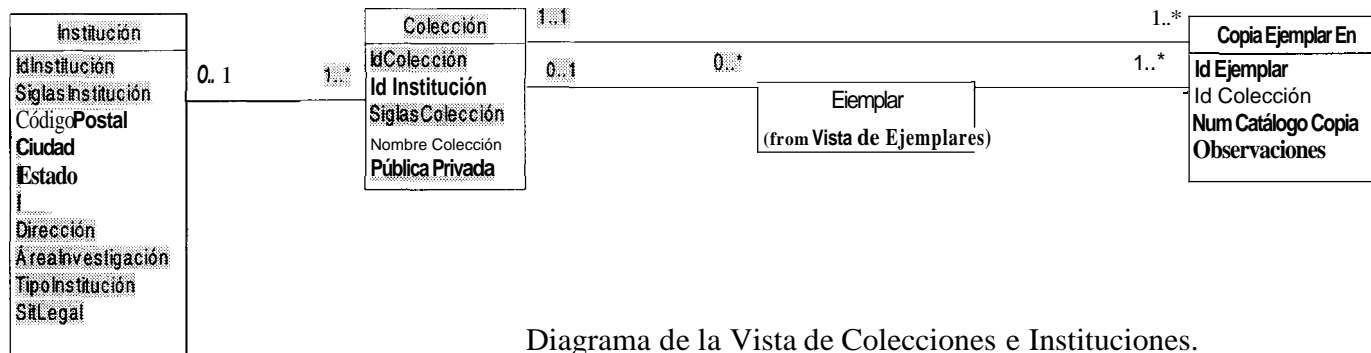


Figura 7.

Diagrama de la Vista de Colecciones e Instituciones.

