



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
ACATLÁN

Diseño e implementación de un sistema de información vía web
para la administración de una parroquia

Tesina

QUE PARA OBTENER EL TÍTULO DE

Lic. en Matemáticas Aplicadas y Computación

PRESENTA

Ramón Apreza Zepeda

Asesor: Lic. Dyana Erika Marrón Luna

Enero de 2016

Santa Cruz Acatlán, Naucalpan, Estado de México



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A Dios, porque él me ha dado la vida,
me concedió unos padres maravillosos,
los cuales me han apoyado a lo largo de toda mi
vida y ahora es quien ha hecho posible la
finalización de mis estudios profesionales.

A mis padres y hermanos,
por apoyarme en cada paso que he dado en la vida.

A mi asesora por darme su confianza, su amistad y
guía a lo largo de la carrera y ahora en el proceso de
titulación.

A mis profesores, compañeros y alumnos,
por haberme enseñado algo nuevo
con sus palabras, acciones y omisiones.

TABLA DE CONTENIDO

INTRODUCCIÓN	iv
CAPÍTULO 1 INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE (SOFTWARE ENGINEERING) .	1
CAPÍTULO 2 PLANTEAMIENTO DEL PROBLEMA.....	20
CAPÍTULO 3 REQUERIMIENTOS DEL SISTEMA	23
CAPÍTULO 4 ESPECIFICACIÓN DEL SISTEMA.....	25
CAPÍTULO 5 DISEÑO DEL SISTEMA	66
CAPÍTULO 6 DESARROLLO DEL SISTEMA	70
CAPÍTULO 7 IMPLEMENTACIÓN DEL SISTEMA.....	93
CONCLUSIONES	97
ANEXO A: DIAGRAMAS ESTILO CHEN DEL CAPÍTULO 4.....	101
ANEXO B: DIAGRAMAS GENERADOS EN MYSQL WORKBENCH DEL CAPÍTULO 5	119
ANEXO C: GLOSARIO	135
ÍNDICE	142

INTRODUCCIÓN

El presente texto contiene la documentación del proceso que culminará con el desarrollo de un sistema de información utilizando la metodología de Ingeniería de Software.

El objetivo principal de este trabajo es utilizar la metodología de ingeniería de software presentada por Bagui (2011, traducción propia) en el desarrollo de un sistema de información que permita al personal de la parroquia Nuestra Señora de Guadalupe digitalizar el conjunto de expedientes y libros que conforman el archivo parroquial y cuando los feligreses soliciten algún servicio con respecto a la recepción de sacramentos, que ingresen nuevos registros, generen consultas y obtengan las actas correspondientes.

El ser miembro de la parroquia y contemplar la dinámica de trabajo del área administrativa de la parroquia, así como, el deseo de utilizar los conocimientos que adquirí en mi carrera para el bien común, me llevaron a desear apoyar a mi comunidad con esta herramienta.

El presente trabajo está compuesto de siete capítulos, a través, de los cuales se describe como se desarrolla un sistema de información, también, denominado aplicación multinivel (Gupta 2013, traducción propia), utilizando la metodología de Ingeniería de Software como lo describe Bagui (2011, traducción propia) tanto en el enfoque general como orientado al diseño de bases de datos.

El primer capítulo es un resumen de la presentación del proceso de Ingeniería de Software de acuerdo con Bagui (2011, traducción propia), acompañado de los conceptos necesarios para adentrarnos al diseño de software y en especial de bases de datos.

En el capítulo dos se expone el problema observado y se propone una herramienta de software como solución, indistintamente llamado sistema de información o aplicación. El sistema antes mencionado cae en la categoría de desarrollo web, por el uso de tecnologías orientadas a ser implementadas en un browser, ya sea utilizando recursos vía remota (internet) o de forma local.

En el capítulo tres se presentan las necesidades, deseos y dinámica actual de trabajo de los futuros usuarios del sistema de información.

En el capítulo cuatro se presenta la información proporcionada por los usuarios que será utilizada en el diseño de las bases de datos y la aplicación. Con respecto a las bases de datos se presentan los atributos que formarán parte de las tablas que a su vez constituirán las bases de datos utilizadas por el sistema. Con respecto al diseño general de la aplicación lo que el usuario espera poder realizar con él, es decir, controlar el acceso, que información podrá almacenar y buscar posteriormente, que documentos se podrán generar, etc.

En el capítulo cinco se procede a realizar el diseño general del sistema, el cual incluye el diseño de las bases de datos, el frontend o interfaz de interacción, y el conjunto de módulos que permiten las consultas a la base de datos y la generación de documentos, también

denominado backend (Gupta 2013, traducción propia). Con respecto a las bases de datos se hace uso de herramientas de software que permiten pasar los esquemas antes obtenidos a una representación gráfica que consideré los tipos de datos utilizados por el DBMS (Sistema de Administración de Bases de Datos) que se ha elegido.

Este capítulo se divide en dos partes, en la primera se hace mención de las herramientas de software a utilizar y en la siguiente se procede a describir como se utilizan dichas herramientas.

En el capítulo seis se describe como fue programado cada componente del sistema a lo largo de las secciones. Con respecto a las bases de datos se hace referencia a que los esquemas y las bases de datos acorde al modelo relacional obtenidos de la implementación de la metodología de Bagui (2011, traducción propia), será transformado en código ejecutable por el DBMS que se ha elegido.

En el capítulo siete se presenta la capacitación, pruebas y utilización del sistema de información por parte de los usuarios.

CAPÍTULO 1

INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE (SOFTWARE ENGINEERING)

1.1 INGENIERÍA DE SOFTWARE

Existe un sólo proceso de ingeniería de software, pero muchas perspectivas. Es agradable la de Bagui (2011, traducción propia) por ser breve y concisa, en ella resalta que en cuanto al término se denomina Ingeniería de Software al proceso de especificar, diseñar, escribir, entregar y finalmente retirar software. Los ingenieros de software frecuentemente se refieren al “ciclo de vida del software”.

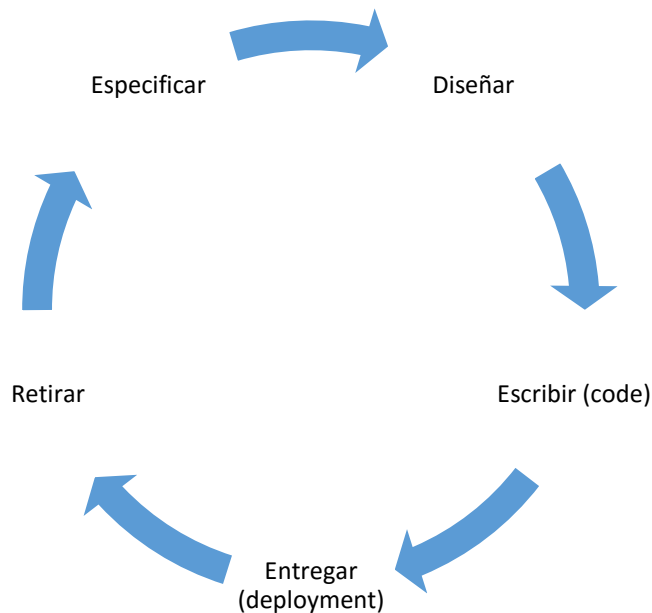


Figura 1.1 Ciclo de vida del software.

Cualquier sistema de información necesita un proceso para ser desarrollado correctamente. Para avanzar a través de este ciclo es necesario una serie de pasos o fases, los cuales garantizan que un pensamiento exhaustivo precede a toda acción. Este “pensar antes de actuar” tiene a su vez distintas representaciones, una de ellas es el modelo “cascada” (Schach, 2011 citado en Bagui 2011). Se hace tal comparación con el proceso de desarrollo de software por ir en una sola dirección y no retroceder. El modelo de cascada implica que una vez que la especificación del software está escrita y es aceptada por el usuario, ya no se cambia, sino que es usada como una base para el diseño.

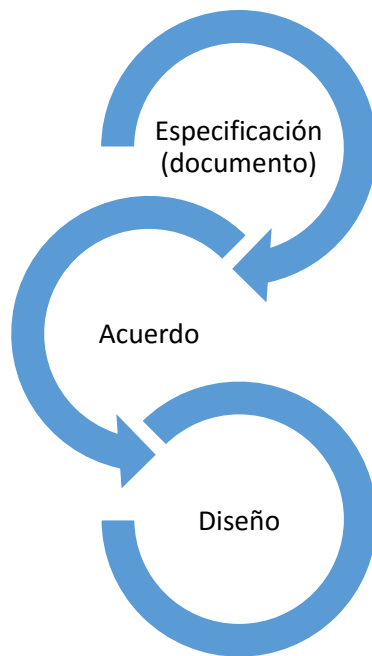


Figura 1.2 Modelo cascada que considera especificación, acuerdo y diseño

Aquellos que participan en el ciclo de vida de desarrollo de software caen en dos ámbitos el de analista y el usuario, el usuario es aquel que expresa sus deseos y necesidades al analista entonces el analista desarrolla software acorde a lo expresado por el usuario.

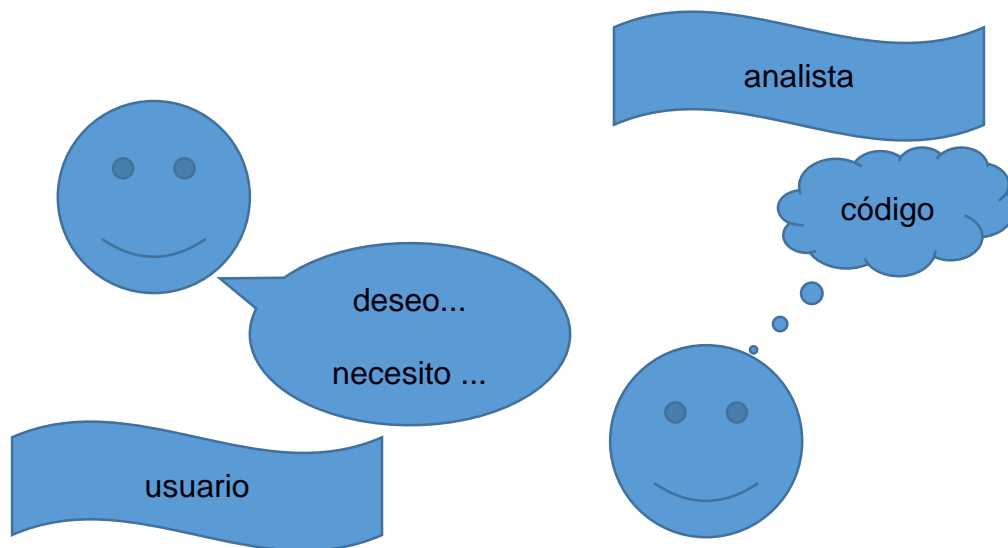


Figura 1.3 Jugadores dentro del proceso de ingeniería de software

Bagui señala que no hay un acuerdo general entre los ingenieros de software con respecto al número exacto de pasos o fases en el modelo de desarrollo tipo cascada. Los modelos varían dependiendo el interés que tenga el investigador de Ingeniería de Software en una parte u otra del proceso (traducción propia).

Después de haber hecho mención del proceso de ingeniería de software y algunos conceptos inherentes a él, Bagui (2011, traducción propia) proporciona los siguientes pasos como una de las tantas descripciones del mismo:

Paso1:

Requerimientos. Averiguar lo que el usuario quiere o necesita. El “procedimiento de averiguar” es frecuentemente llamado “clarificación”.

Paso2:

Especificación. Escribir lo que el usuario quiere o necesita de la forma más precisa posible. En este paso, el usuario y el analista documentan no sólo lo que se desea sino también cuánto costará y cuánto tiempo será necesario. Un credo de la Ingeniería de Software es generar software a tiempo y acorde al presupuesto.

Paso 2a:

Regresar la especificación al usuario para recibir su **retroalimentación**. Una revisión formal del documento de especificación es realizada para ver si el analista ha entendido bien lo que se quiere o necesita.

Paso 2b:

Rehacer la especificación tanto como sea necesario y regresar al paso 2a hasta que el analista y el usuario se entiendan y decidan proseguir con los pasos siguientes”.

Paso 3:

Diseñar –el software es diseñado de forma acorde a la especificación del paso anterior. . .

Paso 3a:

El diseño de software es **revisado de forma independiente con respecto a la especificación**. Si es necesario el diseño es reparado o rehecho hasta que el analista claramente cumpla la especificación. Hay que notar el sentido de acuerdo en el paso de especificación y el uso de este como una base para una acción posterior. Cuando la etapa de diseño comienza, hacer retroceder la cascada es difícil; y se supone que debe ser así. Tal vez detalles menores sobre la especificación pueden ser revisados, pero la idea es avanzar al paso siguiente hasta terminar. Cuando la revisión ha finalizado, ambos tanto el usuario como el analista saben lo que se ha hecho. . .

Paso 4:

Desarrollo. El software es programado (o escrito).

Paso 4a:

En la fase de desarrollo, **el software, a medida que es escrito es comparado con el diseño hasta que el analista claramente cumpla con el diseño**. Hay que notar que la especificación hace mucho que paso, y sólo pequeñas modificaciones del diseño podrían ser toleradas aquí. En la etapa de desarrollo lo más importante es construir el software de acuerdo al diseño . . .

Paso 5:

Implementación. El software es entregado al usuario para que sea utilizado.

Paso 5a:

El usuario prueba el software y lo acepta o lo rechaza hasta que se programado correctamente (eso significa, hasta que cumpla con la especificación y el diseño) . . . Una persona puede pensar que este es el final del ciclo de vida del software, pero hay dos etapas importantes que faltan.

Paso 6:

Mantenimiento. El mantenimiento es realizado en el software hasta que es retirado. No importa que tan bien este especificado, diseñado y programado algunas partes del software podrían fallar. Algunas partes podrían necesitar modificaciones al paso del tiempo para adecuarse al usuario. Los tiempos cambian y las demandas y necesidades también. El mantenimiento es una parte del proceso de software que es cara y consume mucho tiempo (particularmente si el proceso de Ingeniería de Software no se hizo bien). El mantenimiento envuelve corregir fallas ocultas en el software, así como, mejorar su funcionalidad.

En bases de datos, nuevos datos son frecuentemente requeridos; algunos datos viejos puede que ya no sean necesarios. El hardware cambia. Los sistemas operativos cambian. El motor de la base de datos es con frecuencia mejorado –nuevas versiones son impuestas en el mercado.

Paso 7:

Retiro. Eventualmente, cualquier software que ha sido programado llega a ser obsoleto. Los motores de bases de datos, las computadoras y la tecnología en general están evolucionando . . . Básicamente, el proceso de Ingeniería de Software tiene que comenzar de nuevo con nuevas especificaciones. Lo mismo ocurre con respecto a las bases de datos y sistemas diseñados. Hay ocasiones en las que lo más económico es comenzar de nuevo. (Bagui 2011, traducción propia)

Los pasos anteriormente presentados de acuerdo con Bagui (2011, traducción propia), son fundamento de este trabajo; tanto en el diseño general de la aplicación como en el diseño de las bases de datos del mismo, ya que son utilizadas por Bagui para desarrollar su enfoque

de diseño de bases de datos basados en diagramas Entidad-Relación, el cuál es protagonista de la siguiente sección.

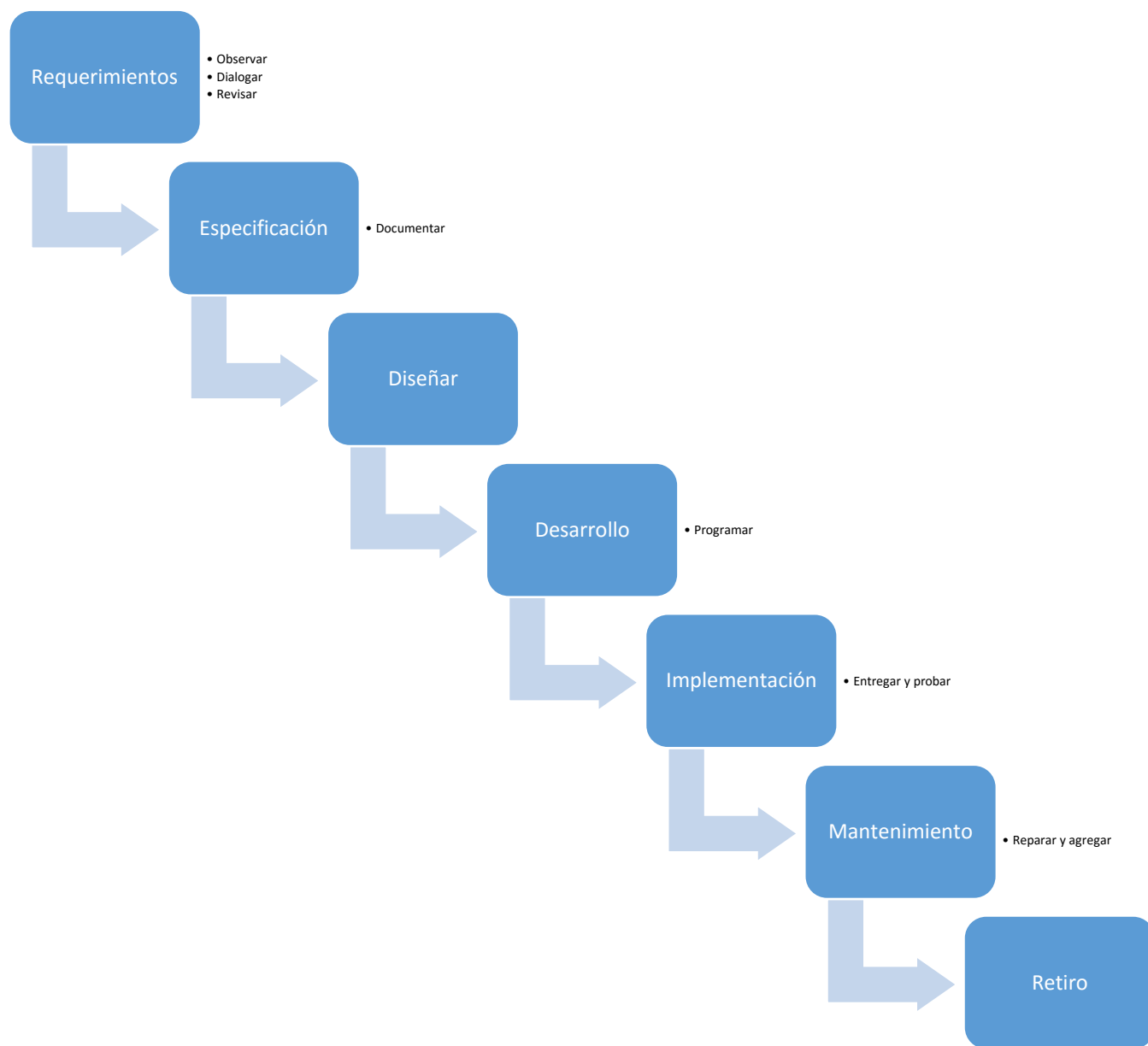


Figura 1.4 Ingeniería de software según Bagui (2011)

1.2 INGENIERÍA DE SOFTWARE APLICADA AL ENFOQUE DE DISEÑO DE BASE DE DATOS UTILIZANDO DIAGRAMAS ENTIDAD-RELACIÓN

Antes de presentar la metodología de Bagui basada en la ingeniería de software y aplicada al diseño de base de datos, hay algunos conceptos que Bagui antepone para una mejor comprensión de su metodología y para mejor comprensión a lo largo del documento.

“Se considera un **modelo conceptual** a la descripción abstracta de los datos” (Bagui 2011, 68).

“Un **esquema de modelado de datos** es un método que nos permite modelar o representar una base de datos “(Bagui 2011, 67).

“Un **diagrama Entidad-Relación** es una herramienta gráfica que facilita el modelado de datos” (Bagui 2011, 67), es decir, abstractamente describir o representar los datos.

“Los diagramas Entidad-Relación son un subconjunto de los 'modelos semánticos' en la jerga de las bases de datos. Los modelos semánticos se refieren a modelos cuya intención es clarificar el significado de los datos” (Bagui 2011, 67).

“**Mapeo** es el proceso de elegir un modelo lógico y entonces pasar de un modelo conceptual . . . aun sistema de archivos de bases de datos físico . . . El mapeo es el puente entre el concepto de diseño y la realización física” (Bagui 2011, 68, traducción propia).

“Un **esquema** implica una descripción fija y permanente de la estructura de los datos” (Bagui 2011, 68, traducción propia).

El estilo de diagrama Entidad-Relación que se utiliza es el estilo Chen, dado a que es el estilo utilizado por Bagui en su enfoque de diseño de bases de datos. Algunas razones por las que Bagui (2011, xxvii, traducción propia) lo escogió son que Chen fue el primero en introducir la idea de los diagramas Entidad-Relación (Elmarsy y Navathe 2007, citado en Bagui 2011, traducción propia), en los libros más completos se utiliza, y en otros se utilizan variantes, no necesita considerar su implementación, es decir, su modelo lógico, en comparación a otros estilos de diagramas, como el estilo Barker u Oracle que están muy apegados al paradigma de base de datos relacional.

A continuación, se presenta la **metodología de Bagui con respecto al diseño de base de datos partiendo de un diagrama Entidad-Relación en su versión original y versión mejorada**.

Paso 1. Se selecciona una entidad principal, de la descripción de requerimientos de la base de datos y se muestra los atributos a ser registrados para esa entidad. Se etiquetan las claves si es apropiado y se proporcionan algunos datos de muestra.

Paso 2. Se usa inglés estructurado (en este caso español) para las entidades, atributos y claves para describir la base de datos como ha sido obtenida.

Paso 3. Se examinan los atributos en las entidades existentes (de preferencia con

asistencia del usuario) para descubrir, si información extra sobre alguna de ellas debe ser registrada. . .

Paso 3a. Si la información sobre un atributo es necesaria, entonces se convierte el atributo en una entidad y:

Paso 3b. Se definen las relaciones de vuelta a la entidad original.

Paso 4. Si otra entidad es apropiada, se dibuja la segunda entidad con sus atributos. Entonces repite los pasos 2 y 3 para ver si la entidad debería ser dividida en más entidades.

Paso 5. Se conectan las entidades con relaciones (una o más) si es que alguna relación existe.

Paso 6. Se indica la naturaleza exacta de las relaciones en inglés estructurado desde todos los extremos (lados), por ejemplo, si una relación es A:B::1:M, entonces hay una relación de A(1) hacia B(M) y de B(M) hacia A(1).

Para relaciones ternarias y de orden mayor (n-arias), se indica la relación en inglés estructurado, siendo cuidadoso de mencionar todas las entidades que participan en la relación n-aria. Se indican las restricciones estructurales tal como están.

Para relaciones de especialización/generalización, se indica la relación en inglés estructurado siendo cuidadoso de mencionar todas las entidades (subclases o especializaciones). Se indican las restricciones estructurales tal como están.

Paso 6a. Se examina la lista de atributos y se determina si alguno de ellos necesita ser identificado por dos (o más) entidades. Si es así, se coloca el atributo en la relación apropiada que conecte las dos entidades.

Paso 6b. Se examina el diagrama en busca de ciclos, los cuales podrían indicar relaciones redundantes. Si una relación es verdaderamente redundante hay que quitar la relación.

Paso 7. Se presentan algunos datos muestra.

Paso 8. Se presenta la base de datos “tal como está diseñada” al usuario junto con el inglés (en este caso español) correspondiente a las entidades, atributos, claves y relaciones. Se refina el diagrama tanto como sea necesario. (Bagui 2011, 281, traducción propia)

La mención de la versión mejorada o EER (Enhanced Entity Relationship) es porque Bagui (2011, traducción propia) lo introduce en los capítulos finales para considerar la presencia de Generalizaciones y Especializaciones (en jerga de programación orientada a objetos, Jerarquía de clases).

Esta metodología es acompañada de gramáticas, reglas de mapeo y consejos de normalización, para obtener una base de datos lo más acorde al modelo lógico relacional y obtener un esquema después de utilizarla, las cuales serán presentadas en las secciones siguientes.

1.2.1 Gramáticas

Las siguientes gramáticas son proporcionadas por Bagui (2011) a lo largo de su libro, que permiten otorgar al usuario una explicación clara de lo que los diagramas representan, obteniendo una retroalimentación y sugerencias por parte de él. Se ha procurado recogerlas a lo largo de los capítulos de su obra y en algunos casos hacerle ciertas adaptaciones a nuestro lenguaje, así como, ciertas generalizaciones.

1.2.1.1 Entidad

“Esta base de datos registra datos sobre Entidad. Por cada Entidad en la base de datos, se registra $att(1)$, $att(2)$, $att(3)$, ... $att(n)$ ”(Bagui 2011, 83, traducción propia).

1.2.1.2 Entidad Intersección

Esta base de datos registra datos sobre la combinación de la Entidad1-Entidad2: Entidad1+Entidad2. Por cada Entidad1+Entidad2 en la base de datos, se registra un $att(j)$ (Bagui 2011, 198, traducción propia).

1.2.1.3 Atributos

Bagui (2011, 84) proporciona gramáticas que consideran distintos tipos de atributos, a continuación, los primeros:

Para atributos atómicos, $att(j)$:

Por cada Entidad, habrá siempre solamente un $att(j)$. El valor para $att(j)$ no será subdividido.

Para atributos compuestos, $att(j)$:

Por cada Entidad, se registra $att(j)$, el cual está compuesto de x, y, z, \dots (x, y, z) son las partes componente de $att(j)$.

Para atributos de valores múltiples, $att(j)$:

Por cada Entidad, se registra $att(j)$'s. Podría haber más de un $att(j)$ registrado por cada Entidad.

Para atributos derivados, $att(j)$:

Por cada Entidad, podría existir $att(j)$'s, los cuales se derivarán de la base de datos. (Traducción propia)

Además de estos atributos que están presentes en la mayoría de bases de datos, hay otros que surgen ante la presencia de entidades especiales, la metodología de Bagui procura contemplarlos, por lo cual en otros capítulos del libro se encuentran estos tipos de entidades y sus gramáticas, las cuales resumo y presento a continuación:

Atributos de una entidad intersección

Por cada combinación Entidad1+Entidad2, habrá solamente un att(j). El valor para att(j) no será subdividido (Bagui 2011, 198, traducción propia).

En caso de muchos atributos simples:

Cada Entidad tendrá solamente un att(1), att(2), att(3), ... att(n). Ninguno de estos atributos será subdividido (Bagui 2011, traducción propia).

Atributos de relaciones

En este tipo de situaciones se antepondrá lo siguiente a las gramáticas ya presentadas:

“Para la relación entre ENTIDAD1 y ENTIDAD2, se registra un att(j). El att(j) depende de ambas entidades ENTIDAD1 y ENTIDAD2 para identificación” (Bagui 2011, 195, traducción propia).

1.2.1.4 Las claves

(a) más de una clave candidata (**entidad fuerte**):

“Por cada Entidad, se tendrán las siguientes claves candidatas: att(j), att(k), ... (donde j, k son atributos). Una clave es elegida como clave principal” (Bagui 2011, 84, traducción propia).

(b) una clave candidata (**entidad fuerte**):

“Por cada Entidad, se tendrá la siguiente clave principal: att(j). La clave principal es la clave candidata” (Bagui 2011, 84, traducción propia).

(c) una clave candidata (**entidad débil**):

“Por cada Entidad débil, se asume que ninguna entidad débil será registrada sin su correspondiente entidad propietaria (entidad fuerte). Por cada entidad débil que tenga una clave principal, se debe registrar la clave principal de su entidad propietaria” (Bagui 2011, 181, traducción propia)

(d) sin claves candidatas (**entidad débil**):

“Por cada entidad débil, no se asume que algún atributo será lo suficientemente único para identificar entidades individuales” (Bagui 2011, 182, traducción propia).

“Dado que la entidad débil no tiene una clave candidata, cada entidad débil será

identificada por una clave o las claves que pertenezcan a su entidad propietaria” (Bagui 2011, 182, traducción propia).

(d) sin claves candidatas (**entidad intersección**):

“La clave de la entidad intersección consistirá de la concatenación de las claves foráneas de las entidades propietarias” (Bagui 2011, 198, traducción propia).

1.2.1.5 Relaciones binarias

Patrón 1 – x:y::k:1

“En el lado de k, participación total (k=1 o M): Las x’s, que son registradas en la base de datos, **deben** estar relacionadas a una y solamente una y. Ninguna x está relacionada a más de una y”(Bagui 2011, 132, traducción propia).

Patrón 2 – x:y::k:1

“En el lado de k, participación parcial (k=1 o M): x, pero no necesariamente todas las x (que son registradas en la base de datos) **podrían** estar relacionadas a una y solamente una y. Algunas x’s no están relacionadas a una y. Las x’s **podrían** no estar relacionadas a más de una y”(Bagui 2011, 133, traducción propia).

Patrón 3 – x:y::k:M

“En el lado de k, participación total (k=1 o M): Las x’s que están registradas en la base de datos, **deben** estar relacionadas a muchas (una o más) y’s. Algunas veces es útil incluir una frase como: 'Ninguna x está relacionada con algo distinto a una y' o 'Nada que no sea una x no está relacionada a una y.' la forma negativa dependerá del sentido de la oración” (Bagui 2011, 133, traducción propia).

Patrón 4 – x:y::k:M

“En el lado de k, participación parcial (k=1 o M): x, pero no necesariamente todas las x (que están registradas en la base de datos), **podrían** estar relacionadas a muchas (una o más) y’s. Algunas x podrían no estar relacionadas a una y” (Bagui 2011, 134, traducción propia).

1.2.1.6 Relaciones n-arias

“La gramática para una relación n-aria debe envolver todas las entidades que están relacionadas. Por lo tanto, una forma informal adecuada podría ser algo así”:

“ENTIDAD1 relación (de/a/por) ENTIDAD2 (y)(de/a/por) ENTIDAD3. Se entiende que el atributo necesitará nombrar las “n” entidades para ser identificado” (Bagui 2011, 237, traducción propia).