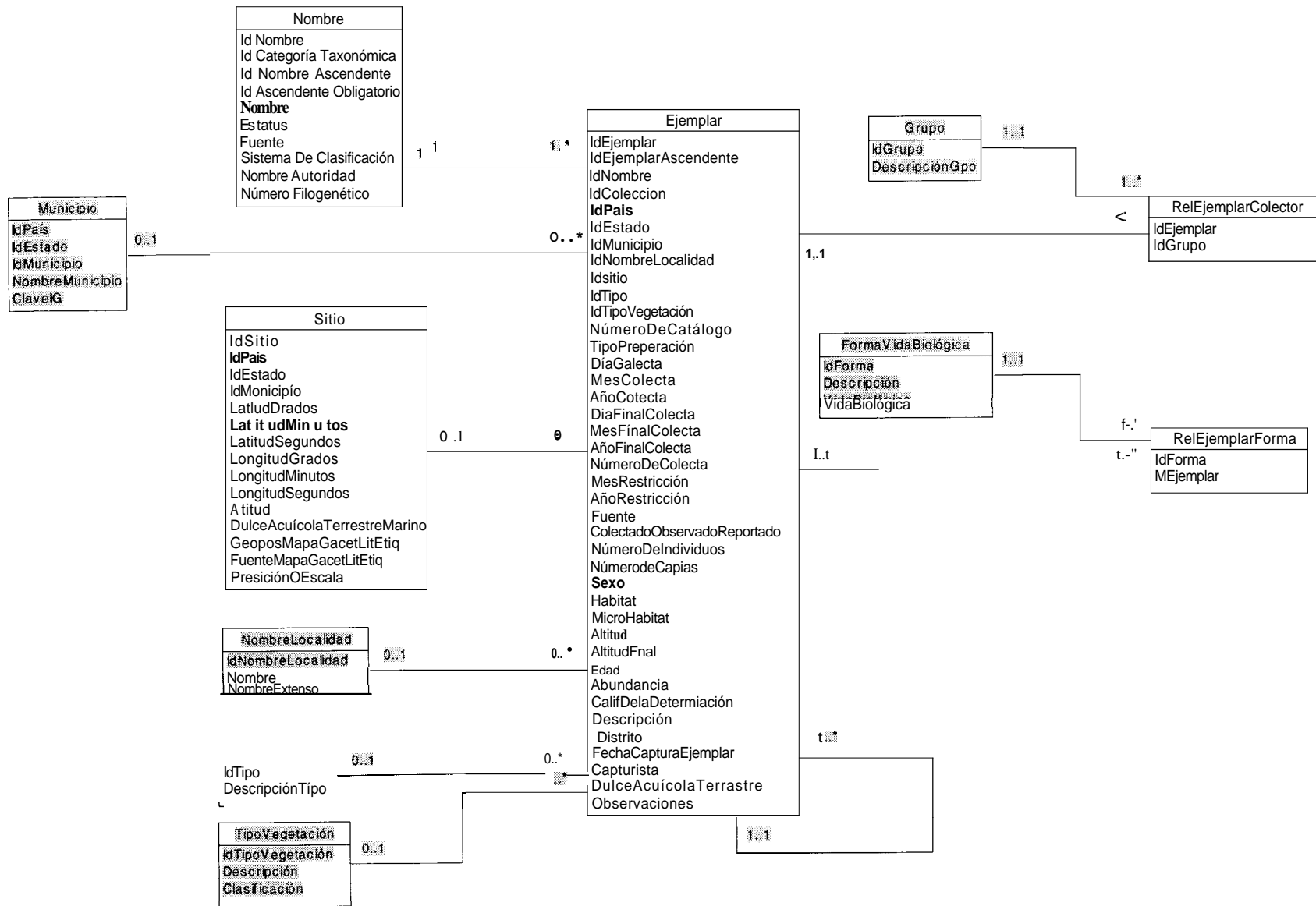


Figura 8. Diagrama de la Vista de Ejemplares.