“Una gramática más exacta para la relación n-aria será una extensión de la desarrollada para relaciones binarias, que podría implicar tres oraciones para una relación ternaria, una comenzando por cada entidad” (Bagui 2011, 238, traducción propia).

En el caso de las relaciones n-arias o de grado n, que superan las tres entidades es necesario una gramática general más estricta, la cual Bagui (2011, 240) define así:

Patrón 5 (n-aria) – x:y:z::a:b:c, en el lado de a, participación total o parcial:

x, que son registradas en la base de datos deben estar relacionadas a

b=m (muchas (una o más))y

b=1 una y solamente una y

y (u otra palabra de enlace apropiada (de, desde, por, a, ...))

c=m (muchas (una o más)) z

c=1 una y solamente una z

Ninguna x está relacionada a más de una z.

Ninguna x está relacionada a más de una y. (traducción propia)

Este se podría considerarse tanto sólo un template estrechamente ligado a tres entidades, pero es tarea de quien utiliza la metodología extender al número de entidades que están vinculadas en la relación.

1.2.1.7 Relaciones Generalización/Especialización

El diagrama entidad relación presentado por Chen (1976, citado en Bagui 2011, traducción propia) tuvo que ser modificado para considerar situaciones semejantes a la jerarquía de clases en el ámbito de programación orientada a objetos de tal forma que pudiera manejar generalizaciones y especializaciones, su nombre es diagrama Entidad-Relación Mejorado o EER por sus siglas en inglés (Elmars and Navathe 2007, citado en Bagui 2011), a continuación, la gramática para estas situaciones:

Por cada especialización, no se asume que algún atributo será lo suficientemente único para identificar entidades individualmente. Ya que la especialización no tiene una clave candidata, cada especialización será identificada por una clave o claves heredadas de la generalización.

Además, las especializaciones se sobreponen (o son disjuntas). Se explica la situación de sobreposición o disjunción.

La especialización individual es identificada por un predicado definición, -nombre del atributo-, el cual se encuentra en la generalización. Si las especializaciones se sobreponen, el predicado definición será de valores múltiples. Se enumeran las especializaciones. (Bagui 2011, 263, traducción propia)

Es necesario conocer este tipo de gramáticas, ya que ante el creciente número de aplicaciones que necesitan utilizar bases de datos se podría tener que tratar este tipo de situaciones (Elmars and Navathe 2007, citado en Bagui 2011). Tal es la situación que Bagui

(2011) dedica un capítulo al diagrama EER para atender estas y otras situaciones en el diseño de bases de datos.

1.2.2 Reglas de mapeo

Anteriormente se había mencionado que la metodología de Bagui (2011, 287) permite concluir con un esquema y una base de datos relacional, eso gracias a los esfuerzos de Bagui por presentarnos las siguientes reglas de mapeo:

1. Mapeo de entidades fuertes. Desarrollar una nueva tabla (relación) por cada entidad fuerte y hacer la clave indicada de la entidad fuerte la clave principal de la tabla. Si se ha indicado más de una clave candidata en el diagrama ER, escoger una como clave principal.

2. Mapeo de atributos atómicos. Para entidades con atributos atómicos, se mapean las entidades a una tabla y se forman columnas por cada atributo atómico.

3. Mapeo de atributos compuestos. Para entidades con atributos compuestos, se mapean las entidades a una tabla y se forman columnas de cada parte principal (atómica) del atributo compuesto.

4. Mapeo de atributos de valores múltiples. Formar una tabla separada para el atributo de valores múltiples. Registrar un renglón por cada valor del atributo de valores múltiples junto con la clave de la tabla original. La clave de la nueva tabla será la concatenación del atributo de valores múltiples más la clave de la entidad propietaria. Se remueve el atributo de valores múltiples de la tabla original.

5. Mapeo de relaciones M:N. Por cada relación M:N, se crea una tabla (relación) con las claves principales de cada una de las dos entidades (entidades propietarias) que están siendo relacionadas en la relación M:N. La clave principal de esta nueva tabla serán las claves concatenadas de las entidades propietarias. Incluir todos los atributos que la relación M:N podría tener en esta nueva tabla.

6. Mapeo de relaciones binarias 1:1 cuando un lado de la relación tiene participación total y el otro tiene participación parcial. Cuando uno de los lados de la relación tiene participación total y el otro tiene participación parcial, entonces colocar la clave principal del lado con restricción de participación parcial en el lado con la restricción de participación total; este atributo es una clave foránea (la cual no está subrayada). Incluir todos los atributos sobre la relación en la misma tabla en la que se agregó la clave.

7. Mapeo de relaciones binarias 1:1 cuando ambos lados tienen restricciones de participación parcial.

7a. Seleccionar cualquiera de las tablas para almacenar la clave de la otra como clave foránea.

7b. Dependiendo de la semántica, se puede crear una nueva tabla para guardar la relación que podría contener la clave de las dos entidades relacionadas.

8. Mapeo de relaciones binaria 1:1 cuando ambos lados tienen restricciones de participación total. Usa la semántica de la relación para seleccionar cuál de las tablas deberían contener la clave de la otra. Si esta elección no es clara, entonces usa la regla de mapeo 7b.

9. Mapeo de relaciones binarias 1:N cuando el lado N tiene participación total. Incluir la clave de la entidad en el lado 1 como clave foránea en el lado N.

10. Mapeo de relaciones binarias 1:N cuando el lado N tiene participación parcial. Esta situación podría ser manejada como la relación binaria M:N con una tabla separada para la relación. La clave de la nueva tabla consistiría de la concatenación de las claves de las entidades relacionadas. Incluir todos los atributos que estaban en la relación en esta nueva tabla.

11. Mapeo de entidades débiles. Desarrollar una nueva tabla (relación) para cada entidad débil. Como en el caso de la entidad fuerte, incluir todos los atributos atómicos de la entidad débil en la tabla. Si hay un atributo compuesto, incluir sólo las partes atómicas del atributo compuesto y asegurarse de calificar las partes atómicas para no perder información. Para relacionar la entidad débil a su propietaria, incluir la clave principal de la entidad propietaria en la tabla de la entidad débil. La clave principal de la entidad débil será la clave parcial en la entidad débil concatenada con la clave principal de la entidad propietaria.

12. Mapeo de relaciones recursivas 1:N. Reincluir la clave principal de la tabla con la relación recursiva en la misma tabla, dándole otro nombre de rol.

13. Mapeo de relaciones recursivas M:N. Crear una tabla separada para la relación como en la regla 5.

14. Mapeo de relaciones n-arias. Por cada relación n-aria, crear una nueva tabla. En la nueva tabla, incluir las claves de las entidades conectadas y todo atributo de la relación. Hacer la concatenación de las claves de las entidades conectadas la clave principal de la nueva tabla.

15. Mapeo de generalizaciones y especializaciones con subclases disjuntas o superpuestas y con restricciones de participación total o parcial (con pocos o muchos atributos en las especializaciones). Por cada situación de entidad generalización/especialización, crear una tabla por cada especialización. Agregar los atributos por cada especialización a sus respectivas tablas (relaciones). Incluir la clave principal de la entidad generalización en las tablas especialización. La clave principal

de las tablas especialización sería la misma clave principal como la tabla generalización.

16. Mapeo de generalizaciones y especializaciones con restricciones de relación disjunta y participación total entre generalizaciones y especializaciones. Crear una tabla separada (subclase) por cada entidad especialización. Incluir los atributos por cada entidad especialización en sus respectivas tablas subclase. La clave principal de las tablas subclase será la clave principal de la entidad generalización.

17. Mapeo de generalizaciones y especializaciones con relaciones disjuntas, restricciones de participación total o parcial, predicado definición con atributos de tipo simple. Crear una única tabla que incluye todos los atributos de la generalización (superclase), así como, los atributos de las especializaciones (subclases). La clave principal de la tabla será la clave principal de la generalización (superclase).

18. Mapeo de relaciones de superposición y generalizaciones/especializaciones con más de una bandera. Crear una única tabla que incluye los atributos de la generalización (superclase) y los atributos de las especializaciones o subclases (incluyendo las banderas de subclase). La clave principal de la tabla es la clave principal de la superclase.

19. Mapeo de subclases compartidas (subclases que heredan de más de una subclase). El mismo criterio que es utilizado para determinar cuál regla sería la mejor para mapear generalizaciones y especializaciones puede ser aplicado para el mapeo de subclases compartidas.

20. Mapeo de categorías o tipos unión (subclases que heredan de más de una superclase) cuando las superclases tienen las mismas claves principales. Crear una nueva tabla para la subclase (o tipo unión) e incluir la clave principal de la superclase o superclases en la subclase (o tipo unión) como la clave principal. Incluir los otros atributos (si hay alguno) de la subclase en esta tabla. Crear tablas separadas para cada una de las otras superclases y mapearlas como se mapearían las entidades regulares.

21. Mapeo de categorías o tipos unión cuando las superclases tienen diferentes claves principales. Crear una nueva tabla para la subclase (o tipo unión). Crear una clave substituta para esta tabla. Esta será la clave principal para esta tabla. Incluir cualquier otro atributo (si hay alguno) de esta subclase en esta tabla. Crear tablas separadas por cada una de las superclases y mapearlas como se mapearían entidades regulares. Incluir la clave substituta en las tablas superclase como una clave foránea. (traducción propia)

Es indispensable conocer las reglas anteriores si se desea un mapeo al modelo lógico relacional, el pasar nuestro modelo conceptual a otro modelo lógico no está considerado en la obra de Bagui (2011).

1.2.3 Normalización

Bagui (2011) aborda el tema de los datos y sus modelos en todo un capítulo y más tarde, proporciona una explicación completa sobre el modelo relacional, en él nos expresa que la premisa de Edgar Codd para bases de datos era que los datos deberían estar en tablas de dos dimensiones no ordenadas. Los datos al estar acomodados de esta manera nos permiten realizar consultas de forma directa e ignorar la forma en la que están almacenados físicamente. Bagui señala que de acuerdo con Codd no nos interesa la vinculación física de los datos, sino que estén en la forma adecuada. La forma adecuada significa que los datos estén normalizados y la normalización es alcanzada por seguir la noción de Dependencias Funcionales (traducción propia). Bagui (2011, 39) señala que “una dependencia funcional es una relación que existe entre un atributo y otro”. Bagui nos expresa que una dependencia funcional o FD (por sus siglas en inglés) indican como un atributo es definido por otro, la definición y utilización de dependencias funcionales nos permite encontrar claves principales, que a su vez permiten encontrar coincidencias únicas de registros (traducción propia).

Al aplicar las reglas de mapeo se debería obtener una base de datos relacional en la tercera forma normal o cerca a esta (Bagui 2011, 293, traducción propia).

A continuación, Bagui (2011, 293) nos da una serie de consejos para garantizar la normalización en la base de datos obtenida de haber utilizado las reglas de mapeo anteriores:

1. 1NF (primera forma normal) – Revisar que no hay atributos no atómicos en alguna tabla. Las presencias de estos se manejan utilizando las reglas de mapeo 3 o 4 para atributos compuestos o de valores múltiples.
2. 2NF (segunda forma normal) – Revisar que todos los atributos en las tablas dependan en la clave principal completa. La mejor manera es preguntándose, “¿Obtendré el mismo valor para el atributo Y cuando yo tengo el valor X cuando X es la clave principal?”. X en este caso podría ser una clave concatenada, en la cual se buscarían dependencias parciales.
3. 3NF (tercera forma normal) – Revisar para que situaciones un atributo que está en la tabla es mejor definido por otro atributo, el cual no es la clave principal. Se recuerda que si la clave principal en una tabla es X y $X \rightarrow YZW$, entonces si $Z \rightarrow W$ es más claro que $X \rightarrow W$, se tendrá una dependencia transitiva y se necesitará normalizarla. (traducción propia)

El uso de la metodología propuesta por Bagui (2011) para diseño de bases de datos, acompañado de análisis nos permitirá obtener una base de datos relacional lista para ser implementada por nuestro sistema de información desarrollado a su vez por su propuesta de ingeniería de software.

A continuación, un breve ejemplo de la utilización de la metodología, con entidades del esquema de una de las bases de datos del sistema:

Paso 1. Se elige la entidad FELIGRÉS de la descripción de requerimientos.

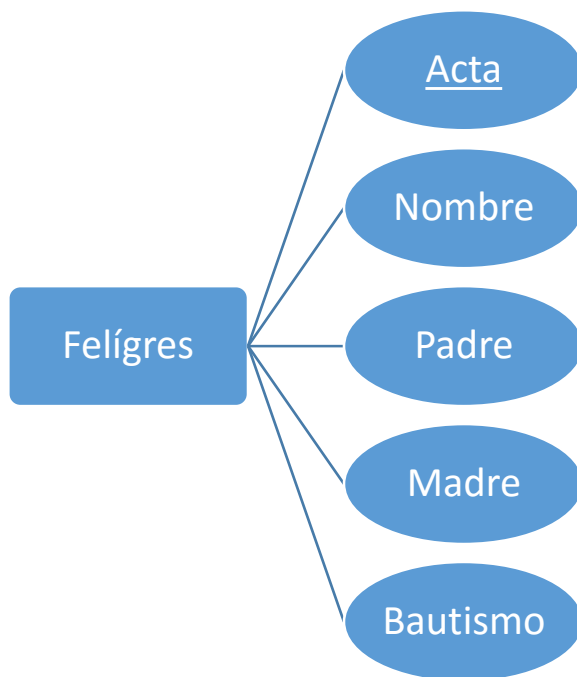


Figura 1.5 Entidad FELIGRÉS con atributos.

Paso 2. Se utiliza en lugar de inglés estructurado español para las gramáticas correspondientes a la entidad y sus atributos.

Feligrés

Esta base de datos registra datos sobre FELIGRESES. Por cada FELIGRÉS, se registra un acta, un nombre, un padre, una madre, un bautismo y una confirmación.

Atributos

Por cada FELIGRÉS, se tendrá solamente un acta, un nombre, un padre, una madre, un bautismo y una confirmación. Ninguno de estos atributos será subdividido.

Clave

Por cada FELIGRÉS, se tendrá la siguiente clave principal: acta.

Paso 3. Se revisa si se necesita más información de algún atributo de la entidad FELIGRÉS, se observa que se necesita más datos con respecto al bautismo (la fecha, el lugar, que sacerdote lo administro y que párroco estaba a cargo de tal lugar).

Paso 3a. Se convierte el atributo bautismo en una entidad.

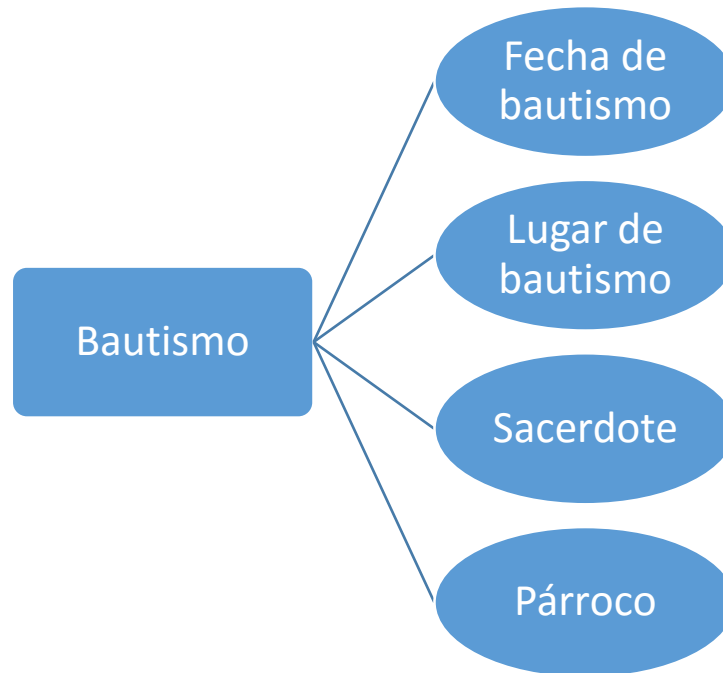


Figura 1.6 Entidad BAUTISMO con atributos.

Bautismo

Esta base de datos registra datos sobre el BAUTISMO. Por cada BAUTISMO, se registra una fecha, un lugar, un sacerdote y un párroco.

Atributos

Por cada BAUTISMO, se tendrá solamente una fecha, un lugar, un sacerdote y un párroco. Ninguno de estos atributos será subdividido.

Clave

Por cada BAUTISMO, se asume que ningún atributo será lo suficientemente único para identificar entidades individuales.

Paso 3b. Se define la relación entre la entidad BAUTISMO y FELIGRÉS como “recibe”.

Paso 5. Se omite el paso 4 ya que se viene desde el paso 3.

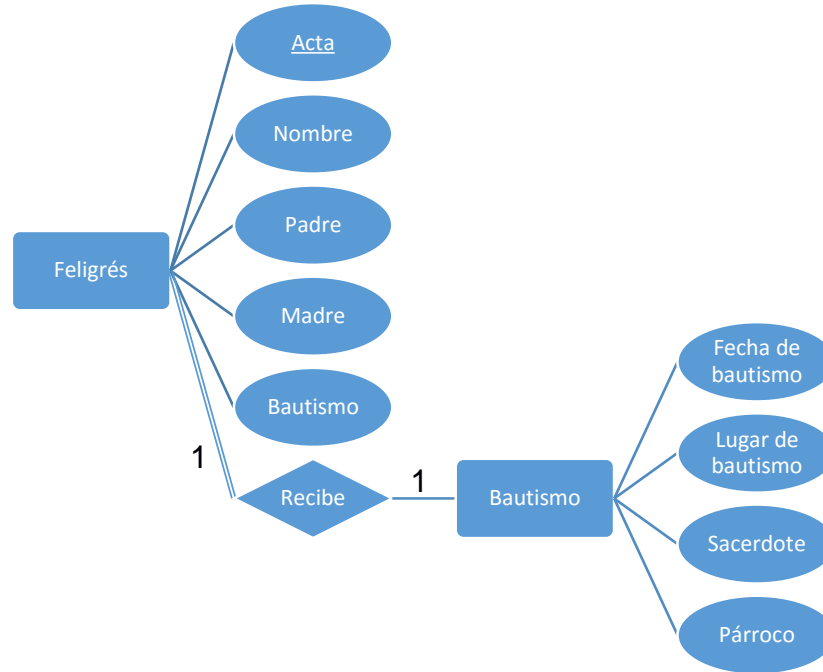


Figura 1.7 Relación entre Feligrés y Bautismo

Paso 6. De las gramáticas se utiliza el patrón 1 (x:y::k:1 donde k=1 y tiene participación total) en ambas direcciones, es decir, desde x= Feligrés y y=Bautismo y x=Bautismo y y=Feligrés.

Relaciones

Los FELIGRESES, que son registrados en la base de datos, **deben** recibir solamente un BAUTISMO. Ningún FELIGRÉS recibe más de un Bautismo.

Los BAUTISMOS, que son registrados en la base de datos, deben ser recibidos por solamente un FELIGRÉS.

Paso 7. Se utilizan las reglas de mapeo para entidades fuertes, débiles y sus respectivos atributos simples.

Feligrés			
<u>Acta</u>	Nombre	Padre	Madre
19850726	Ramón Apreza Zepeda	Ramón Apreza Galván	María del Pilar Zepeda Montes
19861021	Marco Alejandro Apreza Zepeda	Ramón Apreza Galván	María del Pilar Zepeda Montes

Figura 1.8 Tabla que muestra la entidad Feligrés con algunos datos muestra

Bautismo				
<u>Acta</u>	Fecha	Lugar	Sacerdote	Párroco
19850726	07-12-1985	Parroquia de Nuestra Señora de Guadalupe, Estado de México, Cd. Nezahualcoyotl	José de Jesús Sanchez	José de Jesús Sanche
19861021	29-01-1987	Parroquia de Nuestra Señora de Guadalupe, Estado de México, Cd. Nezahualcoyotl	José de Jesús Sanchez	José de Jesús Sanche

Figura 1.8 Tabla que muestra la entidad Bautismo con algunos datos muestra

1.3 COMENTARIOS Y RECOMENDACIONES

La metodología de ingeniería de software puede ser confusa y tan extensa que motivarían a cualquier programador a evitar utilizarla lo más que se pudiera, sustituyéndola por otras propuestas que ofrecen un “desarrollo ágil” o “basado sólo en pruebas”, asegurando concluir un proyecto y evitar lo tedioso del mismo.

Por esa razón se escogió la propuesta de Bagui, al ser tan breve y directa se obtiene la libertad de implementar otras herramientas de diseño y desarrollo (provenientes de otras metodologías) además de utilizar las propias.

A lo largo de las fases de desarrollo y diseño se utilizó herramientas de software y patrones que permiten avanzar disfrutando lo que se hace, razón por la cual se tiene una experiencia grata y enriquecedora.

Se puede recomendar a todo aquel que utilice este trabajo como referencia, que hay que aprovechar los espacios vacíos o las definiciones vagas cuando se utilice una metodología de cualquier índole, llenándolos o enriqueciéndolos con las experiencia y conocimiento de otros, así como, los propios. Es posible que se termine inventando algo nuevo o simplemente uno se divierta a lo largo de la realización de un proyecto.

CAPÍTULO 2

PLANTEAMIENTO DEL PROBLEMA

2.1 DESCRIPCIÓN GENERAL

En la parroquia Nuestra Señora de Guadalupe ubicada en Tacubaya No. 74 Col. Metropolitana 3ª Sección Cd. Nezahualcóyotl, México C.P. 57750 perteneciente a la diócesis de Nezahualcóyotl, se brindan los siguientes servicios a los feligreses:

- venta de criptas
- registro de intenciones para celebraciones públicas y privadas
- administración de sacramentos
- servicios notariales

Venta de criptas

En la parroquia antes mencionada se construyeron criptas, las cuales se pusieron a la venta y fueron adquiridas por los feligreses, algunas ya guardan urnas y otras están vacías, pero tienen un propietario y un beneficiario. La parroquia sigue ofreciendo aquellas que están disponibles y proporciona mantenimiento a todas. Se ha generado un archivo con los expedientes de las criptas vendidas.

Cuando un feligrés desea adquirir una cripta se genera un nuevo expediente, el cual posteriormente es agregado en ese archivo.

Cuando alguien desea colocar urnas debe haber liquidado la cripta, así como, haber realizado los pagos de mantenimiento anuales.

Registro de intenciones para celebraciones públicas o privadas

El servicio de registro de intenciones consiste en anotar intenciones que serán ofrecidas en celebraciones públicas o privadas, las cuales se clasifican como para difuntos, acción de gracias y peticiones. Muchas veces la consulta del calendario de fechas disponibles o la sobreposición de las mismas es el mayor problema con respecto a este servicio.

Administración de sacramentos

Con respecto a la administración de sacramentos se realizan los trámites correspondientes antes de participar en la celebración privada o pública.

Los trámites varían con respecto al sacramento:

En el caso de matrimonio se forma un expediente con los documentos correspondientes de los contrayentes (boleta de bautismo, boleta de confirmación, comprobante de primera comunión, acta de nacimiento –original y copia-, acta de matrimonio civil, comprobante de domicilio, fotografías, acta de matrimonio de los padrinos) y testigos en el próximo matrimonio

a celebrarse, así como, con los documentos que serán generados durante los trámites.

Se realizan varias reuniones tanto con un presbítero, con personas que imparten pláticas y testigos.

Se realiza una investigación con respecto a cada uno de los novios, para garantizar que realmente son solteros, conocer donde han vivido a partir de los catorce años y asegurarse que libremente desean contraer matrimonio.

Se generan documentos acordes al lugar donde se realizará el matrimonio, en la parroquia, en una parroquia que pertenece a la diócesis y en una parroquia que pertenece a otra diócesis. En caso de ser necesario son enviados al lugar correspondiente.

Un expediente de matrimonio se compone de los siguientes documentos:

- acta de matrimonio
- amonestación
- comprobante de pláticas prematrimoniales
- exhortos
- ficha de presentación
- permiso oficial
- permiso provisional
- permiso especial
- además de un documento extenso que contiene la información canónica previa
- los datos generales de los novios
- la declaración de los contrayentes
- la solicitud de los contrayentes
- las declaraciones de los testigos
- la ratificación de las declaraciones
- pruebas de consentimiento
- los datos del matrimonio civil realizado

Después de celebrado el matrimonio se regresan los documentos personales de los contrayentes, se les entrega su acta de matrimonio, su expediente es agregado al archivo parroquial guardando la carpeta con los documentos y asentando los datos del mismo en el libro de matrimonios.

En el caso de bautismo se solicita el acta de bautismo del niño (a), el acta de matrimonio de los padre y padrinos (en caso de ser soltero (a) se entrega un comprobante de soltería), se les pide que participen en unas pláticas a los padres y padrinos; después de celebrado el bautismo se entrega el acta a los padres y se asientan los datos en el libro de bautismos integrándose al archivo parroquial.

Cuando se trata de una confirmación y una primera comunión, se pide la participación del feligrés en un curso de preparación, el cual puede necesitar la participación activa de los padres y padrinos, la celebración del sacramento se realiza en la parroquia o en la catedral de la diócesis y después de entregados los comprobantes a los feligreses, los datos correspondientes son asentados en el libro de confirmaciones o de primera comunión.

Estos sacramentos son mencionados por que son los que implican un registro de información que más tarde es parte fundamental para los servicios notariales de la parroquia. Los cuales consisten en la consulta del archivo parroquial y la emisión de documentos para la realización de trámites en la misma u otra parroquia. Este servicio junto con el de matrimonio son los que exigen más tiempo a las secretarias que son las encargadas del archivo parroquial, en algunos casos pueden tomar días o semanas.

El problema común en cada uno de los servicios es la consulta del archivo parroquial y la generación de documentos, los cuales requieren bastante tiempo para su realización. Ante este problema se desarrollará un sistema de información que permita digitalizar el archivo parroquial y de esta manera facilitar su consulta y la emisión de los documentos correspondientes.

2.2 COMENTARIOS Y RECOMENDACIONES

Es necesario tener una mirada objetiva al momento de plantear el problema, el dialogo y la observación son esenciales, pues de lo contrario, algo que no es un problema podría tomarse como tal y uno real dejarse sin resolver. Además, si se contempla el ambiente, se dialoga con las personas que viven las situaciones que se consideran un problema, es posible con su experiencia delimitar el problema y ser capaz de brindar una solución.

CAPÍTULO 3

REQUERIMIENTOS DEL SISTEMA

3.1 CLARIFICACIÓN DE REQUERIMIENTOS

El usuario como antes se menciona es aquel que solicita al analista el sistema de información y proporciona las especificaciones del mismo, las cuales son sus deseos o necesidades (Bagui 2011, traducción propia). Hay que considerar que el usuario muchas veces no tiene un vasto conocimiento en el área de tecnología de la información, por lo cual identificar estas necesidades o deseos será una ardua tarea para el analista, ya que el usuario procurará describir lo que hace y lo que le gustaría que hiciera el sistema con sus propias palabras.

Después de asistir a la parroquia de Nuestra Señora de Guadalupe y llegar a un acuerdo con el párroco se realizó la presentación con los usuarios del futuro sistema. Al principio eran tres usuarios responsables de la atención de los feligreses, las consultas al archivo, el manejo de archivo de criptas y otras tareas administrativas, a los cuales se estuvo entrevistando y observando continuamente, se les presentaron observaciones como posibles especificaciones, recibiendo su retroalimentación (feedback) y solicitando el mayor número de políticas posibles con respecto a las tareas que realizaban, también se recibieron archivos, formatos e incluso bases de datos pensadas para auxiliarse en la realización de las mismas.

A continuación, se presenta un resumen de los requerimientos del sistema:

El archivo parroquial está compuesto por los libros de sacramentos administrados en la parroquia y por los expedientes relacionados con cada sacramento o servicio que implica un registro (adquisición de criptas, bautismo y matrimonio principalmente). Dicho archivo es necesario digitalizarlo, es decir, capturar todos los datos en el sistema, para realizar las consultas necesarias relacionadas con los servicios brindados a los feligreses.

El proceso más común o la dinámica de trabajo más realizada es la siguiente: recibir la solicitud del feligrés, la cual algunas veces será una consulta de requisitos para recibir un servicio específico y otras veces un servicio como tal. Después es necesario que el feligrés cumpla con los requisitos (entre ellos realizar un pago) o finalice los trámites relacionados con dicho servicio y finalmente reciba el servicio solicitado.

El proceso anterior puede llevar de unos días a meses en su realización, por lo cual se debe poder llevar un cierto control o monitorear el avance del mismo.

El usuario debe registrar a todo feligrés que reciba un servicio, aunque no sea quien lo solicita.

El usuario debe ser capaz de registrar toda adquisición de cripta, llevar el control de pagos y modificar el estado (con respecto a si hay urnas o no) de las mismas.

El usuario va a guardar la información que conforma un expediente de matrimonio, cripta o cualquier otro servicio.

El usuario generará todos los documentos relacionados con un servicio o los trámites que realiza un feligrés.

El usuario podrá registrar múltiples servicios en una misma ocasión con un solo pago o muchos.

El usuario debe evitar que se empalmen servicios especiales, como es el caso de celebraciones privadas de distintos feligreses no relacionados.

El usuario debe de ser capaz de poder obtener información con respecto a cuantos servicios se brindaron en cierto periodo de tiempo, así como el número de pagos relacionados con los mismos.

En caso de ser necesario el sistema de información debe ser capaz de reproducir el archivo parroquial físico que actualmente se resguarda en la parroquia.

El usuario necesita poder identificar que feligrés solicito un servicio, así como, que feligrés recibe o recibió un servicio.

El usuario necesita que el sistema sea capaz de realizar la dinámica de trabajo y políticas antes mencionadas, sea de fácil acceso y pueda ser utilizado por nuevos usuarios.

3.2 COMENTARIOS Y RECOMENDACIONES

Finalizo este capítulo haciendo énfasis en el último punto mencionado, dado que se presentaron ciertas dificultades ante el cambio de usuarios de manera continua e impredecible, continuamente tuve que modificar las especificaciones, operaciones y posibles políticas por la discrepancia entre los usuarios y ante la presencia o ausencia de los mismos.

Es bueno identificar a los usuarios más estables dentro de la organización, para garantizar su permanencia a lo largo del proyecto y de ahí una posible especificación consistente.

Dialogo constante, auxiliarse de grabaciones en audio y vídeo, enseñar a los usuarios lo básico de algoritmos y entablar una buena relación con los usuarios hacen de la clarificación de requerimientos una tarea fácil. Al menos intente enseñarles a los usuarios a hacer una lista de pasos de las tareas que comúnmente realizaban, varias ocasiones les ayude con soporte técnico y algunas otras con consejos con respecto a los consumibles en la oficina, esto me permitió gozar de una total apertura y apoyo por parte de los usuarios, a pesar de ser distintos de con quienes había comenzado este proyecto.

CAPÍTULO 4 ESPECIFICACIÓN DEL SISTEMA

4.1 ESPECIFICACIÓN GENERAL

Este capítulo comienza con una serie de listas de los formatos y archivos proporcionados por los usuarios, que junto con las observaciones antes mencionadas en el capítulo anterior son la especificación general del sistema. Las siguientes secciones desglosan y formalizan dicha especificación al presentar las especificaciones de las bases de datos que serán utilizadas por el sistema, las cuales son resultado de la implementando la metodología de Bagui (2011, traducción propia) orientada al diseño de base de datos.

Formatos

Con respecto al pago de servicios utilizan dos tipos de recibos: recibo de pago general y recibo de criptas.

Con respecto a cada servicio, sacramento o tramite relacionado:

Bautismo

- Requisitos
- Requisitos de actualización de acta
- Comprobante de pláticas
- Acta de bautismo por género
- Boleta de bautismo por género

Matrimonio

- Requisitos
- Acta
- Amonestación
- Comprobante de pláticas
- Exhorto
- Expediente
- Ficha de presentación
- Permisos

Celebraciones

- Requisitos de tres años
- Requisitos de XV años
- Comprobante de pláticas de XV años

Criptas

- Documento de perpetuidad
- Requisitos

- Solicitud de donativo
- Estado de pagos en ambas modalidades
- Esquemas de criptas por cada pasillo (San Juan, San Mateo, San Pablo, San Pedro)

Imágenes de los libros

- Bautismo realizado por el párroco
- Bautismo realizado por el vicario
- Confirmación
- Primera Comuni3n
- Matrimonio

Formatos poblados de ejemplo

Bases de datos

- Bautismo
- Matrimonio

Archivos

- Control de criptas
- Formas de cada documento a emitir por la parroquia
- Intenciones
- Versi3n digital de actas y boletas

Así como Bagui (2011, traducci3n propia) utiliza s3lo letras may3sculas en sus patrones de gramáticas para el diseño de base de datos, se utiliza el mismo estilo para facilitar encontrar las entidades dentro de las gramáticas que acompañan a los modelos. Adem3s, cada gramática es delimitada por el nombre en negritas de la entidad y sus elementos (atributos, claves, relaciones) se encuentran subrayados, esto con el propósito de facilitar su lectura.

Cuando hay relaci3n entre entidades se procura que despu3s de colocar la última gramática de la entidad relacionada, colocar la descripci3n de su relaci3n. Un ejemplo sería el siguiente: existe una relaci3n “tiene” entre FELIGRÉS y DOMICILIO, entonces se coloca las gramáticas de la entidad FELIGRÉS, despu3s, se continua con las gramáticas de la entidad DOMICILIO, al finalizar se coloca un encabezado “Relaciones” donde vendrán descritos dos patrones que describen la relaci3n “tiene”, una desde el sentido de la entidad FELIGRÉS y otra desde el sentido de la entidad DOMICILIO.

4.2 ESPECIFICACI3N DE BASE DE DATOS PARA ARCHIVO PARROQUIAL Y SERVICIOS

De la informaci3n contenida en los formatos y la continua comunicaci3n con los usuarios, pude determinar que una base de datos podría contener, la informaci3n de los feligreses, los sacramentos y servicios recibidos por los mismos, el nombre de la base de datos es “parroquia”, pero me referiré a ella en las siguientes secciones como “archivo y servicios”. La siguiente secci3n muestra los atributos de sus tablas componente. El diagrama se encuentra en los anexos, ya que debido a su extensi3n necesita mucho espacio, así como, sus diagramas

componente que son necesarios para apreciarlo adecuadamente. Después viene sus gramáticas y finalmente cada sección termina con su base de datos relacional.

4.2.1 Atributos de las tablas de la base de datos de archivo y servicios

De toda esta información recibida estas son las listas de atributos de la base de datos:

Feligrés

- ID Feligrés
- Nombre
 - Nombre
 - Apellido paterno
 - Apellido materno
- Fecha de nacimiento
- Lugar de nacimiento
- Fecha de bautismo
- Lugar de bautismo
 - Parroquia
 - Dirección
 - Diócesis
- Padre
- Madre
- Registro civil
- Entidad
- Delegación
- Juzgado
- Acta
- Año
- Fecha de registro
- Libro
- Foja
- Localidad
- Clase
- Genero
- Estado civil
- Teléfono
- Religión
- Grado de estudios
- Profesión u ocupación
- Parroquia a la que pertenece
- Primera comunión
 - Recepción
 - Edad
- Fecha de defunción
- ID Contrayente

Domicilio

- ID Domicilio
- Domicilio
- Tiempo de residencia
- Parroquia a la que pertenece
- Dirección de la parroquia a la que pertenece el domicilio

Pago

- ID Pago
- Fecha
- Concepto
 - Intención
 - Matrimonio
 - XV años
 - Aniversario
 - Presentación
 - Archivo
 - Otro
- Fecha de la Misa
- Hora de la Misa
- Nombre de quien recibió
- Total
- A cuenta
- Resta
- Fecha de liquidación de pago

Bautismo

- Acta
- Nombre del padrino
- Nombre de la madrina
- Libro
- Foja
- Notas marginales
- Sacerdote
- Párroco
- Fecha de expedición

Primera comunión

- Padrino
- Madrina
- Fecha de primera comunión
- Sacerdote
- Párroco

Confirmación

- Padrino
- Madrina
- Lugar de confirmación
- Fecha de confirmación
- Sacerdote
- Párroco

Consagración

- Lugar de la ordenación o profesión religiosa
- Fecha de la ordenación o profesión religiosa
- Sacerdote
- Párroco

Matrimonio

- Acta
- Libro
- Foja
- Parroquia donde se realizaron los tramites
- Sacerdote
- Nombre del párroco
- Fecha de primera publicación
- Fecha de segunda publicación
- Fecha de tercera publicación
- Fecha de primera plática
- Fecha de segunda plática
- Fecha de tercera plática
- Horario de pláticas
- Número de expediente
- Lugar donde se celebrará el matrimonio
- Nombre de la parroquia donde se van a casar
- Dirección de la parroquia donde se van a casar
- Diócesis de la parroquia donde se van a casar
- Presentación completa
- Documentación completa
- Pláticas prematrimoniales
- Proclamas
- Parroquias a las que se envió exhortos
- Fecha de ratificación
- Dispensas
 - Estado
 - Contenido
- Licencias
 - Estado
 - Contenido
- Celebración
 - Estado
 - Contenido

- Notificación
 - Estado
 - Contenido
- Fecha de matrimonio civil
- Lugar de matrimonio civil
- Oficialía
- Libro civil
- Acta civil
- Foja civil
- Lugar de matrimonio civil
- Domicilio del padre o tutor
- Fecha de la presentación
- Hora
- Costo
- Aceptó
- Edad del novio
- Edad de la novia
- Fecha de matrimonio

Exhorto

- Asunto
- Fecha de expedición

Solicitud

- Cumplo con mis deberes religiosos
- Conozco la doctrina cristiana
- Iniciamos nuestro noviazgo, aproximadamente hace
- Mis padres están enterados de mi noviazgo y de mi decisión de contraer matrimonio
- Domicilio después de celebrado el matrimonio
- Fecha de solicitud

Declaración

- Se casa usted por su propia y libre voluntad o hay alguien o algo que le obligue a casarse
- Tiene usted algún compromiso anterior con alguna otra persona relacionado con su matrimonio
- Con quién
- Qué tipo de compromiso
- Cuánto tiempo tienen de noviazgo
- Cómo ha sido su noviazgo
- Sabe usted las obligaciones y derechos del matrimonio
- Está usted dispuesto a cumplir con esos derechos y obligaciones
- Por qué quiere casarse por la Iglesia Católica
- Sabe usted lo que significa que el Matrimonio sea un Sacramento
- Sabe usted que el Matrimonio es una Alianza entre un hombre y una mujer y que es uno e indisoluble, perpetuo y exclusivo hasta la muerte
- Se compromete a vivirlo así
- Cómo considera usted su formación religiosa

- Está usted dispuesto a vivir sus valores cristianos en la familia que va a constituir y en la Comunidad Parroquial
- Existe algún impedimento, o circunstancia física, psíquica, moral o religiosa por parte de usted o del otro contrayente que se oponga a la celebración de este matrimonio
- Cómo considera su estado de salud física y psíquica con relación a la nueva vida que pretende iniciar
- Tiene usted algún vicio o adicción que pudiera trastornar la vida matrimonial
- Sabe si el otro contrayente los tiene

Testigo

- Nombre del testigo
- Domicilio del testigo
- Teléfono del testigo
- Documento oficial de identificación del testigo
- Desde cuándo conoce usted al contrayente
- Por qué motivo
- Sabe usted si el contrayente es soltero
- Sabe usted si el contrayente quiere casarse por su voluntad
- O alguien o algo lo obliga
- Quién
- Hay algún obstáculo o circunstancia física, psíquica, moral o religiosa que se oponga a la celebración de este matrimonio
- Cuál
- Fecha de declaración de testigo

Permiso

- Fecha de permiso
- Nombre del presbítero delegado
- Nombre de la parroquia emisora
- Dirección de la parroquia emisora
- Diócesis de la parroquia emisora
- Fecha de inicio de tramites
- Presbítero que suscribe
- Nombre de la parroquia a su cargo
- Dirección de la parroquia a su cargo
- Nombre del obispo de la diócesis de la parroquia donde se van a casar
- Tipo de permiso

Servicio

- Folio
- Tipo

Cripta

- Número de expediente
- Nombre del propietario
- Dirección del propietario

- Teléfono del propietario
- Nombre del beneficiario
- Dirección del beneficiario
- Teléfono del beneficiario
- Nicho
- Sección
- Nivel
- Pasillo

Registro de pagos cripta

- Tipo de pago
- Cantidad
- Cantidad en letra
- Donativo inicial
- Mensualidades
- Saldo pendiente
- Número de recibo
- Fecha
- Reservación
- Fecha de inicio de pagos
- Fecha de elaboración
- Dirección de la parroquia
- Teléfono de la parroquia
- Concepto
- Recibió

Intención

- Enunciado
- Tipo de intención
- Fecha
- Fecha y hora de la celebración

Archivo

- Error
- Párroco
- Parroquia
- Dirección de la parroquia
- Libro a modificar
- Registro anterior
 - Libro
 - Foja
 - Acta
- Nuevo registro
 - Libro
 - Foja
 - Acta

- Fecha de transcripción
- Fecha de entrega a la curia diocesana
- Correcciones

4.2.2 Diagrama E-R de la base de datos “archivo y servicios”

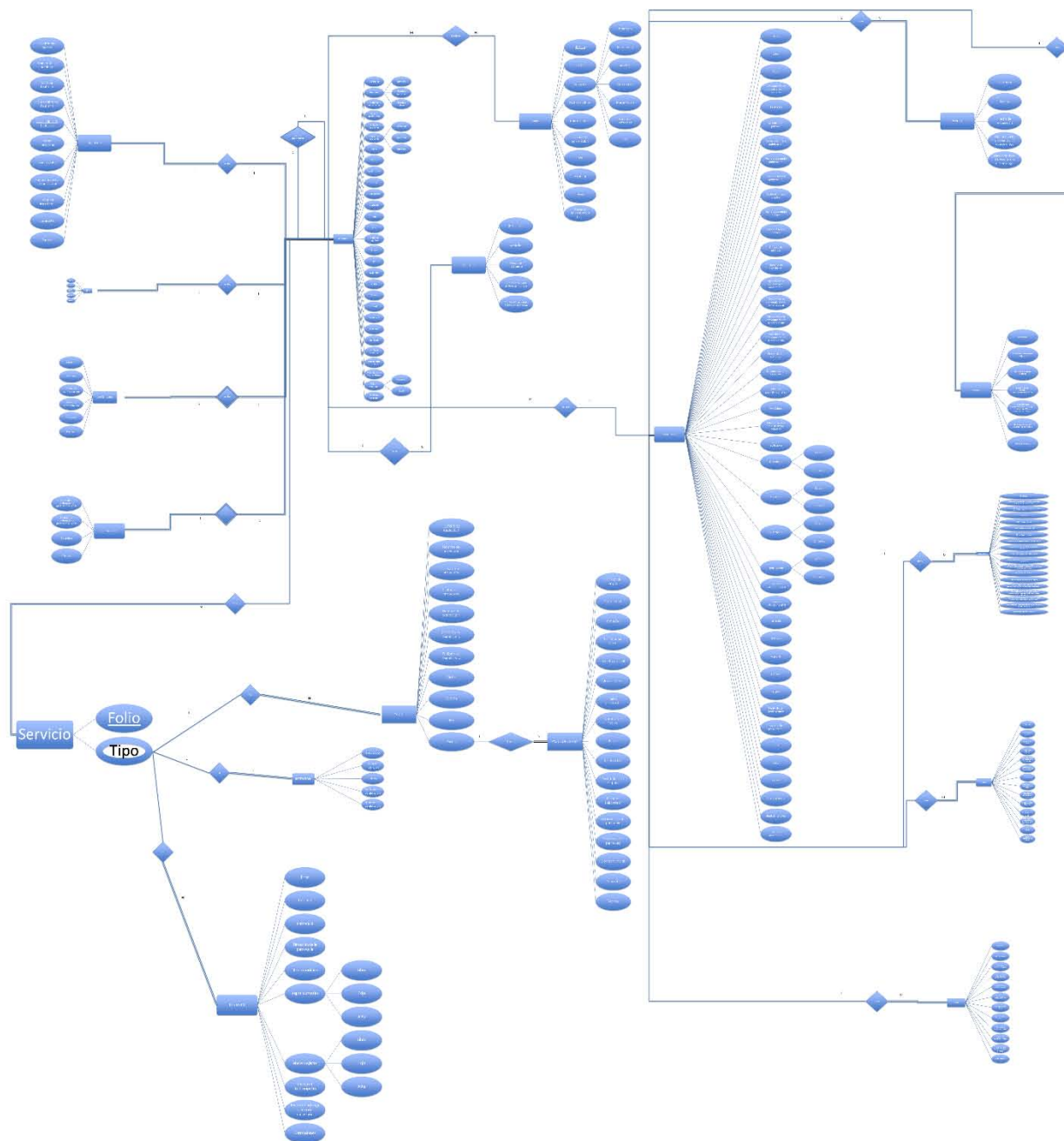


Figura 4.1 Diagrama E-R de la base de datos “archivo y servicios”

En la sección siguiente se encuentran las gramáticas que a detalle describen el significado del modelo Entidad-Relación Mejorado de la base de datos para el archivo parroquial y sus servicios. Una imagen con mayor resolución y sus entidades componente se encuentran en los anexos, esta es tan sólo para ayudar al lector a visualizar el esquema de bases de datos resultante de emplear la metodología de Bagui para diseño de bases de datos.

4.2.3 Gramáticas del diagrama EER de base de datos para archivo parroquial y servicios

Las siguientes son unas notas con respecto a la nomenclatura de las entidades que estarán presentes en el diagrama y en las gramáticas:

El servicio de venta de criptas, que incluye todos los trámites relacionados con él mismo (su adquisición, contrato, registro de pagos, etc.) se denomina entidad CRIPTA.

Servicios notariales se convierte en la entidad DECRETO.

El servicio de registro de intenciones pasa a ser la entidad INTENCIÓN

En este diagrama la entidad principal es FELIGRÉS.

Feligrés

Esta base de datos registra datos sobre FELIGRESES. Por cada FELIGRÉS, se registra un ID Feligrés, un nombre, una fecha de nacimiento, un lugar de nacimiento, una fecha de bautismo, un lugar de bautismo, al padre, a la madre, registro civil, entidad, delegación, juzgado, acta, año, fecha de registro, libro, foja, localidad, clase, género, estado civil, teléfono, religión, grado de estudios, profesión u ocupación, parroquia a la que pertenece, primera comunión y fecha de defunción.

Atributos

Por cada FELIGRÉS en la base de datos, se tendrá solamente un ID Feligrés, una fecha de nacimiento, un lugar de nacimiento, una fecha de bautismo, al padre, a la madre, registro civil, entidad, delegación, juzgado, acta, año, fecha de registro, libro, foja, localidad, clase, género, estado civil, teléfono, religión, grado de estudios, profesión u ocupación, parroquia a la que pertenece y fecha de defunción. Ninguno de estos atributos será subdividido.

Por cada FELIGRÉS en la base de datos, se registra un nombre, el cual está compuesto por nombre, apellido paterno, apellido materno, (nombre, apellido paterno, apellido materno) son las partes componente de nombre.

Por cada FELIGRÉS en la base de datos, se registra un lugar de bautismo, el cual está compuesto por parroquia, dirección y diócesis, (parroquia, dirección y diócesis) son las partes componente de lugar de bautismo.

Por cada FELIGRÉS en la base de datos, se registra una primera comunión, la cual está compuesta de recepción y edad, (recepción y edad) son las partes componente de primera comunión.

Clave

Por cada FELIGRÉS, se tendrá la siguiente clave principal: ID Feligrés.

Domicilio

Esta base de datos registra datos sobre DOMICILIOS. Por cada DOMICILIO, se registra un ID Domicilio, un domicilio, un tiempo de residencia y la parroquia a la que pertenece.

Atributos

Por cada DOMICILIO en la base de datos, se tendrá solamente un ID Domicilio, un domicilio y un tiempo de residencia. Ninguno de estos atributos será subdividido.

Por cada DOMICILIO en la base de datos, se registra una parroquia a la que pertenece, la cual está compuesto por el nombre de la parroquia a la que pertenece y la dirección de la parroquia a la que pertenece, (nombre de la parroquia a la que pertenece y la dirección de la parroquia a la que pertenece) son las partes componente de nombre.

Clave

Por cada DOMICILIO, se tendrá la siguiente clave principal: ID Domicilio.

Relaciones

FELIGRÉS y DOMICILIO

Los FELIGRESES, pero no necesariamente todos los feligreses (que están registrados en la base de datos), podrían tener muchos (uno o más) DOMICILIOS. Algunos FELIGRESES podrían no tener un DOMICILIO.

Los DOMICILIOS, que están registrados en la base de datos, deben pertenecer a solamente un FELIGRÉS. Ningún DOMICILIO pertenece a más de un FELIGRÉS.

Feligrés y Feligrés (matrimonio)

FELIGRÉS y FELIGRÉS (denominado contrayente)

Los FELIGRESES, pero no necesariamente todos los feligreses (que están registrados en la base de datos) podrían contraer matrimonio con solamente un FELIGRÉS. Algunos FELIGRESES podrían no contraer matrimonio con un FELIGRÉS.

Pago

Esta base de datos registra datos sobre los PAGOS. Por cada PAGO realizado se registra un ID Pago, una fecha, un concepto, la fecha de la misa, hora de la misa, el nombre de quien recibió, total, a cuenta, resta y fecha de liquidación de pago.

Atributos

Por cada PAGO en la base de datos, se tendrá solamente un ID Pago, una fecha, la fecha de la misa, hora de la misa, el nombre de quien recibió, total, a cuenta, resta y fecha de liquidación de pago. Ninguno de estos atributos será subdividido.

Por cada PAGO, se registra un concepto, es cual está compuesto de intención, matrimonio, xv años, aniversario, presentación, servicios notariales y otro, (intención, matrimonio, xv años, aniversario, presentación, servicios notariales y otro) son las partes componente de concepto.

Clave

Por cada PAGO, se tendrá la siguiente clave principal: ID Pago.

Relaciones

FELIGRÉS y PAGO (realiza)

Los FELIGRESES, pero no necesariamente todos los FELIGRESES (que están registrados en la base de datos), podrían realizar muchos (uno o más) PAGOS. Algunos FELIGRESES podrían no realizar algún PAGO.

Los PAGOS, que están registrados en la base de datos, deben ser realizados por solamente un FELIGRÉS. Ningún PAGO puede ser realizado por más de un FELIGRÉS.

Servicio

Esta base de datos registra datos sobre SERVICIOS. Por cada SERVICIO se registra folio y tipo.

Atributos

Por cada SERVICIO, habrá solamente un folio y un tipo. Ninguno de estos atributos será subdividido.

Clave

Por cada SERVICIO, se tendrá la siguiente clave principal: folio.

Relaciones

FELIGRÉS y SERVICIO (recibir)

Los FELIGRESES, pero no necesariamente todos los FELIGRESES (que están registrados en la base de datos), podría recibir muchos (uno o más) SERVICIOS. Algunos FELIGRESES podrían no recibir un SERVICIO.

Los SERVICIOS, que están registrados en la base de datos, deben ser recibidos por solamente un FELIGRÉS.

Cripta

Esta base de datos registra datos sobre CRIPTAS. Por cada CRIPTA, se registra un número de expediente, un nombre del propietario, la dirección del propietario, el teléfono del propietario, nombre del beneficiario, la dirección del beneficiario, el teléfono del beneficiario, el nicho, la sección, el nivel y el pasillo.

Atributos

Por cada CRIPTA en la base de datos, se tendrá solamente un número de expediente, un nombre del propietario, una dirección del propietario, un teléfono del propietario, un nombre del beneficiario, una dirección del beneficiario, un teléfono del beneficiario, un nicho, una sección, un nivel y un pasillo. Ninguno de estos atributos será subdividido.

Clave

Por cada CRIPTA, se tendrá la siguiente clave principal: número de expediente.

Relaciones

SERVICIO y CRIPTA (es)

Los SERVICIOS, pero no necesariamente todos los SERVICIOS (que están registrados en la base de datos), podría ser muchas (uno o más) CRIPTAS. Algunos SERVICIOS podrían no ser CRIPTAS.

Las CRIPTAS, que están registrados en la base de datos, deben ser solamente un SERVICIO.

Pagos de criptas

Esta base de datos registra datos sobre PAGOS DE CRIPTAS. Por cada PAGO DE CRIPTA, se registra un ID Pago de cripta, un tipo de pago, una cantidad, una cantidad en letra, un donativo inicial, mensualidades, saldo pendiente, número de recibo, fecha, reservación, fecha de inicio de pago, fecha de elaboración, dirección de la parroquia, teléfono de la parroquia, concepto y recibió.

Atributos

Por cada PAGO DE CRIPTA se tendrá solamente un ID Pago de cripta, un tipo de pago, una cantidad, una cantidad en letra, un donativo inicial, mensualidades, saldo pendiente, número de recibo, fecha, reservación, fecha de inicio de pago, fecha de elaboración, dirección de la parroquia, teléfono de la parroquia, concepto y recibió. Ninguno de estos atributos será subdividido.

Claves

Por cada PAGO DE CRIPTA, se tendrá la siguiente clave principal: ID Pago de cripta.

Relaciones

CRIPTA y PAGO DE CRIPTA (tiene)

Las CRIPTAS, pero no necesariamente todas las CRIPTAS (que están registradas en la base de datos), podrían tener muchos (uno o más) PAGOS DE CRIPTA. Algunas CRIPTAS podrían no tener PAGOS DE CRIPTA.

Los PAGOS DE CRIPTA, que están registrados en la base de datos deben pertenecer a una y solamente una CRIPTA. Ningún PAGO DE CRIPTA pertenece a más de una CRIPTA.

Intención

Esta base de datos registra datos sobre INTENCIONES. Por cada INTENCIÓN, se registra un enunciado, un tipo de intención, una fecha, una fecha y hora de celebración.

Atributos

Por cada INTENCIÓN, se tendrá solamente un enunciado, un tipo de intención, una fecha, una fecha y hora de celebración. Ninguno de estos atributos será subdividido.

Claves

Por cada INTENCIÓN, se tendrá la siguiente clave principal: ID Intención.

Relaciones

SERVICIO e INTENCIÓN (ser)

Los SERVICIOS, pero no necesariamente todas las SERVICIOS (que están registrados en la base de datos), podrían ser muchas (una o más) INTENCIONES. Algunos SERVICIOS podrían no ser INTENCIONES.

Las INTENCIONES, que están registradas en la base de datos deben ser solamente un SERVICIO.

Decreto

Esta base de datos registra datos sobre DECRETO (trámites relacionados con consultas al archivo parroquial). Por cada DECRETO, se registra un error, párroco, parroquia, dirección de la parroquia, libro a modificar, registro anterior, nuevo registro, fecha de transcripción, fecha de entrega a la curia diocesana y correcciones.

Atributos

Por cada DECRETO en la base de datos, se tendrá solamente un error, párroco, parroquia, dirección de la parroquia, libro a modificar, fecha de transcripción, fecha de entrega a la curia diocesana y correcciones. Ninguno de estos atributos será subdividido.

Por cada DECRETO en la base de datos, se almacena un registro anterior, el cual está compuesto de libro, foja y acta, (libro, foja y acta) son las partes componente de registro anterior.

Por cada DECRETO en la base de datos, se almacena un nuevo registro, el cual está compuesto de libro, foja y acta, (libro, foja y acta) son las partes componente de nuevo registro.

Relaciones

SERVICIO y DECRETO (ser)

Los SERVICIOS, pero no necesariamente todos los SERVICIOS (que están registrados en la base de datos), podrían ser muchos (uno o más) DECRETOS. Algunos SERVICIOS podrían no ser un DECRETO.

Los DECRETOS, que están registradas en la base de datos deben ser solamente un SERVICIO.

Bautismo

Esta base de datos registra datos sobre BAUTISMOS. Por cada BAUTISMO, se registra el nombre del padrino, el nombre de la madrina, el libro de bautismo, la foja en el libro de bautismo, el acta en el libro de bautismo, notas marginales, parroquia donde se van a casar, fecha de expedición, sacerdote y párroco.

Atributos

Por cada BAUTISMO, se tendrá solamente un nombre del padrino, nombre de la madrina, libro de bautismo, foja en el libro de bautismo, acta en el libro de bautismo, notas marginales, contrayente, parroquia donde se van a casar, fecha de expedición, sacerdote y párroco. Ninguno de estos atributos será subdividido.

Clave

Por cada BAUTISMO en la base de datos se tendrá la siguiente clave principal: acta en el libro de bautismo.

Relaciones

Los FELIGRESES, que están registrados en la base de datos, deben recibir solamente un BAUTISMO. Ningún FELIGRÉS recibe más de un BAUTISMO.

Los BAUTISMOS, que están registrados en la base de datos, deben ser recibidos por solamente un FELIGRÉS.

Primera comunión

Esta base de datos registra datos sobre la PRIMERA COMUNIÓN. Por cada PRIMERA COMUNIÓN, se registra un padrino, una madrina, una fecha de primera comunión, un sacerdote y un párroco.

Atributos

Por cada PRIMERA COMUNIÓN, se tendrá solamente un padrino, una madrina, una fecha de primera comunión, un sacerdote y un párroco. Ninguno de estos atributos será subdividido.

Clave

Por cada PRIMERA COMUNIÓN, no se asume que algún atributo será lo suficientemente único para identificar entidades individuales sin la referencia a FELIGRÉS, la entidad propietaria.

Dado que la entidad PRIMERA COMUNIÓN no tiene una clave candidata, cada entidad PRIMERA COMUNIÓN será identificada por una clave que pertenece a FELIGRÉS.

Relaciones

Los FELIGRESES, que están registrados en la base de datos, deben recibir solamente una PRIMERA COMUNIÓN. Ningún FELIGRÉS recibe más de una PRIMERA COMUNIÓN.

Las PRIMERAS COMUNIONES, que están registradas en la base de datos, deben ser recibidas por solamente un FELIGRÉS.

Confirmación

Esta base de datos registra datos sobre la CONFIRMACIÓN. Por cada CONFIRMACIÓN, se registra un padrino, una madrina, un lugar de confirmación, una fecha de confirmación, un sacerdote y un párroco.

Atributos

Por cada CONFIRMACIÓN, se tendrá solamente un padrino, una madrina, un lugar de confirmación, una fecha de confirmación, un sacerdote y un párroco. Ninguno de estos atributos será subdividido.

Clave

Por cada CONFIRMACIÓN, no se asume que algún atributo será lo suficientemente único para identificar entidades individuales sin la referencia a FELIGRÉS, la entidad propietaria.

Dado que la entidad CONFIRMACIÓN no tiene una clave candidata, cada entidad CONFIRMACIÓN será identificada por una clave que pertenece a FELIGRÉS.

Relaciones

Los FELIGRESES, que están registrados en la base de datos, deben recibir solamente una CONFIRMACIÓN. Ningún FELIGRÉS recibe más de una CONFIRMACIÓN.

Las CONFIRMACIONES, que están registradas en la base de datos, deben ser recibidos por solamente un FELIGRÉS.

Consagración

Esta base de datos registra datos sobre la CONSAGRACIÓN. Por cada CONSAGRACIÓN, se registra un lugar de ordenación o profesión religiosa y fecha de ordenación o profesión religiosa, un sacerdote y un párroco.

Atributos

Por cada CONSAGRACIÓN, se tendrá solamente un lugar de ordenación o profesión religiosa y fecha de ordenación o profesión religiosa, un sacerdote y un párroco. Ninguno de estos atributos será subdividido.

Clave

Por cada CONSAGRACIÓN, no se asume que algún atributo será lo suficientemente único para identificar entidades individuales sin la referencia a FELIGRÉS, la entidad propietaria.

Dado que la entidad CONSAGRACIÓN no tiene una clave candidata, cada entidad CONSAGRACIÓN será identificada por una clave que pertenece a FELIGRÉS.