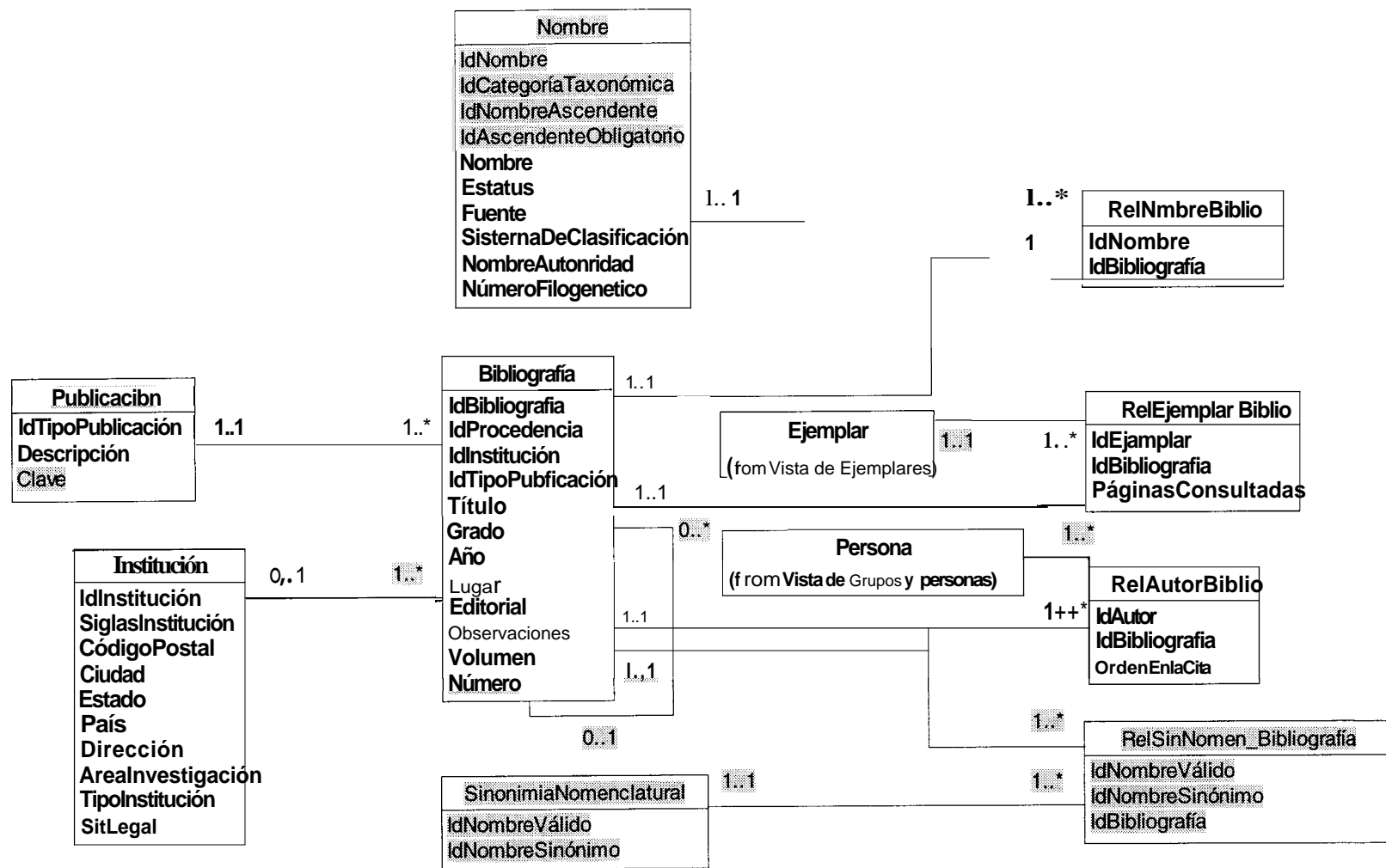


Figura 9. Diagrama de la Vista de Bibliografía.

Anexo 2 Observaciones a la Documentación de la Base de Datos de Biótica

En este Anexo se describen las observaciones realizadas a la 2ª parte de documentación proporcionada. Los comentarios se hacen referenciando a las páginas del manual.

Comentarios específicos al Documento

Página 2

Definición de llave primaria.- Otra característica a considerar es que la llave primaria debe ser óptima, es decir que si un atributo deja de participar en la llave primaria, la propiedad de identificación unívoca deja de cumplirse.

NA y "nulo".- Debe describirse como se hace en el anterior (Dato No Disponible).

Página 5

En el diagrama E-R inferior la entidad Municipio se considera como entidad dependiente y en realidad es una entidad independiente. Supongo que es una porción de la BD en donde es dependiente de País y de Estado, pero como la figura no las incluye, se puede prestar a confusión.

Página 6

Considero que es necesario hacer una descripción del Esquema General de Biótica para poder leer mejor el diagrama.

Página 7

Igual que en el caso anterior, es necesaria una breve descripción del propósito y estructuración de la información Nomenclatural. Esta observación es común para todas las secciones que describen un módulo.

Página 10

Es necesario un texto de introducción para describir la estructura de datos utilizada para describir el Nombre.

Página 16

Es necesario colocar una introducción al problema de sinonimia y basonimia.

Página 18, 19

Mismo comentario de página 7.

Página 20

Es necesario colocar una presentación a las reglas de negocio para el manejo de los tipos de publicación.

Página 22, 25

Mismo comentario de página 7

Página 25

Existe una simbología "rara" en la entidad Ejemplar, que no se comentó en la introducción. Debe ajustarse la entidad para que muestre todos sus atributos dentro de la figura.

Página 28

Mismo comentario de página 7 y página 25. Conviene que se muestre un ejemplo, tal y como se hizo en otros módulos, principalmente porque este es el módulo de mayor manejo operativo. La descripción de la relaciones entre ejemplares no se inserta muy bien pues no se sabe si forma parte del módulo curatorial o del de Directorio, precisamente por la no homogeneidad de los ejemplos.

Página 31

Mismo comentarios de página 28.

Página 35

Los ejemplos de relaciones deben tener una introducción para contextualizarlas.

Página 38 (Información de índices y reglas de integridad)

Necesitan describirse planteando una introducción, además indicar la necesidad dada por el volumen de datos de las tablas.

Comentarios generales

- Faltan una portada y una tabla de contenido.
- Falta una introducción, presentación y propósito del documento donde se describa la estructura y la audiencia del mismo.
- La información detallada de cada tabla puede colocarse en anexos así como la de los índices y reglas de integridad para no distraer la atención del lector en detalles.
- Faltan consideraciones de mantenimiento a la base de datos, tales como cuál información es recomendable que sea respaldada periódicamente o bien indicar las utilerías de respaldo que deben ejecutarse, cuándo y de qué manera.