Relaciones

Los FELIGRESES, que están registrados en la base de datos, deben recibir solamente una CONSAGRACIÓN. Ningún FELIGRÉS recibe más de una CONSAGRACIÓN.

Las CONSAGRACIONES, que están registradas en la base de datos, deben ser recibidas por solamente un FELIGRÉS.

Matrimonio

Esta base de datos registra datos sobre el MATRIMONIO. Por cada MATRIMONIO, se registra una acta, un libro, una foja, una parroquia donde se realizaron los tramites, sacerdote, el nombre del párroco, una fecha de primera publicación, una segunda fecha de publicación, una tercera fecha de publicación, una fecha de primera plática, una fecha de segunda plática, una fecha de tercera plática, un horario de pláticas, un número de expediente, un lugar donde se celebrará el matrimonio, un nombre de la parroquia donde se van a casar, una dirección de la parroquia donde se va a casar, la diócesis de la parroquia donde se van a casar, presentación completa, documentación completa, pláticas prematrimoniales, proclamas, parroquias a las que se envió exhortos, fecha de ratificación, dispensas, licencias, celebración, notificación, fecha de matrimonio civil, lugar de matrimonio civil, oficialía, libro civil, acta civil,

número, lugar, domicilio del padre o tutor, fecha de la presentación, hora, costo, aceptó, edad del novio, edad de la novia, fecha de matrimonio.

Atributos

Por cada MATRIMONIO en la base de datos, se tendrá solamente una acta, un libro, una foja, una parroquia donde se realizaron los tramites, sacerdote, el nombre del párroco, una fecha de primera publicación, una segunda fecha de publicación, una tercera fecha de publicación, una fecha de primera plática, una fecha de segunda plática, una fecha de tercera plática, un horario de pláticas, un número de expediente, un lugar donde se celebrará el matrimonio, un nombre de la parroquia donde se van a casar, una dirección de la parroquia donde se va a casar, la diócesis de la parroquia donde se van a casar, presentación completa, documentación completa, pláticas prematrimoniales, proclamas, parroquias a las que se envió exhortos, fecha de ratificación, fecha de matrimonio civil, lugar de matrimonio civil, oficialía, libro civil, acta civil, número, lugar, domicilio del padre o tutor, fecha de la presentación, hora, costo, aceptó, edad del novio, edad de la novia y fecha de matrimonio. Ninguno de estos atributos será subdividido.

Por cada MATRIMONIO en la base de datos, se registra dispensas, que está compuesto de estado y contenido, (estado y contenido) son las partes componente de dispensas.

Por cada MATRIMONIO en la base de datos, se registra licencias, que está compuesto de estado y contenido, (estado y contenido) son las partes componente de licencias.

Por cada MATRIMONIO en la base de datos, se registra celebración, que está compuesto de estado y contenido, (estado y contenido) son las partes componente de celebración.

Por cada MATRIMONIO en la base de datos, se registra notificación, que está compuesto de estado y contenido, (estado y contenido) son las partes componente de notificación.

Clave

Por cada MATRIMONIO, se tendrá la siguiente clave principal: acta.

Relaciones

FELIGRÉS y MATRIMONIO (celebra)

Los FELIGRESES, pero no necesariamente todos los FELIGRESES (que están registrados en la base de datos), podrían celebrar muchos (uno o más) MATRIMONIOS. Algunos FELIGRESES podrían no celebrar un MATRIMONIO.

Los MATRIMONIOS, que están registradas en la base de datos, deben ser celebrados por un FELIGRÉS.

Exhorto

Esta base de datos registra datos sobre EXHORTOS. Por cada EXHORTO, se registra un asunto y una fecha de expedición.

Atributos

Por cada EXHORTO, se tendrá solamente un asunto y una fecha de expedición. Ninguno de estos atributos será subdividido.

Clave

Por cada EXHORTO, se tendrá la siguiente clave principal: ID Exhorto.

Relaciones

Los MATRIMONIOS, pero no necesariamente todos los matrimonios (que están registrados en la base de datos), podrían tener muchos (uno o más) EXHORTOS.

Los EXHORTOS, que están registrados en la base de datos, deben pertenecer a solamente un MATRIMONIO.

Declaración

Esta base de datos registra datos sobre DECLARACIONES. Por cada DECLARACIÓN, se registra ID Declaración, se casa usted por su propia y libre voluntad o hay alguien o algo que le obligue a casarse, tiene usted algún compromiso anterior con alguna otra persona relacionado con su matrimonio, con quién, qué tipo de compromiso, cuánto tiempo tienen de noviazgo, cómo ha sido su noviazgo, sabe usted las obligaciones y derechos del matrimonio, está usted dispuesto a cumplir con esos derechos y obligaciones, por qué quiere casarse por la iglesia católica, sabe usted lo que significa que el matrimonio sea un sacramento, sabe usted que el matrimonio es una alianza entre un hombre y una mujer y que es uno e indisoluble, perpetuo y exclusivo hasta la muerte, se compromete a vivirlo así, cómo considera usted su formación religiosa, está usted dispuesto a vivir sus valores cristianos en la familia que va a constituir y en la comunidad parroquial, existe algún impedimento, o circunstancia física, psíquica, moral o religiosa por parte de usted o de su novio que se oponga a la celebración de este matrimonio, cómo considera su estado de salud física y psíquica con relación a la nueva vida que pretende iniciar, tiene usted algún vicio o adicción que pudiera trastornar la vida matrimonial y sabe si el otro contrayente los tiene.

Atributos

Por cada DECLARACIÓN, se tendrá solamente un ID Declaración, se casa usted por su propia y libre voluntad o hay alguien o algo que le obligue a casarse, tiene usted algún compromiso anterior con alguna otra persona relacionado con su matrimonio, con quién, qué tipo de compromiso, cuánto tiempo tienen de noviazgo, cómo ha sido su noviazgo, sabe usted las obligaciones y derechos del matrimonio, está usted dispuesto a cumplir con esos derechos y obligaciones, por qué quiere casarse por la iglesia católica, sabe usted lo que significa que el matrimonio sea un sacramento, sabe usted que el matrimonio es una alianza entre un

hombre y una mujer y que es uno e indisoluble, perpetuo y exclusivo hasta la muerte, se compromete a vivirlo así, cómo considera usted su formación religiosa, está usted dispuesto a vivir sus valores cristianos en la familia que va a constituir y en la comunidad parroquial, existe algún impedimento, o circunstancia física, psíquica, moral o religiosa por parte de usted o de su novio que se oponga a la celebración de este matrimonio, cómo considera su estado de salud física y psíquica con relación a la nueva vida que pretende iniciar, tiene usted algún vicio o adicción que pudiera trastornar la vida matrimonial y sabe si el otro contrayente los tiene. Ninguno de estos atributos será subdividido.

Clave

Por cada DECLARACIÓN, se tendrá la siguiente clave principal: ID Declaración.

Relaciones

Los MATRIMONIOS, pero no necesariamente todos los matrimonios (que están registrados en la base de datos), podrían tener muchas (una o más) DECLARACIONES.

Las DECLARACIONES, que están registradas en la base de datos, deben pertenecer a solamente un MATRIMONIO.

Testigo

Esta base de datos registra datos sobre TESTIGOS. Por cada TESTIGO, se registra un ID Testigo, un nombre del testigo, domicilio del testigo, teléfono del testigo, documento oficial de identificación del testigo, desde cuándo conoce usted al contrayente, por qué motivo, sabe usted si el contrayente es soltero, sabe usted si el contrayente quiere casarse por su voluntad, o alguien o algo lo obliga, quién, hay algún obstáculo o circunstancia física, psíquica, moral o religiosa que se oponga a la celebración de este matrimonio, cuál y fecha de declaración de testigo.

Atributos

Por cada TESTIGO, se tendrá solamente un ID Testigo, un nombre del testigo, domicilio del testigo, teléfono del testigo, documento oficial de identificación del testigo, desde cuándo conoce usted al contrayente, por qué motivo, sabe usted si el contrayente es soltero, sabe usted si el contrayente quiere casarse por su voluntad, o alguien o algo lo obliga, quién, hay algún obstáculo o circunstancia física, psíquica, moral o religiosa que se oponga a la celebración de este matrimonio, cuál y fecha de declaración de testigo. Ninguno de estos atributos será subdividido.

Clave

Por cada TESTIGO, se tendrá la siguiente clave principal: ID Testigo.

Relaciones

Los MATRIMONIOS, pero no necesariamente todos los matrimonios (que están registrados en la base de datos), podrían tener muchos (uno o más) TESTIGOS.

Los TESTIGOS, que están registrados en la base de datos, deben pertenecer a solamente un MATRIMONIO.

Solicitud

Esta base de datos registra datos sobre SOLICITUDES. Por cada SOLICITUD, se registra un ID Solicitud, cumpla con mis deberes religiosos, conozco la doctrina cristiana, iniciamos nuestro noviazgo aproximadamente hace, mis padres están enterados de mi noviazgo y de mi decisión de contraer matrimonio, domicilio después de celebrado el matrimonio y fecha.

Atributos

Por cada SOLICITUD, se tendrá solamente un ID Solicitud, cumpla con mis deberes religiosos, conozco la doctrina cristiana, iniciamos nuestro noviazgo aproximadamente hace, mis padres están enterados de mi noviazgo y de mi decisión de contraer matrimonio, domicilio después de celebrado el matrimonio y fecha. Ninguno de estos atributos será subdividido.

Clave

Por cada SOLICITUD, se tendrá la siguiente clave principal: ID Solicitud.

Relaciones

Los MATRIMONIOS, pero no necesariamente todos los matrimonios (que están registrados en la base de datos), podrían tener muchas (una o más) SOLICITUDES.

Las SOLICITUDES, que están registradas en la base de datos, podrían pertenecer a solamente un MATRIMONIO.

Permiso

Esta base de datos registra datos sobre PERMISOS. Por cada PERMISO, se registra un ID Permiso, una fecha de permiso, un nombre del presbítero delegado, un nombre de la parroquia emisora, dirección de la parroquia emisora, diócesis de la parroquia emisora, fecha de inicio de trámites, presbítero que suscribe, nombre de la parroquia a su cargo, dirección de la parroquia a su cargo, nombre del obispo de la diócesis de la parroquia donde se van a casar y tipo de permiso.

Atributos

Por cada PERMISO, se tendrá solamente un ID Permiso, una fecha de permiso, un nombre del presbítero delegado, un nombre de la parroquia emisora, dirección de la parroquia emisora, diócesis de la parroquia emisora, fecha de inicio de trámites, presbítero que suscribe, nombre de la parroquia a su cargo, dirección de la parroquia a su cargo, nombre del obispo de la diócesis de la parroquia donde se van a casar y tipo de permiso. Ninguno de estos atributos será subdividido.

Clave

Por cada PERMISO, se tendrá la siguiente clave principal: ID Permiso.

Relaciones

Los MATRIMONIOS, pero no necesariamente todos los matrimonios (que están registrados en la base de datos), podrían tener muchos (uno o más) PERMISOS.

Los PERMISOS, que están registrados en la base de datos, deben pertenecer a solamente un MATRIMONIO.

4.2.4 Base de datos acorde al modelo lógico relacional

FELIGRÉS

<u>ID Feligrés</u>	nombre.nombre	nombre.apellido paterno	nombre.apellido materno	fecha de nacimiento
--------------------	---------------	-------------------------	-------------------------	---------------------

lugar de nacimiento	fecha de bautismo	lugar de bautismo.parroquia	lugar de bautismo.dirección	lugar de bautismo.diócesis
---------------------	-------------------	-----------------------------	-----------------------------	----------------------------

padre	madre	registro civil	entidad	delegación	juzgado	acta	año	fecha de registro	libro
-------	-------	----------------	---------	------------	---------	------	-----	-------------------	-------

foja	localidad	clase	género	edad	estado civil	teléfono	religión	grado de estudios
------	-----------	-------	--------	------	--------------	----------	----------	-------------------

profesión u ocupación	parroquia a la que pertenece	primera comunión.recepción	primera comunión.edad	fecha de defunción	ID contrayente
-----------------------	------------------------------	----------------------------	-----------------------	--------------------	----------------

DOMICILIO

<u>ID Domicilio</u>	domicilio	tiempo de residencia	parroquia a la que pertenece	dirección de la parroquia a la que pertenece	ID Feligrés
---------------------	-----------	----------------------	------------------------------	--	-------------

PAGO

<u>ID Pago</u>	fecha	concepto.intención	concepto.matrimonio	concepto.xv años	concepto.aniversario
----------------	-------	--------------------	---------------------	------------------	----------------------

concepto.presentación	concepto.servicios notariales	concepto.otro	fecha de la misa	hora de la misa	nombre de quien recibió
-----------------------	-------------------------------	---------------	------------------	-----------------	-------------------------

total	a cuenta	resta	fecha de liquidación de pago
-------	----------	-------	------------------------------

BAUTISMO

<u>ID Feligrés</u>	nombre del padrino	nombre de la madrina	libro de bautismo
--------------------	--------------------	-------------------------	-------------------

foja en el libro de bautismo	acta en el libro de bautismo	notas marginales	sacerdote	párroco	fecha de expedición
---------------------------------	---------------------------------	---------------------	-----------	---------	------------------------

PRIMERA COMUNIÓN

<u>ID Feligrés</u>	padrino	madrina	fecha de primera comunión	sacerdote	párroco
--------------------	---------	---------	---------------------------------	-----------	---------

CONFIRMACIÓN

<u>ID Feligrés</u>	padrino	madrina	lugar de confirmación	fecha de confirmación	sacerdote	párroco
--------------------	---------	---------	--------------------------	--------------------------	-----------	---------

CONSAGRACIÓN

<u>ID Feligrés</u>	lugar de la ordenación o profesión religiosa	fecha de la ordenación o profesión religiosa
--------------------	---	---

sacerdote	párroco
-----------	---------

MATRIMONIO

<u>acta</u>	libro	foja	parroquia donde se realizaron los tramites	sacerdote	nombre del párroco
-------------	-------	------	---	-----------	-----------------------

fecha de primera publicación	fecha de segunda publicación	fecha de tercera publicación	fecha de primera plática	fecha de segunda plática	fecha de tercera plática
------------------------------------	------------------------------------	------------------------------------	--------------------------------	--------------------------------	--------------------------------

horario de platicas	número de expediente	lugar donde se celebrará el matrimonio	nombre de la parroquia donde se van a casar	dirección de la parroquia donde se van a casar
---------------------------	-------------------------	--	--	---

diócesis de la parroquia donde se van a casar	presentación completa	documentación completa	pláticas prematrimoniales
--	--------------------------	---------------------------	------------------------------

proclamas	parroquias a las que se envió exhortos	fecha de ratificación	dispensas.estado	dispensas.contenido
-----------	---	--------------------------	------------------	---------------------

licencias.est ado	licencias.conte nido	celebración.es tado	celebración.cont enido	notificación.es tado	notificación.cont enido
----------------------	-------------------------	------------------------	---------------------------	-------------------------	----------------------------

fecha de matrimonio civil	lugar de matrimonio civil	oficialía	libro civil	acta civil	número	lugar
---------------------------	---------------------------	-----------	-------------	------------	--------	-------

domicilio del padre o tutor	fecha de la presentación	hora	costo	aceptó
-----------------------------	--------------------------	------	-------	--------

edad del novio	edad de la novia	fecha de matrimonio	ID Feligrés
----------------	------------------	---------------------	-------------

EXHORTO

<u>ID Exhorto</u>	acta	asunto	fecha de expedición	nombre de la parroquia a la que se dirige	dirección de la parroquia a la que se dirige
-------------------	------	--------	---------------------	---	--

DECLARACIÓN

ID Declaración	acta	se casa usted por su propia y libre voluntad o hay alguien o algo que le obligue a casarse
----------------	------	--

tiene usted algún compromiso anterior con alguna otra persona relacionado con su matrimonio	con quién
---	-----------

qué tipo de compromiso	cuánto tiempo tienen de noviazgo	cómo ha sido su noviazgo	sabe usted las obligaciones y derechos del matrimonio
------------------------	----------------------------------	--------------------------	---

Está usted dispuesto a cumplir con esos derechos y obligaciones	Por qué quiere casarse por la Iglesia Católica	Sabe usted lo que significa que el Matrimonio sea un Sacramento
---	--	---

Sabe usted que el Matrimonio es una Alianza entre un hombre y una mujer y que es uno e indisoluble, perpetuo y exclusivo hasta la muerte	Se compromete a vivirlo así	Cómo considera usted su formación religiosa	Está usted dispuesto a vivir sus valores cristianos en la familia que va a constituir y en la Comunidad Parroquial
--	-----------------------------	---	--

Existe algún impedimento, o circunstancia física, psíquica, moral o religiosa por parte de usted o de el otro contrayente que se oponga a la celebración de este matrimonio	Cómo considera su estado de salud física y psíquica con relación a la nueva vida que pretende iniciar	Tiene usted algún vicio o adicción que pudiera trastornar la vida matrimonial	Sabe si el otro contrayente los tiene
---	---	---	---------------------------------------

TESTIGO

<u>ID Testigo</u>	acta	nombre del testigo	domicilio del testigo	teléfono del testigo	documento oficial de identificación del testigo
-------------------	------	--------------------	-----------------------	----------------------	---

SOLICITUD

<u>ID Solicitud</u>	acta	cumplo con mis deberes religiosos	conozco la doctrina cristiana	iniciamos nuestro noviazgo aproximadamente hace
---------------------	------	-----------------------------------	-------------------------------	---

mis padres están enterados de mi noviazgo y de mi decisión de contraer matrimonio	domicilio después de celebrado el matrimonio	fecha de solicitud
---	--	--------------------

PERMISO

<u>ID Permiso</u>	acta	fecha de permiso	nombre del presbítero delegado	nombre de la parroquia emisora	dirección de la parroquia emisora	diócesis de la parroquia emisora
-------------------	------	------------------	--------------------------------	--------------------------------	-----------------------------------	----------------------------------

fecha de inicio de tramites	presbítero que suscribe	nombre de la parroquia a su cargo	dirección de la parroquia a su cargo
-----------------------------	-------------------------	-----------------------------------	--------------------------------------

nombre del obispo de la diócesis de la parroquia donde se van a casar	tipo de permiso
---	-----------------

SERVICIO

<u>folio</u>	tipo
--------------	------

CRIPTA

<u>expediente</u>	número de expediente	nombre del propietario	dirección del propietario	teléfono del propietario	nombre del beneficiario	dirección del beneficiario
-------------------	----------------------	------------------------	---------------------------	--------------------------	-------------------------	----------------------------

teléfono del beneficiario	nicho	sección	nivel	pasillo	folio
---------------------------	-------	---------	-------	---------	-------

PAGOS DE CRIPTAS

<u>ID Pago cripta</u>	folio	cantidad	cantidad en letra	donativo inicial	mensualidades	saldo pendiente	numero de recibo
-----------------------	-------	----------	-------------------	------------------	---------------	-----------------	------------------

fecha	reservación	fecha inicio de pago	fecha de elaboración	dirección de la parroquia	teléfono de la parroquia	fecha de donativo	concepto	recibió
-------	-------------	----------------------	----------------------	---------------------------	--------------------------	-------------------	----------	---------

INTENCIÓN

<u>ID Intención</u>	folio	enunciado	tipo de intención	fecha	fecha y hora de la celebración
---------------------	-------	-----------	-------------------	-------	--------------------------------

DECRETO

<u>ID</u> <u>Decreto</u>	folio				dirección de la parroquia	libro a modificar	registro anterior.libro	registro anterior.foja
-----------------------------	-------	--	--	--	---------------------------------	----------------------	----------------------------	---------------------------

nuevo registro.libro	nuevo registro.foja	nuevo registro.acta	fecha de transcripción	fecha de entrega a la curia diocesana	correcciones	parroquia
-------------------------	------------------------	------------------------	---------------------------	--	--------------	-----------

Algunos nombres de atributos serán modificados al momento de ingresarlos al DBMS, debido a su extensión, datos muestra se encuentran en los anexos.

4.3 ESPECIFICACIÓN BASE DE DATOS DOCUMENTOS

El usuario ha solicitado una forma manual de generar los documentos, además, de la forma automática sugerida. Esto con la intención de utilizar los formatos que ya dispone la parroquia y llevar un registro de los mismos. Esta situación me ha obligado a generar otra base datos la cual llevará el nombre de "documentos", a continuación, presento los atributos que formarán parte de sus tablas componente.

4.3.1 Atributos base de datos documentos

Debido a su extensión y el hecho de que la mayoría son atributos antes presentados en la sección anterior, se presenta la lista de atributos en forma resumida.

documentos

- id_documento
- fecha
- tipo

aob (acta o boleta)

- id_aob
- fsacramento
- lsacramento
- sacerdote
- genero
- fnacimiento
- lnacimiento
- nombre
- nombre1
- padre
- madre
- padrino
- madrina
- contrayente
- pmatrimonio

- libro
- foja
- acta
- fexpedicion
- parroco
- parroquia_tramites
- ntestigo
- ntestigo1
- dom
- rc
- entidad
- delegacion
- juzgado
- actarc
- ano
- frc
- librorc
- fojarc
- localidad
- clase

edm (expediente de matrimonio)

- expediente
- apellidos
- novio
- tel
- novia
- telc
- sacerdote
- fecha
- lmatrimonio
- pmatrimonio
- dparroquia_matrimonio
- diopmatrimonio
- fmatrimonio
- pcompleta
- doccompleta
- plasticas
- proclamas
- fppublicacion
- fspublicacion
- ftpublicacion
- parroquias_exhorto
- dispensase
- dispensasc
- licenciase

- licenciasc
- celebracione
- celebracionc
- notificacione
- notificacionc
- ec
- enovio
- padre
- madre
- Inacimiento
- fnacimiento
- rel
- ge
- po
- dcalle
- dnum
- dcol
- dmun
- dedo
- dcp
- parroquia
- otrosdoms
- tiempodoms
- lbautismop
- lbautismod
- lbautismodio
- fbautismo
- librob
- fojab
- actab
- lconfirmacion
- fconfirmacion
- comunior
- comunione
- ecc
- enovioc
- padrec
- madrec
- Inacimientoc
- fnacimientoc
- relc
- gec
- poc
- dcallec
- dnumc
- dcolc

- dmunc
- dedoc
- dcpc
- parroquiac
- otrosdomsc
- tiempodomsc
- lbautismopc
- lbautismodc
- lbautismodioc
- fbautismoc
- librobc
- fojabc
- actabc
- lconfirmacionc
- fconfirmacionc
- comuniorc
- comunionec
- libre_consentimiento
- compromiso_anterior
- quien
- tcompromiso
- noviazgo_duracion
- calidad_noviazgo
- odmatrimonio
- disposicion
- razon
- conocimiento_sacramento
- cualidades_matrimonio
- compromiso
- formacion_religiosa
- disposicion_matrimonio
- impedimento
- salud
- vicios
- vicioso
- libre_consentimientoc
- compromiso_anteriorc
- quienc
- tcompromisoc
- noviazgo_duracionc
- calidad_noviazgoc
- odmatrimonioc
- disposicionc
- razonc
- conocimiento_sacramentoc
- cualidades_matrimonioc

- compromisoc
- formacion_religiosac
- disposicion_matrimonioc
- impedimientoc
- saludc
- viciosc
- viciosoc
- ntestigo
- dtestigo
- ttestigo
- didtestigo
- tccontrayente
- motivo
- eccontrayente
- lccontrayente
- otra_razon
- tquien
- obstaculo
- cual
- fdeclaracion_testigo
- ntestigo1
- dtestigo1
- ttestigo1
- didtestigo1
- tccontrayente1
- motivo1
- eccontrayente1
- lccontrayente1
- otra_razon1
- tquien1
- obstaculo1
- cual1
- fdeclaracion_testigo1
- ntestigo2
- dtestigo2
- ttestigo2
- didtestigo2
- tccontrayente2
- motivo2
- eccontrayente2
- lccontrayente2
- otra_razon2
- tquien2
- obstaculo2
- cual2
- fdeclaracion_testigo2

- ntestigo3
- dtestigo3
- ttestigo3
- didtestigo3
- tccontrayente3
- motivo3
- eccontrayente3
- lccontrayente3
- otra_razon3
- tquien3
- obstaculo3
- cual3
- fdeclaracion_testigo3
- cumplimiento
- conocimiento
- duracion_noviazgo
- padres_conocimiento
- domicilio_futuro
- fsol
- cumplimiento1
- conocimiento1
- duracion_noviazgo1
- padres_conocimiento1
- domicilio_futuro1
- fsol1
- frat
- dpadre
- librom
- fojam
- actam
- fpplatica
- fsplatica
- ftplatica
- horario
- fmatrimonioc
- lmatrimonioc
- lc
- fc
- ac
- oficialia
- fpresentacion
- hora
- costo
- acepto
- fexpedicione
- asunto

- preceptora
- dpreceptora
- fexpedicione1
- asunto1
- preceptora1
- dpreceptora1
- fexpedicione2
- asunto2
- preceptora2
- dpreceptora2
- fpermiso
- presbitero_delegado
- parroquia_emisora
- dparroquia_emisora
- dioparroquia_emisora
- finicio_tramites
- presbitero_sus
- nparroquia
- dparroquia
- obispodioparroquia
- tipo_permiso
- fpermiso1
- presbitero_delegado1
- parroquia_emisora1
- dparroquia_emisora1
- dioparroquia_emisora1
- finicio_tramites1
- presbitero_sus1
- nparroquia1
- dparroquia1
- obispodioparroquia1
- tipo_permiso1
- fpermiso2
- presbitero_delegado2
- parroquia_emisora2
- dparroquia_emisora2
- dioparroquia_emisora2
- finicio_tramites2
- presbitero_sus2
- nparroquia2
- dparroquia2
- obispodioparroquia2
- tipo_permiso2
- documentos_id_documentos

4.3.2 Diagrama E-R de la base de datos documentos

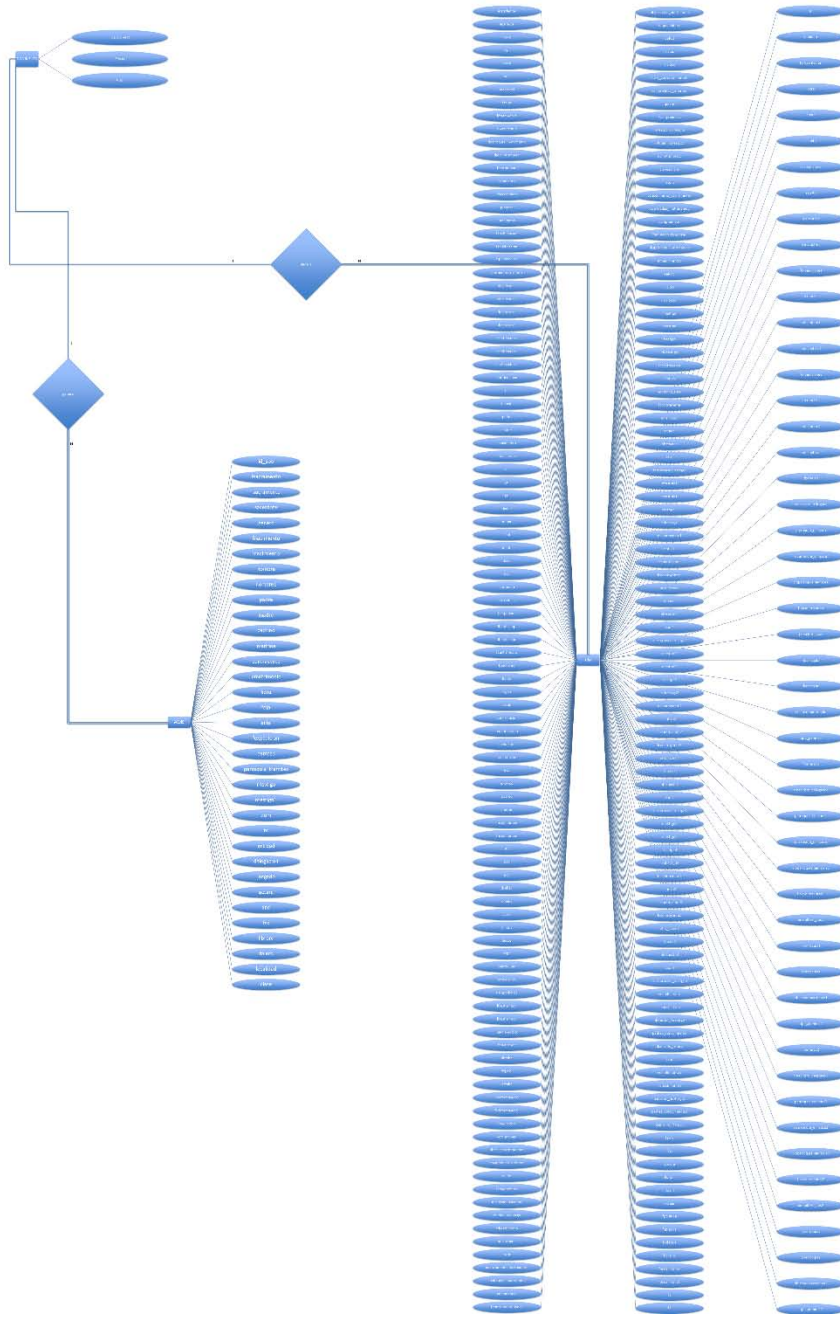


Figura 4.2 Diagrama E-R de la base de datos documentos

En los anexos se encuentra una imagen con mayor resolución, además diagramas E-R de sus entidades componentes, esta es tan sólo para ayudar al lector a visualizar el esquema de bases de datos resultante de emplear la metodología de Bagui para diseño de bases de datos.

4.3.3 Gramáticas base de datos documentos

Documentos

Esta base de datos registra datos sobre DOCUMENTOS. Por cada DOCUMENTO se registra un id_documento, una fecha y un tipo.

Atributos

Por cada DOCUMENTO, se tendrá solamente un id_documento, una fecha y un tipo. Ninguno de estos atributos será subdividido.

Clave

Por cada DOCUMENTO, se tendrá la siguiente clave principal: id_documento.

Aob

Esta base de datos registra datos sobre AOBS. Por cada AOB, se registra un id_aob, fsacramento, lsacramento, sacerdote, genero, fnacimiento, lnacimiento, nombre, nombre1, padre, madre, padrino, madrina, contrayente, pmatrimonio, libro, foja, acta, fexpedicion, parroco, parroquia_tramites, ntestigo, ntestigo1, dom , rc, entidad, delegacion, juzgado, actarc, ano, frc, librorc, fojarc, localidad y clase.

Atributos

Por cada AOB, se tendrá solamente un id_aob, fsacramento, lsacramento, sacerdote, genero, fnacimiento, lnacimiento, nombre, nombre1, padre, madre, padrino, madrina, contrayente, pmatrimonio, libro, foja, acta, fexpedicion, parroco, parroquia_tramites, ntestigo, ntestigo1, dom , rc, entidad, delegacion, juzgado, actarc, ano, frc, librorc, fojarc, localidad y clase. Ninguno de estos atributos será subdividido.

Clave

Por cada AOB, se tendrá la siguiente clave principal: id_aob.

Relaciones

Los DOCUMENTOS, pero no necesariamente todos los FELIGRESES (que están registrados en la base de datos), podría generar muchas (una o más) AOBS. Algunos DOCUMENTOS podrían no ser una AOB.

Las AOBS, que están registradas en la base de datos, deben ser generadas por solamente un DOCUMENTO.

Edm

Esta base de datos registra datos sobre EDMS. Por cada EDM, se registra expediente, apellidos, novio, tel, novia, telc, sacerdote, fecha, lmatrimonio, pmatrimonio,

dparroquia_matrimonio, diopmatrimonio, fmatrimonio, pcompleta, doccompleta, platicas, proclamas, fppublicacion, fspublicacion, ftpublicacion, parroquias_exhorto, dispensase, dispensasc, licenciase, licenciasc, celebracione, celebracionc, notificacione, notificacionc, ec, enovio, padre, madre, Inacimiento, fnacimiento, rel, ge, po, dcalle, dnum, dcol, dmun, dedo, dcp, parroquia, otrosdoms, tiempodoms, lbautismop, lbautismod, lbautismodio, fbautismo, librob, fojab, actab, lconfirmacion, fconfirmacion, comunionr, comunione, ecc, enovioc, padrec, madrec, Inacimientoc, fnacimientoc, relc, gec, poc, dcallec, dnumc, dcolc, dmunc, dedoc, dcpc, parroquiac, otrosdomsc, tiempodomsc, lbautismopc, lbautismodc, lbautismodioc, fbautismoc, librob, fojab, actab, lconfirmacionc, fconfirmacionc, comunionrc, comunione, libre_consentimiento, compromiso_anterior, quien, tcompromiso, noviazgo_duracion, calidad_noviazgo, odmatrimonio, disposicion, razon, conocimiento_sacramento, cualidades_matrimonio, compromiso, formacion_religiosa, disposicion_matrimonio, impedimento, salud, vicios, vicioso, libre_consentimientoc, compromiso_anteriorc, quient, tcompromisoc, noviazgo_duracionc, calidad_noviazgoc, odmatrimonioc, disposicionc, razonc, conocimiento_sacramentoc, cualidades_matrimonioc, compromisoc, formacion_religiosac, disposicion_matrimonioc, impedimientoc, saludc, viciosc, viciosoc, ntestigo, dtestigo, ttestigo, didtestigo, tcontrayente, motivo, econtrayente, lcontrayente, otra_razon, tquien, obstaculo, cual, fdeclaracion_testigo, ntestigo1, dtestigo1, ttestigo1, didtestigo1, tcontrayente1, motivo1, econtrayente1, lcontrayente1, otra_razon1, tquien1, obstaculo1, cual1, fdeclaracion_testigo1, ntestigo2, dtestigo2, ttestigo2, didtestigo2, tcontrayente2, motivo2, econtrayente2, lcontrayente2, otra_razon2, tquien2, obstaculo2, cual2, fdeclaracion_testigo2, ntestigo3, dtestigo3, ttestigo3, didtestigo3, tcontrayente3, motivo3, econtrayente3, lcontrayente3, otra_razon3, tquien3, obstaculo3, cual3, fdeclaracion_testigo3, cumplimiento, conocimiento, duracion_noviazgo, padres_conocimiento, domicilio_futuro, fsol, cumplimiento1, conocimiento1, duracion_noviazgo1, padres_conocimiento1, domicilio_futuro1, fsol1, frat, dpadre, librom, fojam, actam, fpplatica, fsplatica, ftplatica, horario, fmatrimonioc, lmatrimonioc, lc, fc, ac, oficialia, fpresentacion, hora, costo, acepto, fexpedicion, asunto, preceptora, dpreceptora, fexpedicion1, asunto1, preceptora1, dpreceptora1, fexpedicion2, asunto2, preceptora2, dpreceptora2, fpermiso, presbitero_delegado, parroquia_emisora, dparroquia_emisora, dioparroquia_emisora, finicio_tramites, presbitero_sus, nparroquia, dparroquia, obispodioparroquia, tipo_permiso, fpermiso1, presbitero_delegado1, parroquia_emisora1, dparroquia_emisora1, dioparroquia_emisora1, finicio_tramites1, presbitero_sus1, nparroquia1, dparroquia1, obispodioparroquia1, tipo_permiso1, fpermiso2, presbitero_delegado2, parroquia_emisora2, dparroquia_emisora2, dioparroquia_emisora2, finicio_tramites2, presbitero_sus2, nparroquia2, dparroquia2, obispodioparroquia2 y tipo_permiso2.

Atributos

Por cada EDM, se tendrá solamente un expediente, apellidos, novio, tel, novia, telc, sacerdote, fecha, lmatrimonio, pmatrimonio, dparroquia_matrimonio, diopmatrimonio, fmatrimonio, pcompleta, doccompleta, platicas, proclamas, fppublicacion, fspublicacion, ftpublicacion, parroquias_exhorto, dispensase, dispensasc, licenciase, licenciasc, celebracione, celebracionc, notificacione, notificacionc, ec, enovio, padre, madre, Inacimiento, fnacimiento, rel, ge, po, dcalle, dnum, dcol, dmun, dedo, dcp, parroquia, otrosdoms, tiempodoms, lbautismop, lbautismod, lbautismodio, fbautismo, librob, fojab, actab, lconfirmacion, fconfirmacion, comunionr, comunione, ecc, enovioc, padrec, madrec, Inacimientoc, fnacimientoc, relc, gec, poc, dcallec, dnumc, dcolc, dmunc, dedoc, dcpc, parroquiac, otrosdomsc, tiempodomsc, lbautismopc, lbautismodc, lbautismodioc, fbautismoc, librob, fojab, actab, lconfirmacionc, fconfirmacionc, comunionrc, comunione,

libre_consentimiento, compromiso_anterior, quien, tcompromiso, noviazgo_duracion, calidad_noviazgo, odmatrimonio, disposicion, razon, conocimiento_sacramento, cualidades_matrimonio, compromiso, formacion_religiosa, disposicion_matrimonio, impedimento, salud, vicios, vicioso, libre_consentimentoc, compromiso_anteriorc, quienc, tcompromisoc, noviazgo_duracionc, calidad_noviazgoc, odmatrimonioc, disposicionc, razonc, conocimiento_sacramentoc, cualidades_matrimonioc, compromisoc, formacion_religiosac, disposicion_matrimonioc, impedimentoc, saludc, viciosc, viciosoc, ntestigo, dtestigo, ttestigo, didtestigo, tcontrayente, motivo, econtrayente, lcontrayente, otra_razon, tquien, obstaculo, cual, fdeclaracion_testigo, ntestigo1, dtestigo1, ttestigo1, didtestigo1, tcontrayente1, motivo1, econtrayente1, lcontrayente1, otra_razon1, tquien1, obstaculo1, cual1, fdeclaracion_testigo1, ntestigo2, dtestigo2, ttestigo2, didtestigo2, tcontrayente2, motivo2, econtrayente2, lcontrayente2, otra_razon2, tquien2, obstaculo2, cual2, fdeclaracion_testigo2, ntestigo3, dtestigo3, ttestigo3, didtestigo3, tcontrayente3, motivo3, econtrayente3, lcontrayente3, otra_razon3, tquien3, obstaculo3, cual3, fdeclaracion_testigo3, cumplimiento, conocimiento, duracion_noviazgo, padres_conocimiento, domicilio_futuro, fsol, cumplimiento1, conocimiento1, duracion_noviazgo1, padres_conocimiento1, domicilio_futuro1, fsol1, frat, dpadre, librom, fojam, actam, fpplatica, fsplatica, ftplatica, horario, fmatrimonioc, lmatrimonioc, lc, fc, ac, oficialia, fpresentacion, hora, costo, acepto, fexpedicione, asunto, preceptora, dpreceptora, fexpedicione1, asunto1, preceptora1, dpreceptora1, fexpedicione2, asunto2, preceptora2, dpreceptora2, fpermiso, presbitero_delegado, parroquia_emisora, dparroquia_emisora, dioparroquia_emisora, finicio_tramites, presbitero_sus, nparroquia, dparroquia, obispodioparroquia, tipo_permiso, fpermiso1, presbitero_delegado1, parroquia_emisora1, dparroquia_emisora1, dioparroquia_emisora1, finicio_tramites1, presbitero_sus1, nparroquia1, dparroquia1, obispodioparroquia1, tipo_permiso1, fpermiso2, presbitero_delegado2, parroquia_emisora2, dparroquia_emisora2, dioparroquia_emisora2, finicio_tramites2, presbitero_sus2, nparroquia2, dparroquia2, obispodioparroquia2 y tipo_permiso2. Ninguno de estos atributos será subdividido.

Clave

Por cada EDM, se tendrá la siguiente clave principal: id_edm.

Relaciones

Los DOCUMENTOS, pero no necesariamente todos los FELIGRESES (que están registrados en la base de datos), podría generar muchos (uno o más) EDMS. Algunos DOCUMENTOS podrían no ser un EDM.

Los EDMS, que están registradas en la base de datos, deben ser generado por solamente un DOCUMENTO.

4.3.4 Base de datos relacional de documentos

documentos

id_documento	fecha	tipo
--------------	-------	------

aob

id_aob	fsacramento	lsacramento	sacerdote	genero	fnacimiento	lnacimiento
--------	-------------	-------------	-----------	--------	-------------	-------------

nombre	nombre1	padre	madre	padrino	madrina	contrayente
--------	---------	-------	-------	---------	---------	-------------

pmatrimonio	libro	foja	acta	fexpedicion	parroco	parroquia_tramites
-------------	-------	------	------	-------------	---------	--------------------

ntestigo	ntestigo1	dom	rc	entidad	delegacion	juzgado
----------	-----------	-----	----	---------	------------	---------

actarc	ano	frc	librorc	fojarc	localidad	clase	id_documento
--------	-----	-----	---------	--------	-----------	-------	--------------

edm

expediente	apellidos	novio	tel	novia	telc	sacerdote
------------	-----------	-------	-----	-------	------	-----------

fecha	lmatrimonio	pmatrimonio	dparroquia_matrimonio	diopmatrimonio	fmatrimonio	pcompleta
-------	-------------	-------------	-----------------------	----------------	-------------	-----------

doccompleta	platicas	proclamas	fppublicacion	fspublicacion	ftpublicacion	parroquias_exhorto
-------------	----------	-----------	---------------	---------------	---------------	--------------------

dispensase	dispensasc	licenciase	licenciasc	celebracione	celebracionc	notificacione
------------	------------	------------	------------	--------------	--------------	---------------

notificacionc	ec	enovio	padre	madre	lnacimiento	fnacimiento
---------------	----	--------	-------	-------	-------------	-------------

rel	ge	po	dcalle	dnum	dcol	dmun
-----	----	----	--------	------	------	------

dedo	dcp	parroquia	otrosdoms	tiempodoms	lbautismop	lbautismod
------	-----	-----------	-----------	------------	------------	------------

lbautismodio	fbautismo	librob	fojab	actab	lconfirmacion	fconfirmacion
--------------	-----------	--------	-------	-------	---------------	---------------

comunionr	comunione	ecc	enovioc	padrec	madrec	lnacamientoc
-----------	-----------	-----	---------	--------	--------	--------------

fnacamientoc	relc	gec	poc	dcallec	dnumc	dcolc
--------------	------	-----	-----	---------	-------	-------

dmunc	dedoc	dcpc	parroquiac	otrosdomsc	tiempodomsc	lbautismopc
-------	-------	------	------------	------------	-------------	-------------

lbautismodc	lbautismodioc	fbautismoc	librobc	fojabc	actabc	lconfirmacionc
-------------	---------------	------------	---------	--------	--------	----------------

fconfirmacionc	comunionrc	comunionec	libre_consentimiento	compromiso_anterior	quien	tcompromiso
----------------	------------	------------	----------------------	---------------------	-------	-------------

noviazgo_duracion	calidad_noviazgo	odmatrimonio	disposicion	razon	conocimiento_sacramento	cualidades_matrimonio
-------------------	------------------	--------------	-------------	-------	-------------------------	-----------------------

compromisoa	formacion_religiosa	disposicion_matrimonio	impedimient	o	salud	vicios	vicioso
-------------	---------------------	------------------------	-------------	---	-------	--------	---------

libre_consentimientoc	compromiso_anteriorc	quien	tcompromisoc	noviazgo_duracionc	calidad_noviazgoc	odmatrimonioc
-----------------------	----------------------	-------	--------------	--------------------	-------------------	---------------

disposicionc	razonc	conocimiento_sacramentoc	cualidades_matrimonioc	compromisoc	formacion_religiosac	disposicion_matrimonioc
--------------	--------	--------------------------	------------------------	-------------	----------------------	-------------------------

impedimientoc	saludc	viciosc	viciosoc	ntestigo	dtestigo	ttestigo
---------------	--------	---------	----------	----------	----------	----------

didtestigo	tcontrayente	motivo	eccontrayente	lcontrayente	otra_razon	tquien
------------	--------------	--------	---------------	--------------	------------	--------

obstaculo	cual	fdeclaracion_testigo	ntestigo1	dtestigo1	ttestigo1	didtestigo1
-----------	------	----------------------	-----------	-----------	-----------	-------------

tcontrayente1	motivo1	eccontrayente1	lcontrayente1	otra_razon1	tquien1	obstaculo1
---------------	---------	----------------	---------------	-------------	---------	------------

cual1	fdeclaracion_testigo1	ntestigo2	dtestigo2	ttestigo2	didtestigo2	tcontrayente2
-------	-----------------------	-----------	-----------	-----------	-------------	---------------

motivo2	eccontrayente2	lcontrayente2	otra_razon2	tquien2	obstaculo2	cual2
---------	----------------	---------------	-------------	---------	------------	-------

fdeclaracion_testigo2	ntestigo3	dtestigo3	ttestigo3	didtestigo3	tcontrayente3	motivo3
-----------------------	-----------	-----------	-----------	-------------	---------------	---------

eccontrayente3	lcontrayente3	otra_razon3	tquien3	obstaculo3	cual3	fdeclaracion_testigo3
----------------	---------------	-------------	---------	------------	-------	-----------------------

cumplimiento	conocimiento	duracion_noviazgo	padres_conocimiento	domicilio_futuro	fsol	cumplimiento1
--------------	--------------	-------------------	---------------------	------------------	------	---------------

conocimiento1	duracion_noviazgo1	padres_conocimiento1	domicilio_futuro1	fsol1	frat	dpadre
---------------	--------------------	----------------------	-------------------	-------	------	--------

librom	fojam	actam	fpplatica	fsplatica	ftplatica	horario
--------	-------	-------	-----------	-----------	-----------	---------

fmatrimonio1	lmatrimonio1	lc	fc	ac	oficialia	fpresentacion
--------------	--------------	----	----	----	-----------	---------------

hora	costo	acepto	fexpedicion1	asunto	preceptora	dpreceptora
------	-------	--------	--------------	--------	------------	-------------

fexpedicion1	asunto1	preceptora1	dpreceptora1	fexpedicion2	asunto2	preceptora2
--------------	---------	-------------	--------------	--------------	---------	-------------

dpreceptor1	fpermiso	presbitero_delegado	parroquia_emisor1	dparroquia_emisor1	dioparroquia_emisor1	finicio_tramites
-------------	----------	---------------------	-------------------	--------------------	----------------------	------------------

presbitero_sus	nparroquia	dparroquia	obispodioparroquia	tipo_permiso	fpermiso1	presbitero_delegado1
----------------	------------	------------	--------------------	--------------	-----------	----------------------

parroquia_emisor1	dparroquia_emisor1	dioparroquia_emisor1	finicio_tramites1	presbitero_sus1	nparroquia1	dparroquia1
-------------------	--------------------	----------------------	-------------------	-----------------	-------------	-------------

obispodioparroquia1	tipo_permiso1	fpermiso2	presbitero_delegado2	parroquia_emisor2	dparroquia_emisor2	dioparroquia_emisor2
---------------------	---------------	-----------	----------------------	-------------------	--------------------	----------------------

finicio_tramites2	presbitero_sus2	nparroquia2	dparroquia2	obispodioparroquia2	tipo_permiso2	id_documento
-------------------	-----------------	-------------	-------------	---------------------	---------------	--------------

4.4 ESPECIFICACIÓN BASE DE DATOS USUARIOS

Los usuarios que tienen permitido usar el sistema son aquellos con acceso a las computadoras de la oficina parroquial, por lo tanto, se agrega una base de datos que almacena a los usuarios.

4.4.1 Atributos base de datos usuarios

usuarios

- usuario
- contraseña

4.4.2 Diagrama E-R de la base de datos usuarios

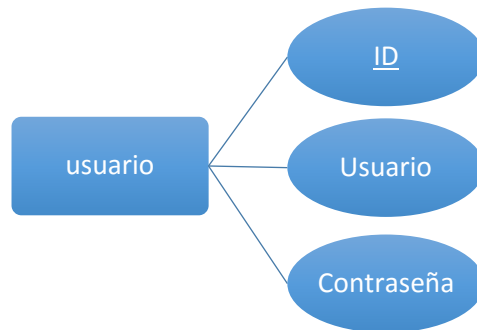


Figura 4.3 Diagrama E-R de la base de datos usuarios

4.4.3 Gramáticas base de datos usuarios

Usuarios

Esta base de datos registra datos sobre USUARIOS. Por cada USUARIO, se registra un usuario y una contraseña.

Atributos

Por cada USUARIO, se tendrá solamente un usuario y una contraseña. Ninguno de estos atributos será subdividido.

Clave

Por cada USUARIO, se tendrá la siguiente clave principal: id.

4.4.4 Base de datos relacional de usuarios

usuarios

usuario	contraseña
---------	------------

4.5 COMENTARIOS Y RECOMENDACIONES

Es necesario considerar herramientas de software que permitan realizar diagrama E-R de forma eficiente, una herramienta bastante útil es SmartArt de Microsoft Office, con ellos prácticamente se necesita una lista tabulada y unos cambios de formas. Existen alternativas web y tipo escritorio que fueron consideradas (Draw.io, Concept Draw, Edraw, Drawing, etc.) pero SmartArt es la mejor recomendación que se puede hacer si se utiliza la metodología de Bagui (2011, traducción propia) para diseño de bases de datos.

Se debe considerar que el grado de complejidad de una base datos puede ser bastante alto, el acompañar con las gramáticas nuestros esquemas manifiesta las buenas prácticas en cuanto a la generación de la especificación o documentación, pero al momento de presentarla al usuario bastará con proporcionar tan sólo una parte de ella, dado que el usuario tiene tiempo limitado y que además uno se da cuenta en el momento de que el usuario ha entendido o no lo que se le presenta.

Es necesario considerar la semántica al utilizar las gramáticas y hacer los cambios necesarios para que sea fácil de entender por el usuario, sugerencia de Bagui (2011, traducción propia) al proporcionar las mismas.

CAPÍTULO 5

DISEÑO DEL SISTEMA

5.1 DISEÑO GENERAL

El sistema de información se puede ver como una aplicación multinivel (multitier), el nivel backend (backend tier) donde se realizan las conexiones a bases de datos o sistemas legados, el nivel medio (middle tier) donde se realizan las transacciones u operaciones con la información obtenida de las bases de datos y el nivel frontend (frontend tier) donde el usuario interactúa con la aplicación mediante una interfaz desarrollada en un framework, en este caso varios frameworks web (Gupta 2013, traducción propia).

Existen varios stacks de desarrollo (conjunto de herramientas de software) que se pueden utilizar para desarrollar aplicaciones web, así como plataformas de desarrollo y patrones de diseño. Un stack muy conocido es LAMP que es un acrónimo de Linux, Apache, MySQL y PHP, es decir, sistema operativo utilizado, servidor web, DBMS y lenguaje utilizando en el backend. El mismo stack, pero en distintos sistemas operativos es WAMP y XAMP. Estos comentarios previos son debido a que entre desarrolladores es común utilizar el stack como punto de referencia a lo largo del proceso de ingeniería de software.

Uno actualmente de moda es MEAN, acrónimo de MongoDB, Express.js, Angular.js, y Node.js. Un stack híbrido menos popular es MyENJs, puede que se esté inventando o sea conocido de otra manera, con este acrónimo se hace referencia a MySQL, Express.js, Node.js, y JavaScript. Se le llama híbrido porque comúnmente Node.js se utiliza junto a una base de datos no relacional como MongoDB, pero a lo largo de este proyecto, MySQL será el DBMS que se empleará. Con respecto a las otras letras de él acrónimo, Express.js será el framework que se usará para el middle tier, Node.js será el ambiente de ejecución y servidor web, en el backend se utilizará JavaScript como lenguaje principal, y para la interfaz en el frontend se beneficiará de Stylus, Bootstrap y Jade que facilitan el uso e implementación de HTML5 y CSS.

Cuando se está inmerso en el ámbito de desarrollo de software se sabe que existen “patrones”, los cuales son las mejores formas en las que se aborda o soluciona un problema específico, es decir, un problema se presentó y se resolvió de muchas maneras, la mejor fue adoptada más y más hasta hacerla popular, en la popularidad revisada, optimizada y estandarizada para su implementación (Jones 2014, traducción propia). En el campo del desarrollo web existen plataformas de desarrollo basadas en estándares (un patrón, a fin de cuentas) como Java SE (Java Standard Edition), Java EE (Java Enterprise Edition) que se auxilian del MVC (Model View Controller) -otro patrón-, este último que se a portado al desarrollo móvil y otras áreas.

Brevemente se explica el patrón MVC (Modelo Vista Controlador) que fue utilizado para la aplicación y cada pequeño modulo en ella, acorde con npm (2015) o middleware según Express project (2015).

El “Modelo” hace alusión a las clases, métodos, funciones o módulos que generan la infraestructura de la aplicación (Jones 2014, traducción propia).

La “Vista” se refiere a la interfaz de interacción o presentación de datos obtenidos mediante el uso del “Controlador” que es la ejecución o llamada de las funciones o módulos que forman el “Modelo” (Jones 2014, traducción propia).

Haciendo énfasis en el sistema con el stack antes mencionado, el patrón MVC está presente en los módulos o middlewares que se definen y utilizan al implementar el framework Express.js, la interfaz de interacción que se desarrolló utilizando Stylus, Jade, Bootstrap, CSS y HTML5 y como se controla todo mediante el uso de JavaScript.

5.2 HERRAMIENTAS DE SOFTWARE UTILIZADAS EN DISEÑO DE SOFTWARE

Microsoft PowerPoint fue el software que se empleó para generar los diagramas según el estilo de Chen que se encuentran en los capítulos anteriores y en los anexos.

MySQL Workbench “es una herramienta visual unificada para arquitectos de bases de datos, desarrolladores y DBAs. . . proporciona modelado de datos, desarrollo SQL y herramientas completas para la configuración del servidor, administración de usuarios, respaldo y más” (MySQL 2015, traducción propia).

Como el DBMS que se usará será MySQL, se decidió utilizar la herramienta antes mencionada para agregar a los diagramas los tipos de datos soportados por el mismo, así como, el código necesario para crear las bases de datos, además de consultas y conexiones a las bases de datos.

Con respecto al diseño de vistas y controladores se empleó Sublime Text, que es “un sofisticado editor de texto para codificar, etiquetar y escritura en prosa” (Sublime HQ, 2015).

5.3 DIAGRAMAS ACORDE A DBMS

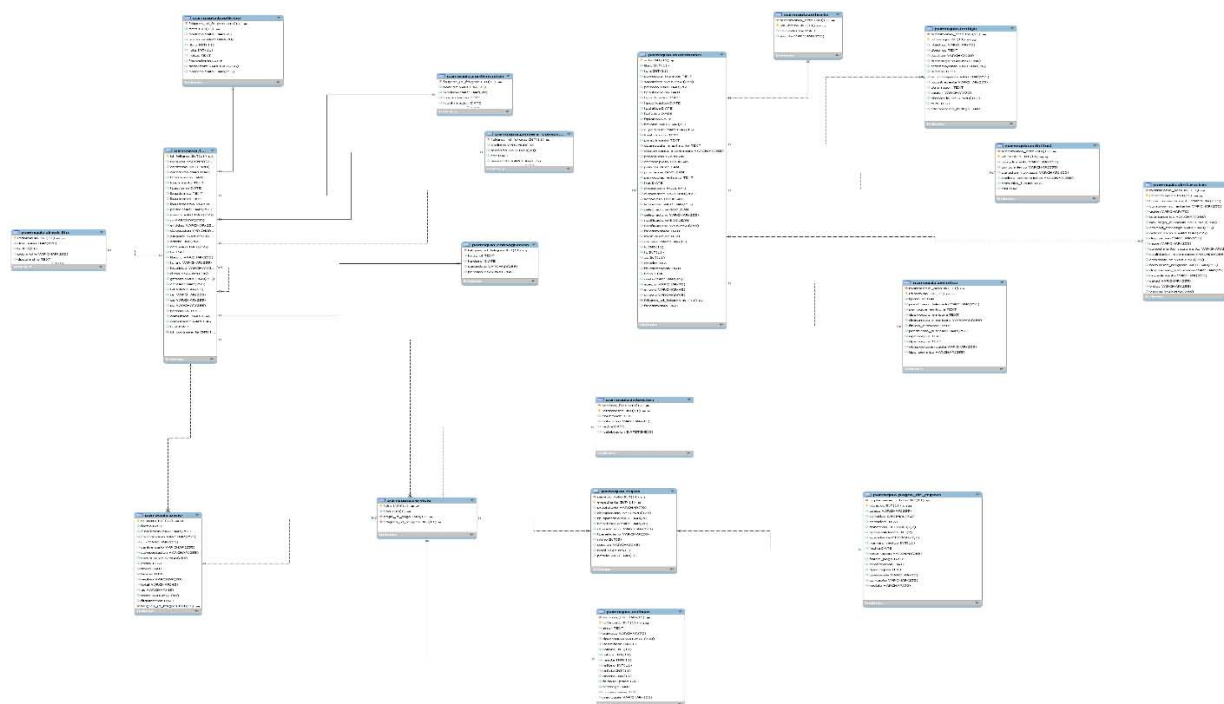


Figura 5.1 Esquema de la base de datos “parroquia” elaborado en MySQL Workbench

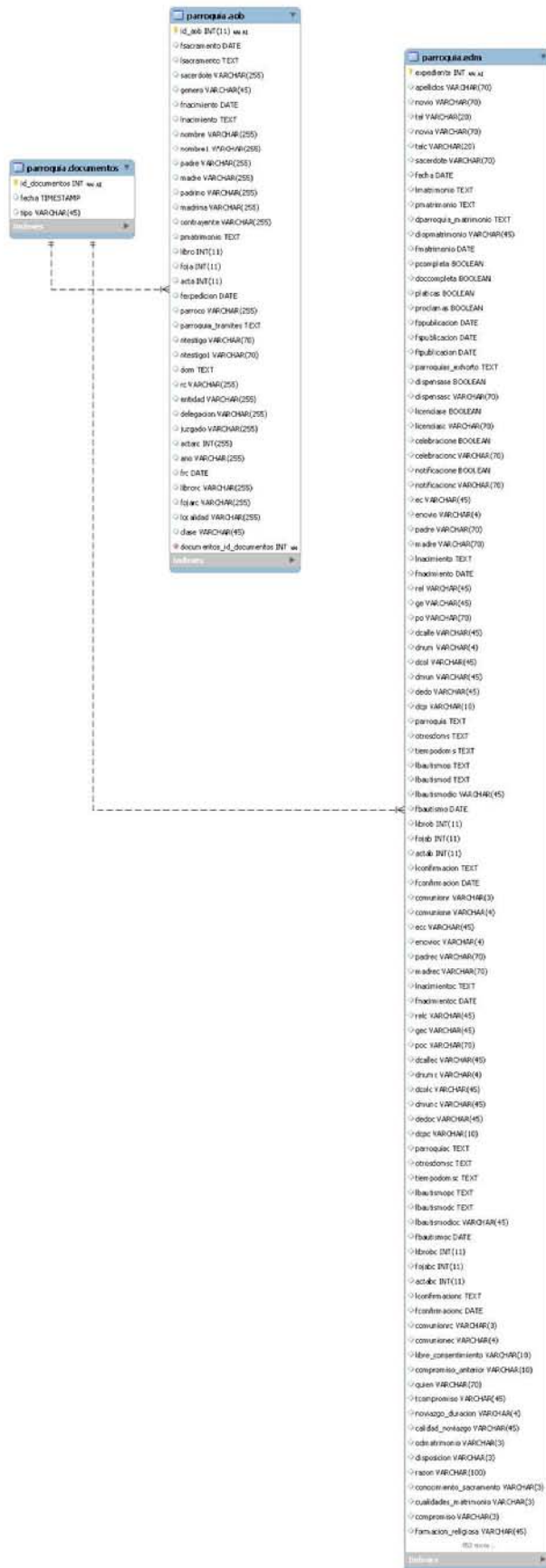


Figura 5.2 Esquema de la base de datos documentos elaborado en MySQL Workbench

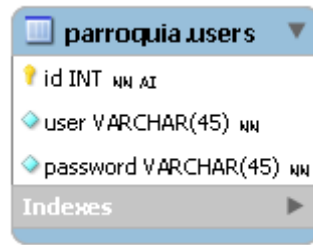


Figura 5.3 Esquema de la base de datos usuarios elaborado en MySQL Workbench

5.4 NOTAS DE DISEÑO

Estos diagramas son el resultado de la implementación constante del paso 3a mencionado en el primer capítulo, fueron ajustados constantemente para cumplir con la especificación.

Los diagramas presentados tan sólo son una adaptación de los diagramas obtenidos después de aplicar la metodología de Bagui con respecto al diseño de bases de datos que se encuentran en el estilo Chen, cuyas versiones completas y con sus componentes se encuentran en los anexos para mayor apreciación del lector.

En los anexos se encuentran también sus tablas componente para una mejor apreciación, así como, las imágenes anteriores en mayor resolución y en forma completa.

5.5 COMENTARIOS Y RECOMENDACIONES

Hay ocasiones en las que el usuario deja al analista todo con respecto al diseño general, sólo desea una forma de realizar sus tareas y no le interesa si el sistema o la aplicación será tipo escritorio o web, por lo cual es buena oportunidad para libremente elegir el tipo de aplicación según sea el dominio o experiencia en determinado campo. Si por el otro lado el usuario sugiere el uso de cierta tecnología o dependencia a cierto sistema operativo, se le pueden hacer sugerencias u orientarlo al respecto.

Una ventaja de los diagramas tipo Chen obtenidos de haber empleado la metodología de Bagui para diseño de bases de datos, es que permanecen inalterables sin importar que herramienta de diseño visual de bases de datos utilizemos para generar diagramas o código ejecutable orientado a un DBMS específico.

Revisar material de referencia que plantea distintas plataformas de desarrollo, desarrollo de aplicaciones de escritorio y aplicaciones móviles, tanto en ebooks, así como, vídeo training aportó bastantes ideas con respecto al diseño del sistema, así que, es una sugerencia a todo aquel que se apoye en este trabajo para realizar algún proyecto. Al principio se había considerado utilizar la plataforma Java EE 7 para el sistema, pero al final se decidió utilizar un stack web con Node.js.

CAPÍTULO 6

DESARROLLO DEL SISTEMA

6.1 DESARROLLO GENERAL DEL SISTEMA

En el capítulo anterior se mencionó el stack y el patrón de diseño que se utilizó a lo largo de la fase de diseño y desarrollo, ahora brevemente se describe lo que se hizo en la etapa de desarrollo, las herramientas de software que se utilizaron, además, se presenta a modo de tutorial cómo se desarrolla la aplicación desde cero, el módulo de autenticación, como programar una unidad base del sistema (se le llama unidad base a una aplicación base que en conjunto genera el sistema) y una descripción general del proceso de generación de documentos.

El presente trabajo se limita a eso debido a que entrar en detalle con respecto a cada unidad base del sistema extendería demasiado la longitud del mismo y se considera que esta información es más que suficiente para manifestar como desarrollar cada componente del sistema utilizando la metodología de ingeniería de software presentada por Bagui (2011, traducción propia), ya que tan sólo son variantes en el código mediante programación en JavaScript con respecto a otras unidades de software del sistema.

Primero se define la unidad base, la cual será una “aplicación CRUD (Create, Read, Update, Delete), es decir, una aplicación que me permita CREAR, LEER, ACTUALIZAR Y BORRAR datos” (Nuzum 2014, traducción propia).

Después se hace una serie de bosquejos de lo que se quiere que el usuario vea, agregando routes o paths que se utilizarán, a este paso (Nuzum, 2014) le llama la planeación de la aplicación.

Finalmente se comienza a codificar, primero el modelo un archivo.js después la vista un archivo. jade y finalmente se hace el deployment (implementación) utilizando node, controlando todo internamente utilizando otros scripts o archivos .js o middlewares que son montados en express.js.

Antes de pasar a las siguientes secciones, se señala que cuando se inserte código, se presenten instrucciones, nombres de programas, módulos, versiones, paquetes, variables, objetos (o sus propiedades), nombres de archivos y directorios, el tamaño de fuente se reducirá a ocho conservando la misma fuente.

Además, para referirse a archivos que contengan JavaScript, Jade o algún otro lenguaje se hará mención de su extensión.

Por ejemplo:

“los archivos .js”

o

“los archivos .jade”

En algunas situaciones se generaliza cualquier tipo de archivo según su rol o alguna característica:

Por ejemplo:

“controlador.js”

o

“vista.jade”

Todo esto se hace con la intención de ayudar en la lectura de este capítulo.

6.2 HERRAMIENTAS DE SOFTWARE PARA DESARROLLO

Node.js ambiente de ejecución y desarrollo.

Node.js se enfoca en el desarrollo de aplicaciones web de alto rendimiento Syed (2014, traducción propia), Syed (2014, traducción propia) dedica varios capítulos para mostrar su diseño interno, aborda el problema de escalado de I/O y nos muestra en distintas ocasiones su popularidad y resultado de benchmarks. A lo largo de la etapa de desarrollo se utilizó la versión 0.12.7 de Node.

Además de los módulos que `express` (versión ~4.13.1) utiliza en su template, se empleó los siguientes:

`fs` para la interacción con el file system (sistema de archivos). A lo largo de la etapa de desarrollo se usó la versión 0.0.2.

`moment` para generar formatos personalizados en fechas. Se utilizó la versión ^2.10.6.

`mysql` para interactuar con el DBMS. Se empleó la versión ^2.9.0.

`passport` y `passport-local` para autenticación de usuarios. Se usaron las versiones ^0.3.0 y ^1.0.0 respectivamente.

`pdfkit` para la generación de archivos pdf. Se utilizó la versión ^0.7.1

`method-override` para anular métodos http y sustituirlos por otros. La versión ^2.3.5 fue empleada.

Una parte esencial de `node.js` es el servidor web, el cual deja a un lado el estilo tradicional utilizado por los servidores web de proceso por solicitud y pila de threads para manejar solicitudes con un solo thread (Syed 2014, traducción propia).

Una herramienta fundamental es `Sublime Text`, debido a que con ella se realizó todo el código del sistema, a excepción del generado por `MySQL Workbench`, `Express`, `Node` y cualquier otro paquete según `npm` (2015, traducción propia) o módulo. La versión 2.0.2 fue utilizada.

MySQL Workbench fue utilizado nuevamente pero ahora su principal objetivo fue establecer las conexiones a las bases de datos y pruebas a medida que una aplicación CRUD lo necesitaba. Se empleó la versión 6.3.

MySQL como DBMS fue instalado para ser utilizado a lo largo de toda esta etapa, es el software que permitió la creación de bases de datos e interacción con las mismas, utilizando el código generado por MySQL Workbench y los modelos y controladores (principalmente archivos .js). La versión 5.6 fue usada a lo largo de todo este proyecto.

6.3 DESCRIPCIÓN A MODO TUTORIAL

6.3.1 Preparando el ambiente de desarrollo, así como, nos los presenta (Nuzum 2014, traducción propia)

El stack es MyENJs, así que es necesario descargar e instalar MySQL(MariaDB), Node.js y MySQL Workbench.

6.3.1.1 Instalando Express.js

En una terminal o un cmd node (si se realiza en Windows), se instala express.js con la siguiente instrucción:

```
npm install -g express-generator
```

6.3.1.2 Creando directorio y árbol de directorios según express.js

Se crea un directorio que albergara el sistema de información o aplicación multinivel, en este caso se le denomina:

```
lupitas
```

Se utiliza la siguiente instrucción para generar una aplicación con el template de Express.js, el cual agregará Jade para utilizar HTML5 y Stylus para beneficiarse de CSS sin necesidad de preocuparse por etiquetas, paréntesis, llaves, corchetes y sintaxis necesaria para obtener código funcional:

```
express -css stylus
```

Se genera el siguiente árbol de directorios:

```
lupita
|-bin
|-models
|-node_modules
|-public
|-routes
|-views
app.js
package.json
```

Aquí hay dos archivos importantes `package.json` que contiene todas las dependencias y `script` de nuestra aplicación y `app.js` que es nuestra aplicación en sí (Nuzum 2014, traducción

propia).

El directorio `bin`, contiene la configuración del servidor web.

El directorio `models`, guardará los modelos que se definirán.

El directorio `node_modules`, almacenará todos los módulos que utilizara nuestra aplicación, cada vez que se instale un nuevo módulo vía el administrador de paquetes npm utilizando la opción `--save` o utilizando el archivo `package.json` dichos módulos serán agregados a este directorio con todos los archivos y directorios que los componen.

El directorio `public`, contendrá las hojas de estilo y sus templates, así como, los archivos `.js` que otorgarán funcionalidad a las vistas.

El directorio `routes`, almacenará los archivos `.js` que definen las rutas y las solicitudes http que son atendidas en ellas.

El directorio `views`, guardará los archivos `.jade` que compondrán las vistas de la aplicación.

6.3.1.3 Instalación de Bootstrap e integración al ambiente de desarrollo

Se descarga `Bootstrap` vía `GitHub` o en su sitio oficial, ya que estamos utilizando `CSS` para nuestras vistas, se elige la versión `css`.

Dentro de `public` hay dos directorios:

```
javascripts
stylesheets
```

En `javascripts` se guarda el archivo:

```
bootstrap.min.js
```

En `stylesheets` se almacenan los siguientes archivos:

```
bootstrap-theme.css
bootstrap.min.css
```

Se modifica el archivo `style.styl` con las siguientes líneas:

```
@import "bootstrap.min.css"
@import "bootstrap-theme.css"
```

De esta manera `Bootstrap` queda instalado. “`Bootstrap` es un framework `HTML`, `CSS`, `JS` que permite crear vistas de forma intuitiva y elegante” (`Bootstrap 2015`, traducción propia).

En el archivo `layout.jade` se agrega el siguiente código:

```
doctype html
html
  head
    meta(charset='utf-8')
    meta(name = 'viewport', content = 'width=device-width, initial-scale=1.0')
    meta(name = 'description', content = "")
    meta(name = 'author', content = "")

    title= title
    link(rel='stylesheet', href='/stylesheets/style.css')
  body
    .navbar.navbar-inverse.navbar-fixed-top
      .container
        .navbar-header
```

```
collapse')
        button.navbar-toggle(type='button', data-toggle='collapse', data-target='.navbar-
            span.sr-only Toggle Navigation
            span.icon-bar
            span.icon-bar
            span.icon-bar
        a.navbar-brand(href='/') Lupitas

    block content

    script(type="text/javascript", src="/javascripts/jquery.min.js")
    script(type="text/javascript", src="/javascripts/main.js")
    //solucion a dropdown que no funcionaba
    script(type="text/javascript", src="/javascripts/bootstrap.min.js")
```

Con el código anterior se ha creado una barra de navegación que estará disponible para todo la aplicación, sólo será necesario agregar al archivo `.jade` la instrucción `extend layout`.

Con las etiquetas `script` he agregado `jquery`, `main` y `bootstrap` para la funcionalidad que requiero en las vistas cuando se haga uso de `bootstrap`.

6.3.1.4 Instalación del módulo mysql

Basta utilizar las siguientes instrucciones para agregar el módulo `mysql` a la aplicación:

```
npm install mysql
```

La instrucción anterior instalará el modulo para el usuario que ha iniciado sesión.

Un método alternativo es agregar al archivo `package.json` la siguiente línea:

```
mysql: "*"
```

Con el asterisco se indica que se instalará cualquier versión (por lo general la más reciente).

También se puede especificar que versión se desea con una pequeña modificación:

```
mysql: "^2.9.0"
```

Otro método es agregar la opción `--save` a la siguiente instrucción:

```
npm install mysql --save
```

Al hacer esto desde el directorio de aplicación (el nivel más alto o `root`), se indica que se desea instalar el módulo y agregarlo al archivo `package.json` de la aplicación.

6.3.1.5 Definición del modelo que se encargará de las conexiones a bases de datos

Para automatizar el proceso de realizar conexiones a la base de datos se ha diseñado un modelo que se llama `mysql_config.js`, el archivo se encuentra en el directorio `models` y su contenido es el siguiente:

```
var mysql = require('mysql');
var mysql_config = {};
```



```

mysql_config.type = 'mysql';
mysql_config.charset = 'utf8';

// usuarios
mysql_config.username = usuario;
mysql_config.password = password;

//host
mysql_config.host = '127.0.0.1';

mysql_config.connection = mysql.createConnection({
  host: mysql_config.host,
  user: mysql_config.username,
  password: mysql_config.password,
  database: mysql_config.dbname
});

mysql_config.connection.connect( function(err, res){
  if(err){
    console.log('ERROR connecting to: ' + mysql_config.host + ' '+ err);
  } else {
    console.log('MySQLDB connected successfully to al conectar ' + mysql_config.host);
  }
});

module.exports = mysql_config;

```

Este código presenta la solicitud del módulo `mysql` con la instrucción:

```
var mysql = require('mysql');
```

La definición de un objeto `mysql_config`:

```
var mysql_config = {};
```

En sus propiedades se agrega la configuración del DBMS, conjunto de caracteres de la base de datos, el tipo, y las credenciales para realizar la conexión al mismo:

```

mysql_config.type = 'mysql';
mysql_config.charset = 'utf8';

mysql_config.username = usuario;
mysql_config.password = password;

mysql_config.host = '127.0.0.1';

```

Además, en una de ellas llamada `connection` se agrega el método para crear una conexión a DBMS:

```

mysql_config.connection = mysql.createConnection({
  host: mysql_config.host,
  user: mysql_config.username,
  password: mysql_config.password,
  database: mysql_config.dbname
});

```

Usando la propiedad antes mencionada se realiza la conexión a DBMS y se incluye como se manejará un error en caso de presentarse:

```

mysql_config.connection.connect( function(err, res){
  if(err){
    console.log('ERROR connecting to: ' + mysql_config.host + ' '+ err);
  } else {
    console.log('MySQLDB connected successfully to al conectar ' + mysql_config.host);
  }
}

```

```
});
```

Finalmente, el objeto `mysql_config` se agrega al objeto `module` para ponerlo a disposición de toda la aplicación:

```
module.exports = mysql_config;
```

6.3.1.6 Realizar queries (consultas) a las bases de datos

Después de haber definido modelo `mysql_config.js`, en cualquier archivo `controlador.js` que necesite una conexión a alguna base de datos bastará con agregar las siguientes líneas:

Para establecer la conexión con el DBMS:

```
var mysql_config = require("../models/mysql_config.js");
```

Para indicar que base de datos se utilizará:

```
mysql_config.connection.query('USE base_de_datos');
```

Para hacer un query a la base de datos antes seleccionada:

```
mysql_config.connection.query('query ', function(err,registros_obtenidos){
    if(err){
        "código en caso de error"
    }
    else{
        "código en caso de conexión y query exitosos"
    }
})
```

6.3.1.7 Instalar, iniciar e implementar la aplicación (o sistema)

Un paquete en `node` que utiliza el administrador de paquetes `npm` “es todo directorio que contiene un programa descrito por un archivo `package.json`” (npm 2015, traducción propia).

Es necesario instalar la aplicación para adquirir e instalar las dependencias definidas en el archivo `package.json` y de esa manera poder iniciar e implementar la aplicación utilizando `node` (Nuzum 2014, traducción propia). La siguiente instrucción instala la ampliación:

```
npm install
```

Para iniciar e implementar la aplicación usando el servidor web incluido en `node` (o definido por `express` o nosotros mismos), se utiliza la siguiente instrucción:

```
npm start
```

Nuestra aplicación ahora esta implementada en nuestro servidor web y puede ser utilizada a través de un browser, sólo tenemos que ingresar la siguiente dirección:

<http://localhost:3000>

La dirección anterior nos llevará al `index` de nuestra aplicación, en este caso nos llevaría al `login` del sistema.

6.3.2 Autenticación

Se podría utilizar una base de datos y una serie de funciones personalizadas, pero valiéndose de las bondades de `Express.js` se utilizará otros middlewares, `express-session` para una sesión cifrada con una credencial o vía `passport.js` que permite definir estrategias de autenticación y se encarga de la persistencia de sesión en conjunción con `express-mysql-session` u otro dependiendo el DBMS.

6.3.2.1 Instalación de passport.js

Instalo `passport.js` con la siguiente instrucción:

```
npm install passport --save
```

o utilizando el archivo `package.json`

```
"passport": "^0.3.0"
```

6.3.2.2 Creación del modelo de estrategias locales

Para agregar el módulo de estrategia local se agregan las siguientes líneas en el archivo `app.js`:

```
var LocalStrategy = require('passport-local').Strategy;
```

se indica que se utilizará en la sesión en esta aplicación con las siguientes líneas:

```
app.use(passport.initialize());
app.use(passport.session());
```

Se define el modelo de estrategias para registrar, iniciar sesión y conectarse a la base de datos de usuarios con el archivo `local_register_login.js`. Un proyecto de GitHub que pertenece a manjeshpv y la documentación oficial fueron de bastante ayuda. Este es el contenido del archivo antes mencionado:

```
// configuracion de estrategia local
var passport = require('passport');
var LocalStrategy = require('passport-local').Strategy;
var mysql_config = require('./mysql_config');

// configuracion para persistencia de sesion
passport.serializeUser(function(user, done){
  mysql_config.connection.query('USE parroquia');
  done(null, user.id);
});

passport.deserializeUser(function(id, done){
  mysql_config.connection.query('USE parroquia');
  mysql_config.connection.query("SELECT * from users where id = ?", [id], function(err, rows){
    done(err, rows[0]);
  });
});

//local register
passport.use('local-register', new LocalStrategy({
  usernameField: 'username',
  passwordField: 'password',
  passReqToCallback: true
```

```

    },
    function(req, username, password, done){

        mysql_config.connection.query('USE parroquia');
        mysql_config.connection.query("SELECT * from users where user = ?" ,[username ], function(err, rows){
            console.log(rows);
            console.log("Row in the database above.");

            if (err)
                return done(err);

            if(rows.length){
                return done(null, false, console.log('That user is already taken.));
            } else {

                var newUserMysql = {};

                newUserMysql.username = username;
                newUserMysql.password = password;

                var insertQuery = "INSERT INTO users (user, password) values (" + username + "," + password + ")";
                console.log(insertQuery);

                mysql_config.connection.query(insertQuery, function(err, rows){
                    newUserMysql.id= rows.insertId;

                    return done(null, newUserMysql);

                });
            }
        });
    });

// local login
passport.use('local-login', new LocalStrategy({
    usernameField: 'username',
    passwordField: 'password',
    passReqToCallback: true
}),
function(req, username, password, done){
    mysql_config.connection.query('USE parroquia');
    mysql_config.connection.query("SELECT * from users where user = ?",[username], function(err, rows){
        if (err)
            return done(err);

        if(!rows.length){
            return done(null, false, console.log('No user found.));
        }

        console.log(password);
        console.log(rows[0].password);

        if(!(rows[0].password == password)){
            return done(null, false, console.log('Ooop! Wrong password.));
        }

        return done(null, rows[0]);

    });
});

module.exports = {};

```

Las primeras líneas de código son para solicitar los módulos de `passport` y el modelo que nos permite realizar conexiones al DBMS:

```

var passport = require ('passport');
var LocalStrategy = require('passport-local').Strategy;
var mysql_config = require ('./mysql_config');

```

A continuación, se presentan dos métodos, el primero define la sesión en el browser, generando cookies e indicando que base de datos se utilizará:

```
passport.serializeUser(function(user, done){
  mysql_config.connection.query('USE parroquia');
  done(null, user.id);
});
```

El segundo se encarga de finalizar la sesión después de verificar que el id del usuario pertenezca a la base de datos y haya terminado la sesión:

```
passport.deserializeUser(function(id, done){
  mysql_config.connection.query('USE parroquia');
  mysql_config.connection.query("SELECT * from users where id = ?" ,[id], function(err, rows){
    done(err,rows[0]);
  });
});
```

Las siguientes líneas pertenecen a dos métodos, que son denominados estrategias según la documentación de `passport.js`. La primera `local-register` se encarga de agrega un nuevo usuario para que pueda utilizar el sistema y la segunda `local-login` permite que un usuario anteriormente registrado pueda iniciar sesión.

6.3.2.3 Definición del controlador de estrategias locales

Después de haber sido definidas las estrategias locales de autenticación ahora se necesita definir un controlador para implementarlas, para eso será necesario dar de alta una route, es decir, una ruta.

Al gozar de las bondades de `express`, cada vez que esa ruta sea utilizada en el browser se pueden manejar las solicitudes HTTP como GET, POST, PUT y DELETE.

A medida que se presentan en el código se explica para que sirve cada método.

Se agregan las siguientes líneas al archivo `app.js`:

```
var auth = require('./routes/auth');
app.use('/auth/', auth);
```

Se genera los archivos `auth.js` (el controlador), `login.jade` (vista de inicio de sesión), `register.jade` (vista de registro de nuevo usuario), más adelante al crear la aplicación CRUD se describe la estructura y funcionamiento de las funciones o middlewares contenidos en ellos.

auth.js

```
var passport = require('passport');
var local_register_login = require('../models/local_register_login');
var express = require('express');
var router = express.Router();

router.route('/register')
  .get(function(req, res, next){
    res.render('register',[]);
  })
  .post(passport.authenticate('local-register', {successRedirect: '/principal', failureRedirect: '/auth/register'}));

router.get('/login', function(req, res, next){
  res.render('login', {user: req.user});
});

router.post('/login',
  passport.authenticate('local-login'), function(req, res, next){
```

```

        res.redirect('/principal');
    });

    router.all('/logout',function(req,res, next){
        req.logout();
        res.redirect('/');
    });

    module.exports = router;

```

El método `route` permite indicar que se van a manejar todas las posibles solicitudes realizadas desde esa ruta, en este caso `/register`. Ahora se colocan solamente dos métodos que responderán a las solicitudes `get` y `post`.

En el primer caso se emplea el método `render` desde el objeto `res` (o respuesta) para indicar que se va a llamar a un archivo vista, `register.jade`, cuando desde esa vista se publiquen los datos con el método `post` se utiliza la función `passport.authenticate` para validar los datos recibidos, registrar un nuevo usuario e ingresar al sistema.

Después viene dos métodos más pero ahora desde la ruta `/login`, si se quiere iniciar sesión se llama a la vista `login.jade`, se reciben datos al publicar con el método `post`, se valida con la función antes mencionada y con la estrategia antes definida y se ingresa en el sistema.

El último método permite terminar la sesión y salir del sistema.

Cabe mencionar que al haber definido la ruta `/auth`, estas rutas que son definidas aquí en realidad son relativas, las rutas absolutas serían `/auth/login`, `/auth/register` y `/auth/logout`.

A continuación, las vistas antes mencionadas y utilizadas:

login.jade

```

extends layout

block content
    .navbar.navbar-inverse.navbar-fixed-top
        .container
            .navbar-header
                button.navbar-toggle(type='button', data-toggle='collapse', data-target='.navbar-collapse')
                span.sr-only Toggle Navigation
                span.icon-bar
                span.icon-bar
                span.icon-bar
                a.navbar-brand(href='/') Lupitas

    h1 Iniciar sesión
    form(role="form", action="/auth/login", method="post")
        .form-group
            label(for="username") Nombre de usuario
            input#username.form-control(name="username", type="text")
        .form-group
            label(for="password") Contraseña
            input#password.form-control(name="password", type="password")
    p
        a.btn.btn-link(href="/") Cancel
        button.btn.btn-default(type="submit", value="Log In") Log in

```

register.jade

extends layout

block content

```
.navbar.navbar-inverse.navbar-fixed-top
  .container
    .navbar-header
      button.navbar-toggle(type='button', data-toggle='collapse', data-target='.navbar-collapse')
      span.sr-only Toggle Navigation
      span.icon-bar
      span.icon-bar
      span.icon-bar
      a.navbar-brand(href='/') Lupitas

h1 Registrarse
form(role="form", action="/auth/register", method="post")
  .form-group
    label(for="username") Nombre de usuario
    input#username.form-control(name="username", type="text")
  .form-group
    label(for="password") Contraseña
    input#password.form-control(name="password", type="password")
p
  a.btn.btn-link(href="/") Cancel
  button.btn.btn-default(type="submit", value="Register") Register
```

El código anterior es de los archivos considerados vistas, debido a que muestran etiquetas para solicitar información y mediante un formulario se reciben los datos proporcionados por el usuario, esos datos son procesados con el controlador `auth.js` antes descrito.

Al implementar los archivos que contienen el código anterior el usuario puede ir al `home` o `index` (`localhost:3000/`), de ahí utilizar el botón que dirige a la ruta `/auth` y los botones que hacen solicitudes a las rutas antes definidas, después de autenticarse el usuario podrá utilizar el sistema.

6.3.3 Desarrollo de unidad base del sistema (aplicación “CRUD”)

6.3.3.1 Planeación de la aplicación

Se hacen una serie de bosquejos de la aplicación CRUD intención:

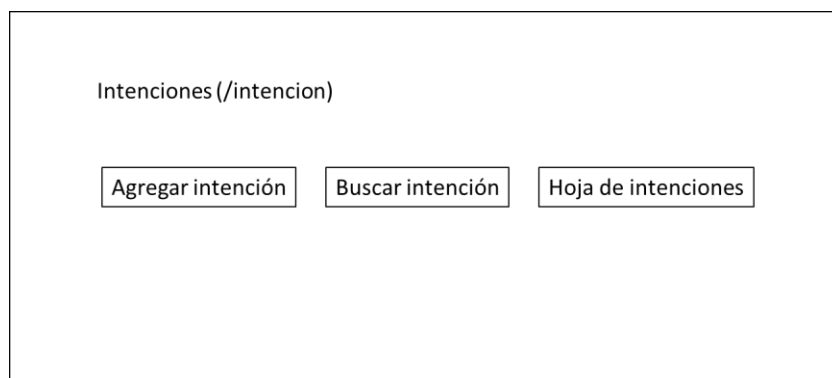


Figura 6.1 Vista principal de la aplicación CRUD Intención

Intención (/add)

Nombre:

Tipo:

.

.

Fecha de Eucaristía:

Figura 6.2 Agregar intención

Búsqueda (/intencion/buscar_intencion)

Figura 6.3 Buscar intención

Intenciones

Fecha: _____

Registro(s) encontrado(s):

	<input type="button" value="Borrar"/>	<input type="button" value="Actualizar"/>
	<input type="button" value="Borrar"/>	<input type="button" value="Actualizar"/>

Figura 6.4 Lista de intenciones

6.3.3.2 Programación del controlador que atenderá solicitudes http en la ruta de la aplicación CRUD Intención

Express.js utiliza el middleware `connect` que maneja solicitudes `http`, auxiliándose de otros middlewares que al ser parte del catálogo de `node` han sido programados en JavaScript.

`routes` es el middleware que hace posible procesar `paths` o `urls` y la información en ellas ingresadas en el browser en conjunción con `connect` y `express` utilizando JavaScript.

Las líneas que permiten utilizar `express` en la aplicación son las siguientes:

```
var express = require('express');
app.use(express.static(path.join(__dirname, 'public')));
```

Esta y otras más son agregadas por haber utilizado el template (en una sección anterior).

En el archivo `app.js` se agregan las `routes` (rutas) que desean ser atendidas por el servidor web (integrado en `node`, definido por el desarrollador o algún módulo específico), indistintamente se les llamará `paths` o `urls` también.

```
var intencion = require('./routes/intencion');
app.use('/intencion', intencion);
```

Ahora se crea el archivo `intencion.js` que controlará las vistas y utilizará los modelos (archivos `.js` de `express`, `connect` y otros que serán definidos). A continuación, el código de este archivo será presentado en subsecciones, las cuales tienen como finalidad resaltar la función de los bloques de código contenidos por cada una de ellas.

6.3.3.2.1 Recepción de solicitudes `get`

Las primeras líneas del archivo `intencion.js` son para solicitar los módulos y modelos necesarios:

```
var express = require('express');
var router = express.Router();
var mysql_config = require('../models/mysql_config');
var moment = require('moment');
```

Las siguientes líneas de código contienen un método para la ruta `/`, con él ante una solicitud `get`, es decir, llegar a esa ruta en el browser; se utilizará el método `render` desde el objeto `res`, entonces se invoca la primera vista `intencion.jade`.

```
router.get('/', function(req, res, next){
    res.render('intencion');
});
```

Es necesario señalar que al utilizar `/` (slash) se hace referencia a la raíz (root) de la ruta.

Una forma de ver `/` es como si fuera un path relativo, ya que habiendo definido la ruta `/intencion` (el path absoluto) en el archivo de la aplicación `app.js`, siempre que un método sea definido desde el archivo con el mismo nombre (o el asignado a tal ruta), el servidor web manejará `/` como si fuera `/intencion`.

Se resalta esto porque en cada aplicación CRUD futura, se comenzará por un método con / que procesará una solicitud `http`, por lo general una solicitud `get` indicando que en el browser se ha accedido a dicha ruta, aunque podría ser otro tipo de solicitud `http`.

Con respecto a la llamada a la vista, cabe señalar que dentro del código no es necesario colocar la extensión `.jade` porque el método ya sabe que va a recibir un archivo vista con esa extensión, se hace esta aclaración porque aunque en el texto se coloca la extensión dentro del código no estará presente.

6.3.3.2 Atendiendo solicitudes `get` y `post` para agregar una intención

Utilizando nuevamente el módulo y objeto `router` se define un nuevo método `route`, el cual permitirá responder a cualquier tipo de solicitud `http` emitida desde la ruta que se coloca como primer parámetro, a diferencia del código anterior donde `router.get` sólo atiende solicitudes `get`.

```
router.route('/agregar_intencion')
  .get(function(req, res){
    res.render('agregar_intencion');
  })
  .post(function(req, res){
    var intencion = {};

    mysql_config.connection.query('USE parroquia');
    mysql_config.connection.query('INSERT INTO servicio SET tipo=?,1, function(err,servicio){
      if(err) res.status(400).send('Error agregando tipo de servicio: ' + err);

      intencion.enunciado = req.body.enunciado;
      intencion.tintencion = req.body.tintencion;
      intencion.fecha = req.body.fecha;
      intencion.fcelebracion = req.body.fcelebracion;

      mysql_config.connection.query('USE parroquia');
      mysql_config.connection.query('INSERT INTO intencion SET ? , servicio_folio =?',
[intencion,servicio.insertId], function(err,_intencion){
        if(err){
          res.status(400).send('Error agregando nueva intencion: ' + err);
        } else {
          res.send("Nueva intencion agregada.");
        }
      })
    })
  })
})
```

En el código anterior si desde esa ruta se emite una solicitud `post`, es decir, desde una vista se manda un formulario, se puede acceder a su contenido utilizando el objeto `req` (solicitud) y el método `body`, por lo cual sabiendo cuales son los nombres de los componentes del formulario en `html` se puede pasar a variables su contenido. En este caso se genera un objeto y se agrega una nueva propiedad por cada valor del formulario recibido (Objeto `intencion` y sus propiedades `enunciado`, `tintencion`, `fecha` y `fcelebración`).

Después utilizando el modelo antes definido `mysql_config.js` se utiliza el módulo `mysql` para establecer la conexión a la base de datos “parroquia” y se inserta un nuevo registro en la tabla `intencion` con los datos en el objeto `intencion`.

El código finaliza enviando un mensaje si hay un error al agregar el nuevo registro o si fue agregado correctamente.

6.3.3.2.3 Presentando la información de las intenciones antes registradas utilizando las vistas

El siguiente bloque de código presenta dos métodos, el primero `get` para recibir ese tipo de solicitud desde la ruta `/hoja_intenciones` y hacer una llamada a la vista `buscar_intencion`; el segundo método `post` procesa el valor de fecha enviado desde la ruta antes mencionada, con ese valor se conecta a la base de datos `parroquia` y se busca el registro en la tabla `intencion` cuya fecha coincida con la antes recibida.

A continuación, se guardan todos los datos del registro en un arreglo, se utiliza el método `render` para llamar al archivo `hoja_intenciones_fcel.jade`, al cual se le envía el arreglo que contiene los valores del registro en la base de datos junto con dos objetos más `moment` y `fecha`, los cuales serán utilizados en la vista para mostrar su contenido con cierto formato específico.

```
router.route('/hoja_intenciones')
  .get(function(req, res){
    res.render('buscar_intencion');
  })

  .post(function(req, res){
    var fecha = {};
    fecha = req.body.fecha;

    mysql_config.connection.query('USE parroquia');
    mysql_config.connection.query('SELECT * FROM intencion WHERE date(fcelebracion) =?', [fecha],
function(err,intenciones){
    if(err){
      res.status(400).send('Error buscando intencion: ' + err);
    } else {
      res.render('hoja_intenciones_fcel', {intenciones: intenciones, moment: moment,
fecha:fecha});
    }
  })
})
```

El siguiente bloque de código es idéntico al antes descrito salvo el nombre de la vista que en esta ocasión es `intenciones`.

```
router.route('/buscar_intencion')
  .get(function(req, res){
    res.render('buscar_intencion');
  })

  .post(function(req, res){
    var fecha = {};
    fecha = req.body.fecha;

    mysql_config.connection.query('USE parroquia');
    mysql_config.connection.query('SELECT * FROM intencion WHERE fecha =?', [fecha],
function(err,intenciones){
    if(err){
      res.status(400).send('Error buscando intencion: ' + err);
    } else {
      res.render('intenciones', {intenciones: intenciones, moment: moment, fecha_hoja:fecha});
    }
  })
})
```

El siguiente bloque de código recibe el mismo comentario del previo, el nombre de la vista ahora es `hoja_intenciones.jade`.

```
router.post('/buscar_intencion/:fecha_hoja', function(req, res, next){
    var fecha = {};
    fecha = req.params.fecha_hoja;

    mysql_config.connection.query('USE parroquia');
    mysql_config.connection.query('SELECT * FROM intencion WHERE fecha =?', [fecha],
function(err,intenciones){
    if(err){
        res.status(400).send('Error buscando intencion: ' + err);
    } else {
        res.render('hoja_intenciones', {intenciones: intenciones, moment: moment, fecha:fecha});
    }
    })
})
```

6.3.3.2.4 Actualizando información de intenciones

En el siguiente bloque de código la ruta es `/buscar_intencion/:servicio_folio`, la última parte de esta ruta o url `servicio_folio` es un número el cual se necesita, se emplea el método `param` desde el objeto `req` y se extrae ese número.

Se procede a utilizar el método `all` para definir una variable que pueda ser utilizada en todos los métodos siguientes, lo que se logra al utilizar el método `next`, `idServ` será el nombre de esa variable.

El valor que guarda la variable antes mencionada es una `primary key` (clave principal) , lo que indica que un registro específico se puede obtener con el mismo.

Cuando se recibe la solicitud `get` el método `all` es invocado, `get` recibe el valor de la `primary key` contenido en `idServ`, con este valor se realiza una consulta a la base de datos, se obtiene un registro, se almacena en un arreglo (`intenciones`) y entonces este arreglo es enviado a una vista `editi.jade`.

Desde `editi.jade` se puede ver la información del registro antes mencionado, modificar los valores y finalmente publicarlos pero usando el módulo `method-override` para indicar que es una solicitud `put`. Se realiza lo anterior para activar dicho método en el archivo `intencion.js` y no el método `post`, los datos actualizados se guardan en un objeto y más tarde se emplea un query para actualizar los datos de ese registro en la base de datos.

```
router.route('/buscar_intencion/:servicio_folio')
    .all(function(req, res, next){
        idServ= req.params.servicio_folio;

        next();
    })
    .get(function(req,res, next){
        mysql_config.connection.query("USE parroquia");
        mysql_config.connection.query("SELECT * FROM intencion WHERE servicio_folio=?", [idServ], function(err,
intencion){
            if(err) res.status(400).send('Error al buscar intencion: ' + err);
            res.render('editi', {intencion: intencion[0]});
        })
    })
```

```

    .put(function(req, res, next){
        var intencion = {};

        intencion.enunciado = req.body.enunciado;
        intencion.tintencion = req.body.tintencion;
        intencion.fecha = req.body.fecha;
        intencion.fcelebracion = req.body.fcelebracion;

        mysql_config.connection.query('USE parroquia');
        mysql_config.connection.query('UPDATE intencion SET ? WHERE servicio_folio =?', [intencion, idServ],
function(err,_intencion){
            if(err){
                res.status(400).send('Error actualizando intencion: ' + err);
            } else {
                res.send("Intencion actualizada");
            }
        })
    })

    .delete(function(req,res, next){
        mysql_config.connection.query('USE parroquia');
        mysql_config.connection.query("DELETE FROM intencion WHERE servicio_folio = ? ", [idServ],
function(err,intencion){
            if(err){
                res.status(400).send("Error eliminando intencion: " + err);
            } else {
                mysql_config.connection.query('USE parroquia');
                mysql_config.connection.query("DELETE FROM servicio WHERE folio = ? ", [idServ],
function(err,servicio){
                    if(err) res.status(400).send("Error eliminando servicio: " + err);
                    res.send('Intención eliminada.');
```

6.3.3.2.5 Descripción del código que componen las vistas

Jade permite olvidar la tarea tediosa de etiquetar y cuidar el etiquetado al utilizar `html`, basta con utilizar espacios o tabuladores, es decir, indentación o sangría.

Por ejemplo, al utilizar una etiqueta `div`, es equivalente a `<div> contenido </div>`. Para cada elemento o componente HTML hay un equivalente en Jade, de forma similar Stylus facilita el uso de CSS. No es parte del objetivo de este trabajo explicar cómo utilizar Jade o Stylus, así que, la explicación de su uso se limita a las líneas anteriores.

A continuación, se presenta el contenido de los archivos vistas (`.jade`),

intencion.jade

```

extend layout

block content
    .navbar.navbar-default
        .container
            .navbar-brand Intención
            ul.nav.navbar-nav
                li
                    a#agregar_intencion(href='/intencion/agregar_intencion') Agregar
                li
                    a#buscar_intencion(href='/intencion/buscar_intencion') Buscar
```

```

intención
li
a#hoja_intenciones(href="/intencion/hoja_intenciones") Hoja de
intenciones

```

El código anterior es equivalente al primer bosquejo (Figura 6.1), muestra el título “Intención” y tres botones que me permiten agregar, buscar y listar las intenciones.

agregar_intencion.jade

```

extend layout
block content
  .row
    .col-md-6
      .navbar.navbar-default
        .navbar-brand Agregar intención
        form(role="form", method="post", action="")
          .form-group
            label(for="enunciado") Enunciado
            input#enunciado.form-control(name="enunciado", type="text" required)
          .form-group
            label(for="tintencion") Tipo de intención
            input#tintencion.form-control(name="tintencion", type="text" required)
          .form-group
            label(for="fecha") Fecha de solicitud
            input#fecha.form-control(name="fecha", type="date" required)
          .form-group
            label(for="fcelebracion") Fecha y hora de celebración
            input#fcelebracion.form-control(name="fcelebracion" type="datetime-local"
required)
          button.btn.btn-primary(type="submit") Agregar intencion

```

El código anterior es de un formulario que recibe los datos ingresados por el usuario y genera una solicitud `post` al utilizar el botón “agregar intención”, la cual es atendida por el archivo controlador `intencion.js`.

El archivo `editi.jade` posee un código similar, las únicas diferencias son los nombres de las etiquetas.

El código siguiente junto con `buscar_intencion_fcel.jade` realizan la misma tarea recibir una fecha ingresada por el usuario, razón por la que se omite el archivo antes mencionado.

buscar_intencion.jade

```

extend layout
block content
  .navbar.navbar-default
    .container
      .navbar-brand Buscar intención
      ul.nav.navbar-nav
        li
          a
            label Fecha de solicitud
        li
          a
            form.navbar-search.pull-left(role = 'form' method = 'post'
.action="")
              .form-group
                input#fecha.form-
control(name="fecha", type="date" required)
                button.btn.btn-primary(type="submit") Buscar

```

```

intenciones.jade

extend layout

block content
  .navbar.navbar-default
    if !intenciones.length
      .navbar-brand No hay intenciones:
    else
      .navbar-brand Intenciones:

    if intenciones.length
      table.table
        thead
          tr
            th
            th
            th Enunciado
            th Tipo de intención
            th Fecha de solicitud
            th Fecha de celebración

        tbody
          each intencion in intenciones
            tr
              td
                form(role = 'form' method='post'
action="/intencion/buscar_intencion/#{intencion.servicio_folio}?_method=delete")
                button.btn.btn-default(type = 'submit') Borrar
              td
                a.btn.btn-
default(href="/intencion/buscar_intencion/#{intencion.servicio_folio}' type='submit') Actualizar
              td #{intencion.enunciado}
              td #{intencion.tintencion}
              td #{moment(intencion.fecha).locale('es').format('DD MMMM YYYY')}
              td #{moment(intencion.fcelebracion).locale('es').format('DD MMMM
YYYYY, h:mm:ss A')}}

                form(role="form", method="post", action="/intencion/buscar_intencion/#{fecha_hoja}")
                button.btn.btn-primary(type="submit") Hoja de intenciones

```

El código anterior genera una tabla y mediante un ciclo coloca los valores del arreglo de `intenciones` en ella, donde se ve el objeto `moment (...)` es para dar un formato de fecha distinto al que utiliza JavaScript por default.

Los archivos `hoja_intenciones.jade` y `hoja_intenciones_fcel.jade` son variantes de este código por eso se omite colocar su código.

De esta manera mi aplicación CRUD “intención” está finalizada.

Terminar el desarrollo del sistema implica agregar los archivos correspondientes a los directorios `models`, `routes` y `views` para genera otras aplicaciones CRUD (más de veinte) que trabajarán conjuntamente.

6.3.4 Generación de documentos

Para la generación de documentos se necesitan dos módulos `fs` y `pdfkit` lo cual se realiza con las siguientes instrucciones:

```

npm install fs --save
npm install pdfkit --save

```

Se describe a continuación de forma breve y en general el proceso.

Las vistas son generadas utilizando código en JavaScript en conjunción con el módulo `mysql`, ahora en lugar de enviar los arreglos a un archivo (`.jade`) dentro de los métodos se agregan las siguientes líneas:

```
var doc = new PDFDocument;  
doc.pipe(fs.createWriteStream('/ruta_en_sistema_de_archivos/nombre_del_documento.pdf'));  
    código en JavaScript en conjunción con el módulo pdfkit  
doc.end();
```

Con las líneas anteriores se genera un archivo pdf cuyo contenido depende del código previo a emplear el método `end`.

A petición del usuario la tarea esencial para la generación de documentos es producir una “caída”, es decir, texto que coincida con los campos en sus formatos, razón de que se emplearán los módulos antes mencionados.

Se hizo la propuesta de generar los formatos desde el sistema, pero el usuario prefirió que se les proporcionase una forma de utilizar los formatos de la parroquia (debido a la inversión realizada en ellos). Además se desarrollaron dos formas de generarlos, la manual que hace uso de la base de datos “documentos” y la automática que emplea la base de datos “parroquia”. En la primera situación se ingresan los datos que formarán el documento, se almacena en la base de datos y finalmente se genera el pdf a imprimir. En la segunda se da click en el botón con el nombre del documento a generar y se genera el archivo pdf.

Para finalizar esta sección se presenta el código (bloque de código que pertenece al archivo `documentos.js`) que genera un acta de primera comunión, junto con una explicación de cada parte del código y sus consideraciones previas.

Un acta de primera comunión tiene los siguientes campos a llenar: nombre, padrinos, presbítero y fecha de primera comunión. En cualquiera de las dos formas antes mencionadas se obtienen los datos que formarán la caída del documento.

Habiendo realizado una consulta a una base de datos se almacena el arreglo obtenido en un objeto llamado `apc`.

```
apc = db_aob[0];
```

Se define el tamaño de la hoja, su orientación y márgenes:

Considerando que una pulgada es equivalente a 72 puntos, la conversión para una hoja tamaño carta es de 612 de ancho por 792 de alto.

Asignamos la orientación horizontal o landscape.

Para indicar que no se considerarán márgenes se coloca un cero en la propiedad `margin`.

Lo antes mencionado se encuentra en las siguientes líneas:

```
var doc = new PDFDocument({
    size: [612, 792.00],
    layout: 'landscape',
    margin: 0
});
```

Se indica la ubicación del archivo y su nombre:

En Windows:

```
doc.pipe(fs.createWriteStream('C:\\Users\\Ramon\\Apreza\\Zepeda\\Desktop\\acta de primera comunión.pdf'));
```

En un sistema Unix o GNU/Linux:

```
doc.pipe(fs.createWriteStream('/home/xmissio/desktop/acta de primera comunión.pdf'));
```

Habiendo obtenido las coordenadas de los campos del formato y convertido a puntos, se asigna el tipo y tamaño de fuente.

En este ejemplo se deja la fuente por default pero se indica que el tamaño será de 12 puntos con el método `fontSize`.

Para colocar el valor de una propiedad del objeto `apc`, se emplea el método `text` con la propiedad y las coordenadas de donde se desea que aparezca en el pdf como argumentos.

El método `text` considera el primer argumento como una cadena de texto, así que, aunque se emplea el módulo y objeto `moment` para cambiar el formato de algunas propiedades, `text` lo recibe como argumento válido.

Lo antes mencionado se observa en el siguiente bloque de código:

```
doc.fontSize(12)
    .text(moment(apc.fsacramento).locale('es').format('DD'), 360, 478.08);
doc.fontSize(12)
    .text(moment(apc.fsacramento).locale('es').format('MMMM'), 460.8, 478.08);
doc.fontSize(12)
    .text(moment(apc.fsacramento).locale('es').format('YYYY'), 604.8, 478.08);
doc.fontSize(12)
    .text(apc.sacerdote, 460.8, 429.12);
doc.fontSize(12)
    .text(apc.nombre, 201.6, 282.24);
doc.fontSize(12)
    .text(apc.padrino, 201.6, 316.8);
doc.fontSize(12)
    .text(apc.madrina, 201.6, 351.36);
```

La siguiente línea indica que se ha terminado de crear el archivo pdf.

```
doc.end();
```

La siguiente imagen es el acta de primera comunión que se obtiene de imprimir el archivo `primera comunión.pdf` en su formato correspondiente:

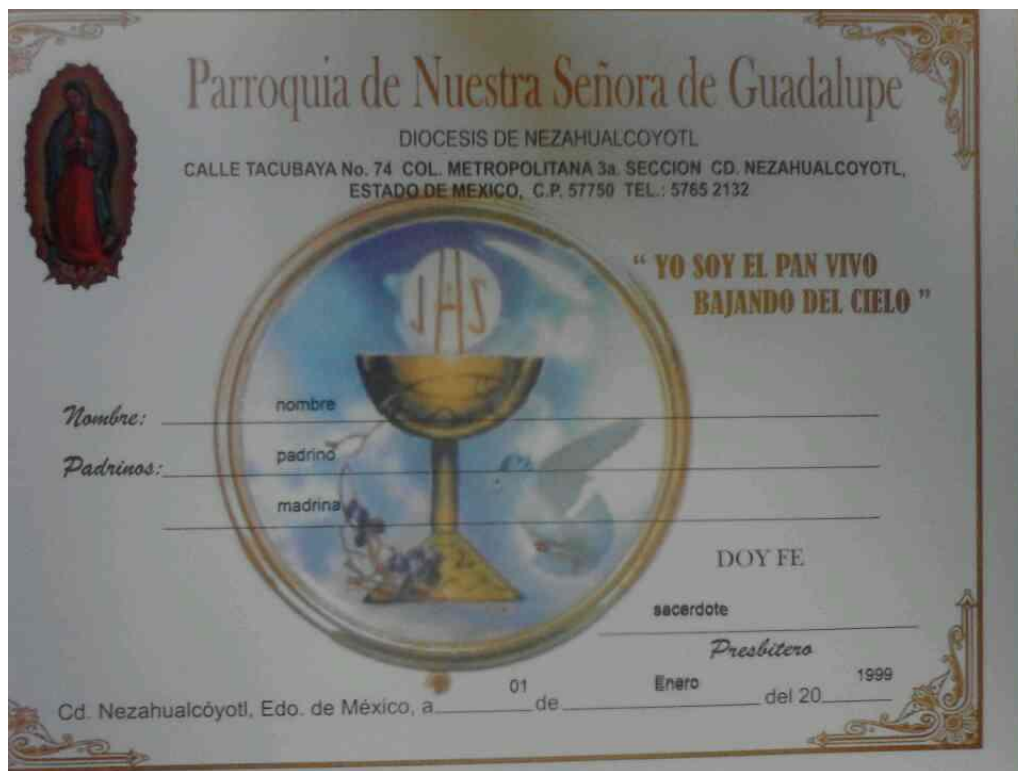


Figura 6.5 Acta de primera comunión

Cabe señalar que dependiendo la impresora puede diferir donde es insertado el texto, los archivos pdf que genera el sistema sólo consideran las coordenadas adecuadas para la impresión en la parroquia.

6.4 COMENTARIOS Y RECOMENDACIONES

Para trabajar con Node.js se puede hacer como la forma presentada a lo largo de este documento o empleando un ambiente de desarrollo integrado (IDE por sus siglas en inglés), algunos de ellos son Enide, WebStorm y VisualStudio.

Existen distintas versiones de módulos para mysql, la elegida fue aquella que permitió realizar las tareas necesarias en este proyecto.

Cuando se agregan módulos es recomendable utilizar el archivo `package.json` y agregar caret (circunflejo o tilde) antes de cada versión para asegurar compatibilidad con la versión de node, express o algún módulo empleado en el proyecto.

Es necesario poseer o desarrollar lógica de programación y experiencia con C, C++, o Java o algún otro lenguaje de programación, además documentarse con respecto a JavaScript para poder explotar el potencial de Node.js.

Stackoverflow es un sitio en el que como desarrollador puedes exponer tus preguntas y recibir la respuesta de la comunidad. Si no se desea ser miembro, al realizar una consulta en cualquier motor de búsqueda se puede agregar "stackoverflow" como argumento para recibir enlaces con respuestas del sitio, por lo general si se agrega la duda, el stack y la palabra antes mencionada se obtendrá la mayoría de los enlaces con la respuesta que se busca.

CAPÍTULO 7

IMPLEMENTACIÓN DEL SISTEMA

Este capítulo está compuesto por cuatro secciones, en la primera se muestra la implementación del de una versión de prueba que fue puesta a disposición del usuario para que el usuario le diera el visto bueno. En el proceso de pruebas se descubrieron varios errores y el usuario aporato bastantes sugerencias para agregar ciertas funciones y mejoras al sistema. La siguiente sección es un comentario breve sobre la capacitación del usuario y la última sección son notas de implementación, comentarios y recomendaciones.

7.1 IMPLEMENTACIÓN DEL SISTEMA

A continuación, presento screenshots del sistema, la autenticación y el uso de la aplicación CRUD creada en el capítulo anterior.



Figura 7.1 Login del sistema lupitas.



Figura 7.2 Registro de un usuario



Figura 7.3 Página principal del sistema



7.4 Acceso a aplicación CRUD intención

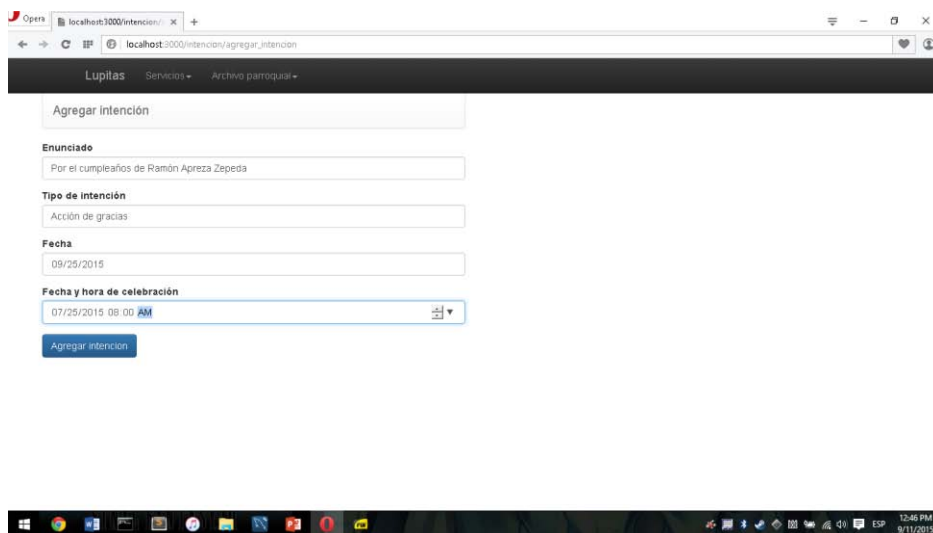


Figura 7.5 Agregar una intención



Figura 7.6 Búsqueda por fecha para obtener la hoja de intenciones



Figura 7.7 Hoja de intenciones del día 25 de Julio



Figura 7.8 Botones para eliminar o actualizar intenciones

7.2 CAPACITACIÓN

La capacitación fue de la siguiente manera delante de los usuarios se utilizó, se visitó cada sección, se agregó, actualizó y borró información.

Después se les pidió a los usuarios que utilizarán el sistema, se observó cómo ingresaban información, contestando sus preguntas y dándoles sugerencias.

Mientras utilizaban el sistema se identificaron varios errores, se presentaron por la forma en la que ingresaban la información, debido a campos que llenaban de más, otros donde no sabían que escribir.

Los usuarios sugirieron algunos cambios en la forma de ingresar información, en especial el área de criptas, uno que otro cambio en campos de los feligreses y en cuanto a los sacramentos también.

El sistema se quedó en funcionamiento mientras se procedía a realizar modificaciones para implementar posteriormente.

Progresivamente se fueron proporcionando nuevas versiones a los usuarios, recibiendo su retroalimentación, implementando correcciones y mejoras.

La versión final es el resultado del proceso antes mencionado.

7.3 COMENTARIOS Y SUGERENCIAS

Se ha utilizado HTML5 en algunas vistas por lo cual si se quieren usar las ventajas de los tipos especializados es necesario una versión actualizada del browser.

El comportamiento del usuario es impredecible, el analista no es capaz de predecir que conocimiento posee y su forma de utilizar las herramientas de software. El analista debe supervisar la interacción del usuario con el sistema de tal forma que el usuario se sienta en confianza y libertad, sin temor de ser humillado al equivocarse o por propiciar alguna falla en el sistema.

La mejor herramienta para la implementación de un sistema es la comunicación con el usuario, a lo largo de este proyecto algunas veces el analista paso bastante tiempo en compañía de los usuarios en su área de trabajo, en otras ocasiones utilizando redes sociales y otras por medios de comunicación habituales todo esto fue la clave para llegar a la versión final del sistema.

CONCLUSIONES

El sistema ha sido puesto a disposición de los usuarios, con la esperanza de que su utilización les permita llevar a cabo las tareas administrativas (mencionadas en capítulos anteriores) en menor tiempo, además, de ayudarles evitar incurrir en los errores comunes que se presentaban en su realización antes de su implementación.

Los pasos que propone Bagui (2011, traducción propia) con respecto al proceso de ingeniería de software y su enfoque de diseño de bases de datos, permiten desarrollar un sistema con gran libertad y flexibilidad, además de dar la pauta a seguir a lo largo del desarrollo del mismo.

El analista observo como en la etapa de desarrollo las bases de datos generadas en MySQL Workbench podían ser modificadas a voluntad sin afectar los esquemas en el estilo de Chen.

Cuando se iba de una etapa a otra en el proceso de ingeniería de software el revisar las fases del proceso de ingeniería de software según Bagui (2011, traducción propia) orientaba al analista y le permitía avanzar tanto en el proceso mismo como en la realización de este documento.

Se puede concluir que este aporte de Bagui (2011, traducción propia) es una herramienta fundamental para personas que comienzan a adentrarse en el diseño y desarrollo de bases de datos y sistemas de información.

El utilizar JavaScript como lenguaje de programación en el backend fue una experiencia nueva y realmente grata debido a la presencia de Node.js. El analista antes había utilizado JavaScript en el middle tier y en el frontend, pero realmente desconocía su uso en el backend, además, del amplio catálogo de módulos que proporcionan una riqueza en cuanto herramientas de desarrollo. Se puede reiterar el aprecio por este lenguaje junto con una amplia recomendación para su empleo a todo lector de este trabajo.

Node.js no es sólo una herramienta invaluable de software, que proporciona un ambiente de ejecución y desarrollo para JavaScript y libera de una dependencia a un web browser, sino que es un gran proyecto que provee a sus miembros y todo desarrollador del apoyo y soporte para innovar y crear nuevo software capaz de satisfacer las necesidades de usuarios más exigentes utilizando la tecnología más reciente. Ejemplos de esto son el administrador de paquetes (npm) que permite acceder fácil y rápidamente al catálogo de módulos siempre en aumento y la documentación, tutoriales y material de referencia que los miembros de la comunidad en torno a node publica y que fueron una herramienta irremplazable para el analista.

El analista antes experimentó la preocupación de la implementación de aplicaciones, como el servidor web limitaba respecto al stack o la plataforma de desarrollo. Por lo cual algo más para alabar en node es su servidor web diseñado para proporcionar un alto rendimiento en el proceso de atender solicitudes, su capacidad de ser definido y personalizado y las alternativas para implementación remota. Express y node liberaron al analista de toda preocupación con respecto a la implementación del sistema de forma local.

La independencia de plataforma que Node.js ofrece permiten trabajar de forma continua y a alta velocidad, en caso de trabajar en equipo la interacción sería excepcional. El analista experimento imprevistos con su estación de trabajo por lo cual tuvo que migrar de una Mac con OSX a un pc con Windows y más tarde a otra con GNU/Linux, el cambio de un equipo a otro fue cuestión de minutos y el trabajo de desarrollo no fue interrumpido.

Es motivo de mención como Jade, Stylus y Bootstrap permiten desarrollar una interfaz gráfica elegante, llamativa e intuitiva en cuestión de minutos, liberando de la preocupaciones del etiquetado de HTML y CSS, así como, la necesidad de implementar software que podría reducir el desempeño de la aplicación u otros frameworks que hacen crecer de forma inmensurable el número de dependencias y limitan a un ambiente de desarrollo específico para obtener los mismos resultados, algo que ha experimentado el analista.

Para finalizar este trabajo se puede agregar que siempre es necesaria una guía adecuada y una gran apertura por parte de quien lleva a cabo cualquier proyecto, a pesar de que se considere innecesaria y se pueda menospreciar que se ofrezca o exija para alguna tarea. Sin la gracia de Dios, sin la metodología de ingeniería de software, sin la orientación que el analista recibió del trabajo de Bagui (2011, traducción propia), sin material de apoyo como el elaborado por Nuzum (2014, traducción propia), por mencionar a otro de los autores que más apoyo proporcionaron al analista a lo largo de este trabajo, careciendo de la experiencia de su asesora, siendo nulo el apoyo de su familia, la ausencia de dedicación, disciplina y esfuerzo no se podría haber concluido el sistema de información y este documento, el cual se espera sea de apoyo para cualquiera que lo tome como material de referencia.

LISTA DE REFERENCIA

Bagui S. *Database Design Using Entity-Relationship Diagrams* [Inglés Diseño de bases de datos usando diagramas Entidad-Relación], Second Edition, Auerbach Publications, 10/2011.

Nuzum, M., *Learning Express Web Application Development* [Inglés Aprendiendo Express para desarrollo de aplicaciones web], Packt publishing, 12/2014.

Gupta, A., *Java EE 7 Essentials* [Inglés Lo esencia de Java EE 7], O'Reilly Media Inc., 08/2013

Jones, D., *Javascript: Novice to Ninja* [Inglés De novato a ninja con Javascript], SitePoint, 11/2014

Syed, B.A., *Beginning Node.js* [Inglés Comenzando con Node.js], Apress, 12/2014

npm, 2015, *Frequently Asked Questions* [Inglés Preguntas frecuentes], Fecha de última modificación Julio 03. <https://docs.npmjs.com/>.

Oracle, 2015, *MySQL Workbench*, Fecha de acceso Octubre 17. <https://www.mysql.com/products/workbench/>.

Sublime HQ Pty Ltd, 2015, *Home*, Fecha de acceso Octubre 17. <http://www.sublimetext.com/>.

Bootstrap, 2015, *Home*, Fecha de acceso Octubre 17. <http://getbootstrap.com/>.

The University of Chicago, 2015, *Chicago-Style Citation Quick Guide* [Inglés Guía de referencia rápida estilo Chicago]. Fecha de acceso Octubre 19. http://www.chicagomanualofstyle.org/tools_citationguide.html.

Librería Editrice Vaticana, 2015, *SEGUNDA PARTE LA CELEBRACIÓN DEL MISTERIO CRISTIANO*, fecha de acceso Octubre 22. http://www.vatican.va/archive/catechism_sp/p2s1c2a1_sp.html .

QuinStreet, 2015, *Code*, fecha de acceso Octubre 22. <http://www.webopedia.com/TERM/C/code.html> .

QuinStreet, 2015, *CSS*, fecha de acceso Octubre 22. <http://www.webopedia.com/TERM/C/CSS.html> .

Librería Editrice Vaticana, 2015, Capítulo III *Del proceso para las dispensas del matrimonio rato y no consumado*, fecha de acceso Octubre 22. http://www.vatican.va/archive/ESL0020/_P6N.HTM.

TechTarget, 2015, *framework* [marco, estructura, sistema], fecha de acceso octubre 22. <http://whatis.techtarget.com/definition/framework>.

QuinStreet, 2015, *function* [función], fecha de acceso Octubre 22. <http://www.webopedia.com/TERM/F/function.html>.

QuinStreet, 2015, *function* [función], fecha de acceso Octubre 22.
<http://www.webopedia.com/TERM/F/function.html>.

QuinStreet, 2015, *HTML5*, fecha de acceso Octubre 22.
<http://www.webopedia.com/TERM/H/HTML5.html> .

Oracle, 2015, *Inheritance* [herencia], fecha de acceso Octubre 22.
<https://docs.oracle.com/javase/tutorial/java/land/subclasses.html>.

QuinStreet, 2015, *method* [método], fecha de acceso Octubre 22.
<http://www.webopedia.com/TERM/M/method.html> .

QuinStreet, 2015, *MongoDB*, fecha de acceso Octubre 22.
<http://www.webopedia.com/TERM/M/mongodb.html>.

QuinStreet, 2015, *MySQL*, fecha de acceso Octubre 22.
<http://www.webopedia.com/TERM/M/MySQL.html> .

Librería Editrice Vaticana, 2015, *CAPÍTULO VI DE LAS PARROQUIAS, DE LOS PARROCOS Y DE LOS VICARIOS PARROQUIALES*, fecha de acceso Octubre 22.
http://www.vatican.va/archive/ESL0020/_P1T.HTM .

QuinStreet, 2015, *PHP*, fecha de acceso Octubre 22.
<http://www.webopedia.com/TERM/P/PHP.html> .

QuinStreet, 2015, *object-oriented programming* [programación orientada a objetos], fecha de acceso Octubre 22.
http://www.webopedia.com/TERM/O/object_oriented_programming_OOP.html.

QuinStreet, 2015, *query* [consulta], fecha de acceso Octubre 22.
<http://www.webopedia.com/TERM/Q/query.html>.

Librería Editrice Vaticana, 2015, *PARTE I DE LOS SACRAMENTOS*, fecha de acceso Octubre 22. http://www.vatican.va/archive/ESL0020/_P2R.HTM .

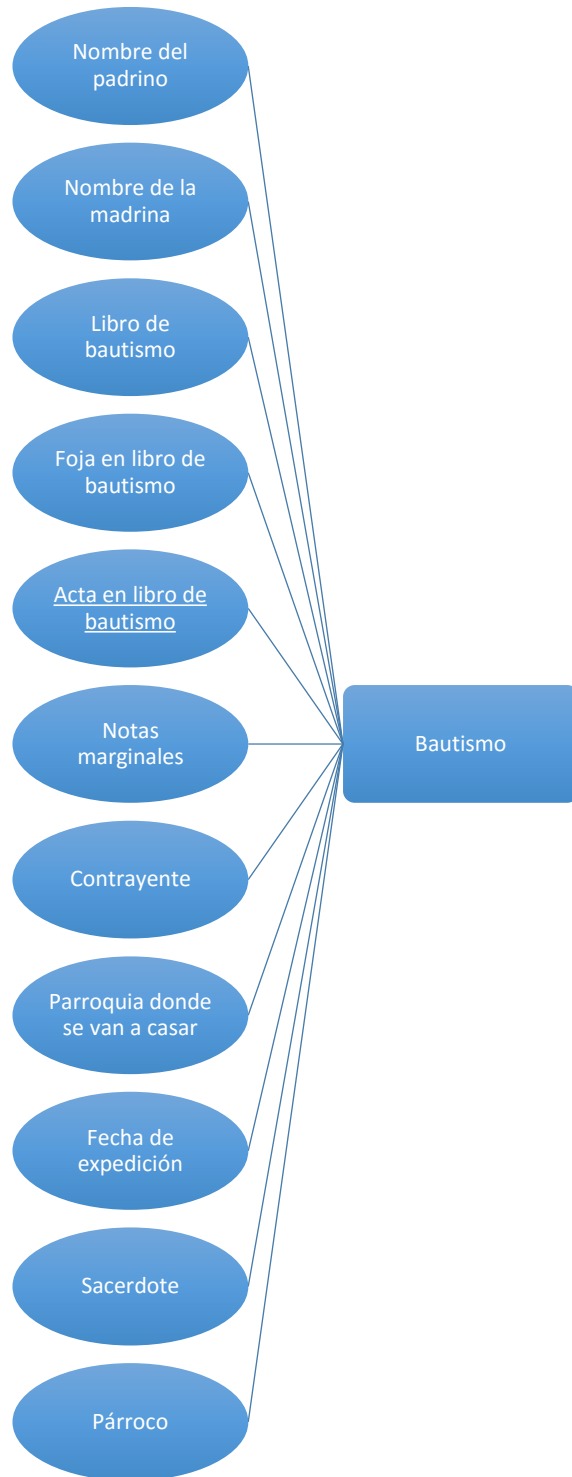
QuinStreet, 2015, *system* [sistema], fecha de acceso Octubre 22.
<http://www.webopedia.com/TERM/S/system.html> .

QuinStreet, 2015, *Web stack*, fecha de acceso Octubre 22.
http://www.webopedia.com/TERM/W/Web_stack.html.

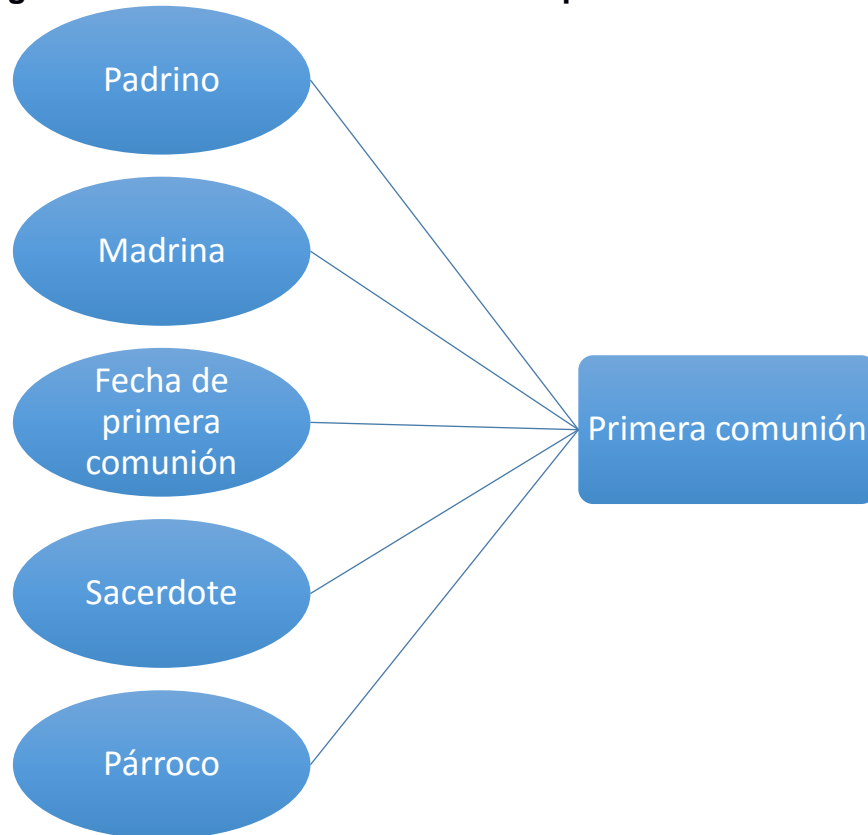
QuinStreet, 2015, *Template*, fecha de acceso Octubre 22.
<http://www.webopedia.com/TERM/T/template.html>.

Nota: El material de referencia fue mayor el presentado en esta lista de referencia, pero al elegir el estilo Harvard para referencias y la descripción del mismo acorde al “The Chicago Manual Style” (The University of Chicago, 2015), décimo sexta edición; señala que si elijo el sistema autor fecha se crear una lista de referencia sólo del material citado, si hubiera elegido el estilo de notas y bibliografía hubiera agregado una bibliografía, en la cual se incluye el material citado y el material de referencia consultado (traducción propia).

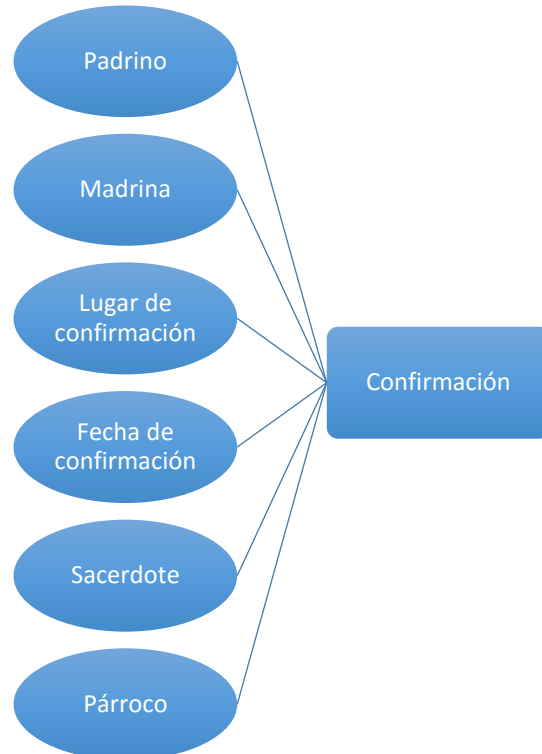
A.2 Diagrama E-R estilo Chen de la entidad bautismo



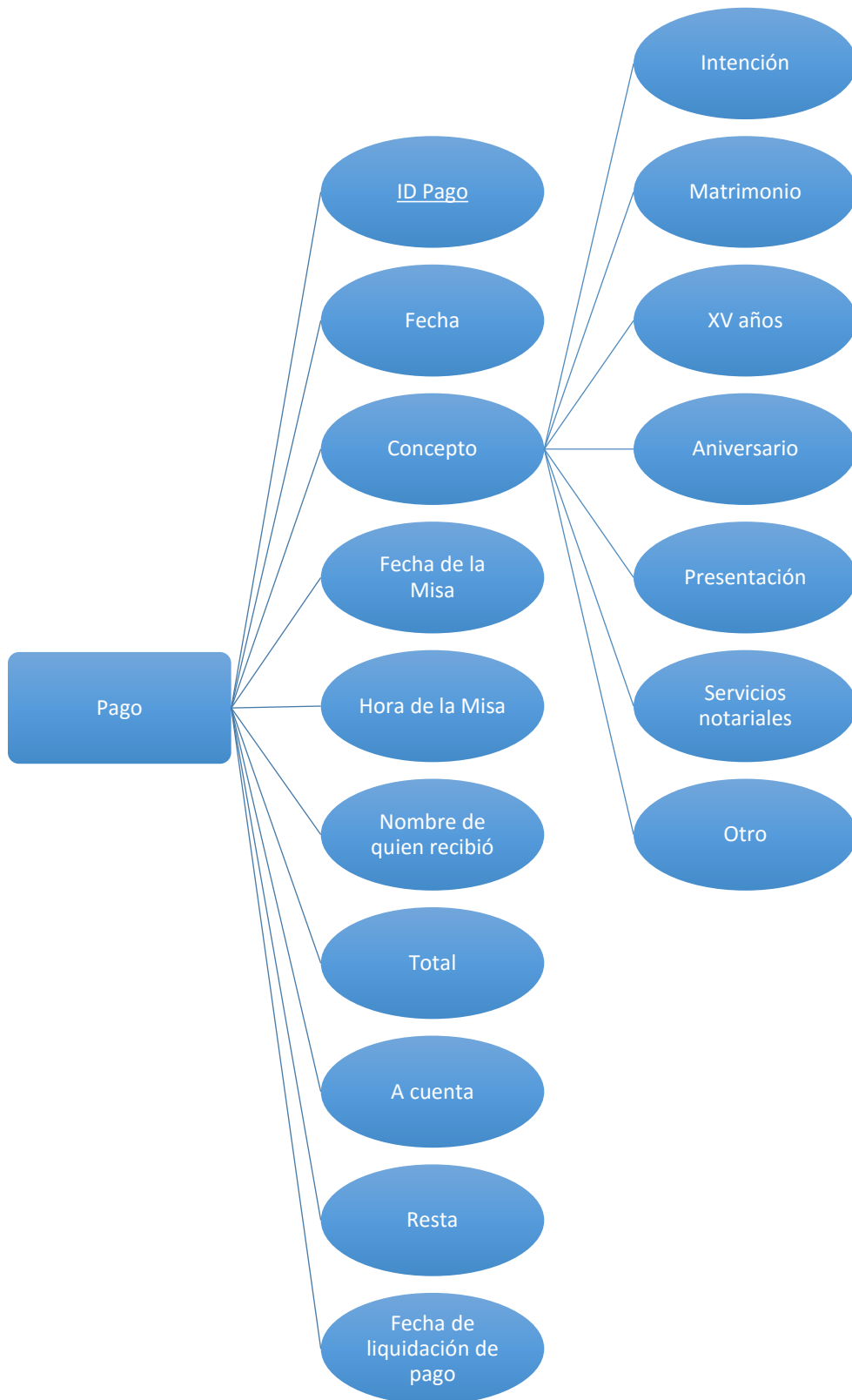
A.3 Diagrama E-R estilo Chen de la entidad primera comunión



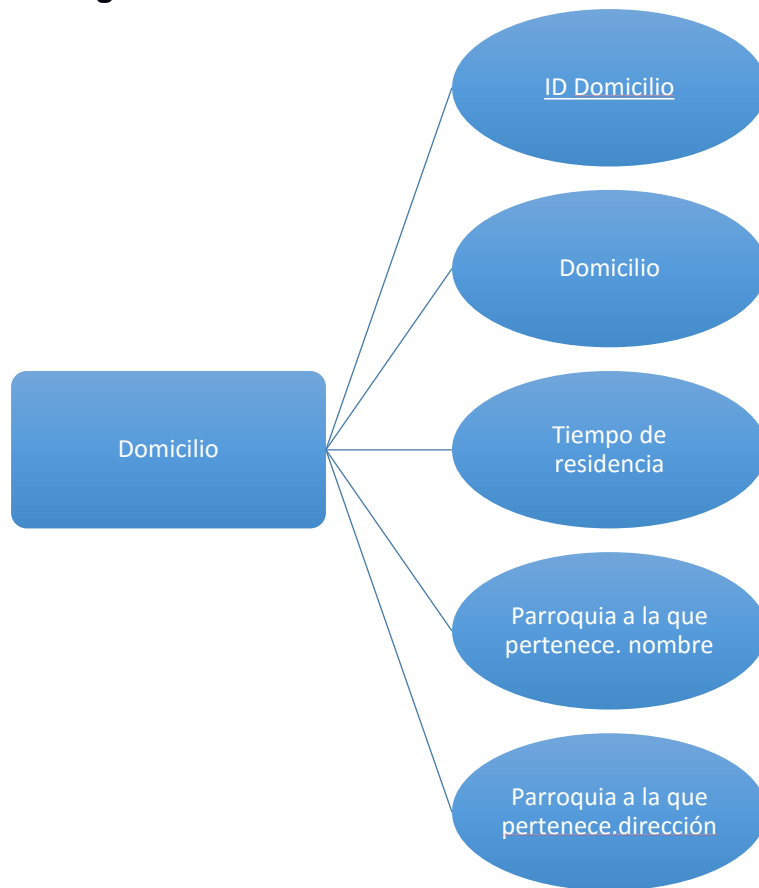
A.4 Diagrama E-R estilo Chen de la entidad confirmación



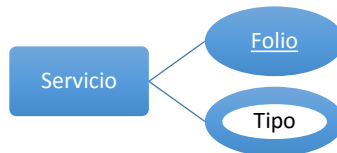
A.5 Diagrama E-R estilo Chen de la entidad pago



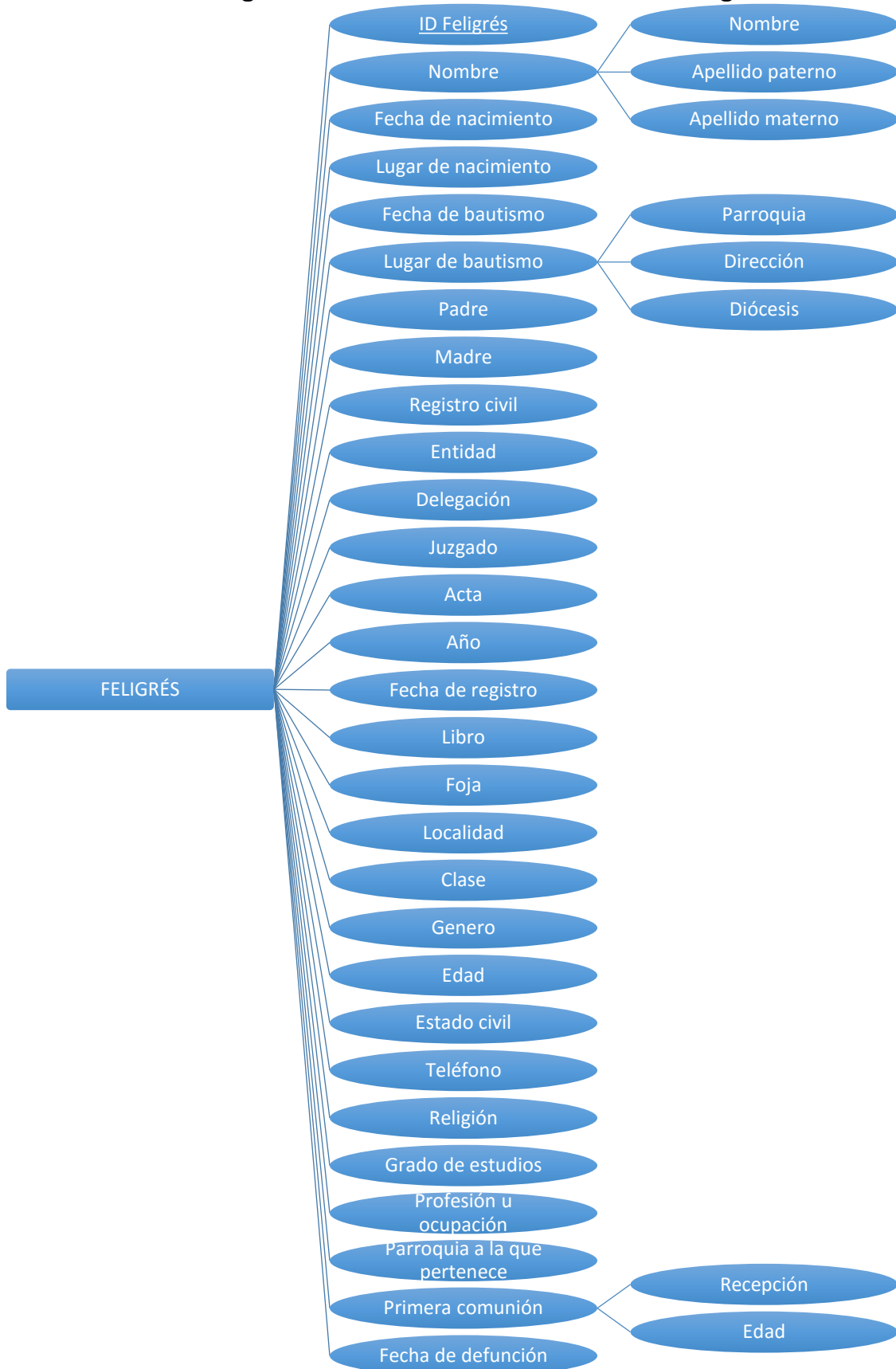
A.6 Diagrama E-R estilo Chen de la entidad domicilio



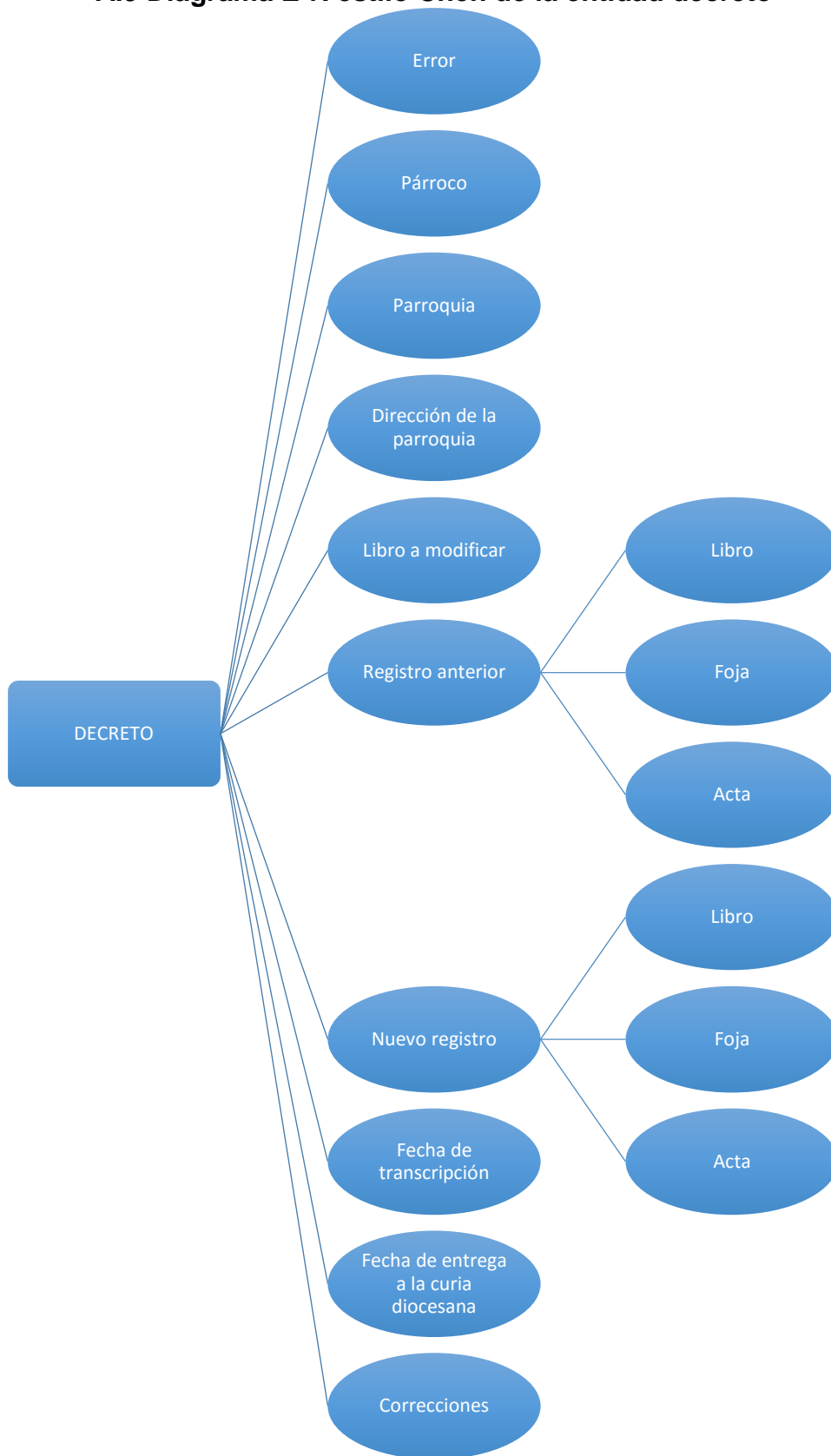
A.7 Diagrama E-R estilo Chen de la entidad servicio



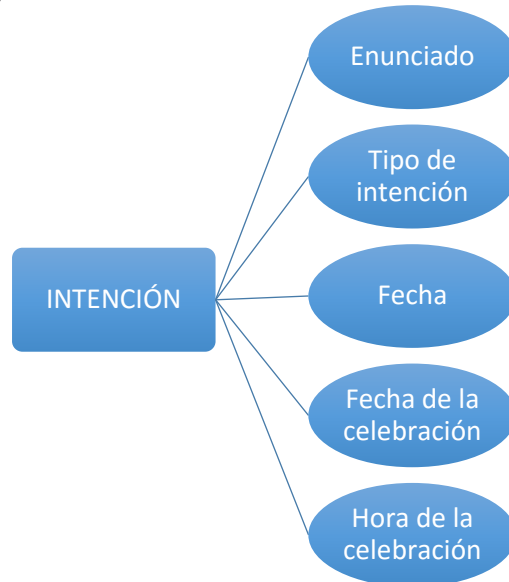
A.8 Diagrama E-R estilo Chen de la entidad feligrés



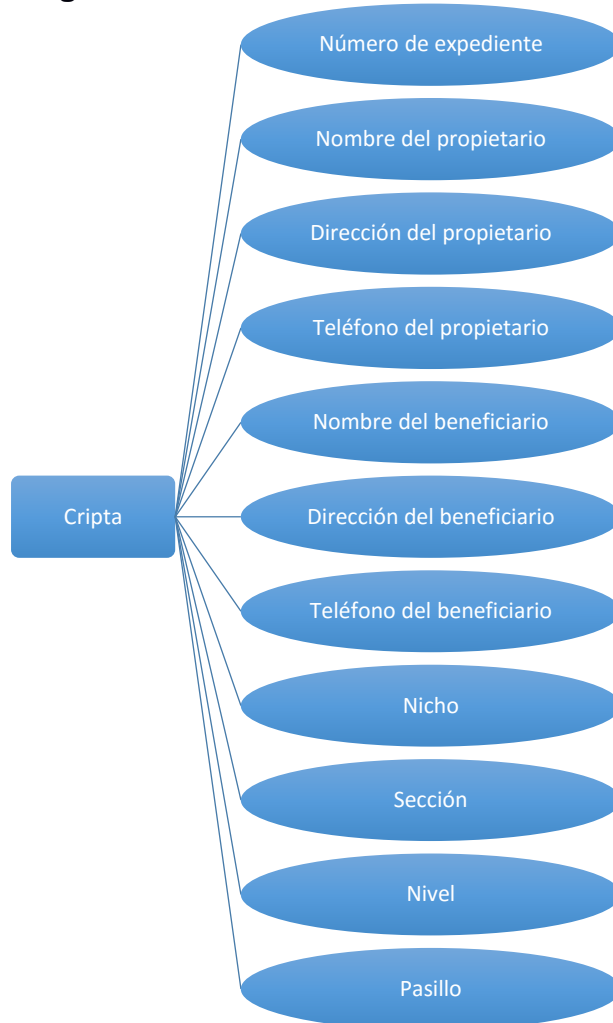
A.9 Diagrama E-R estilo Chen de la entidad decreto



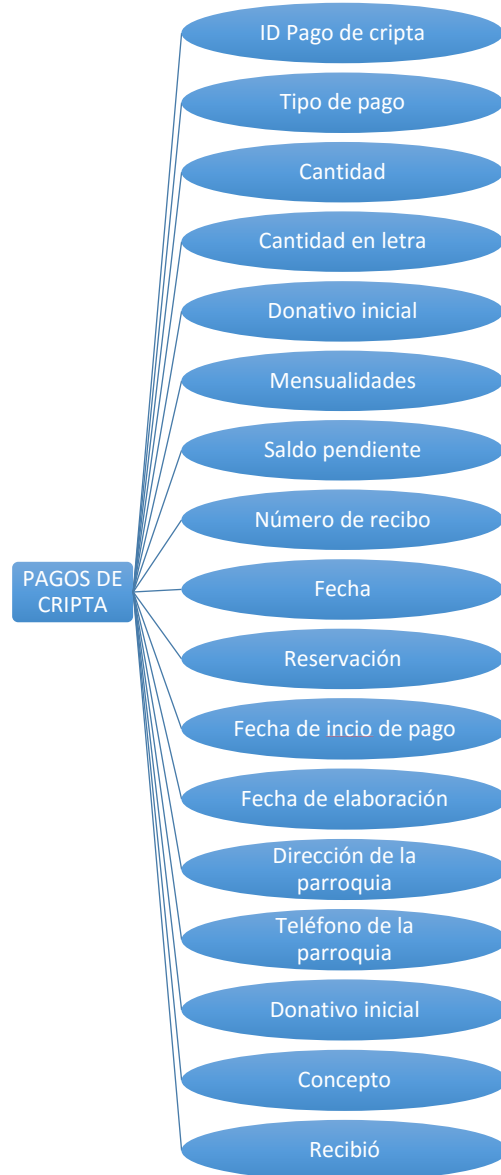
A.10 Diagrama E-R estilo Chen de la entidad intención



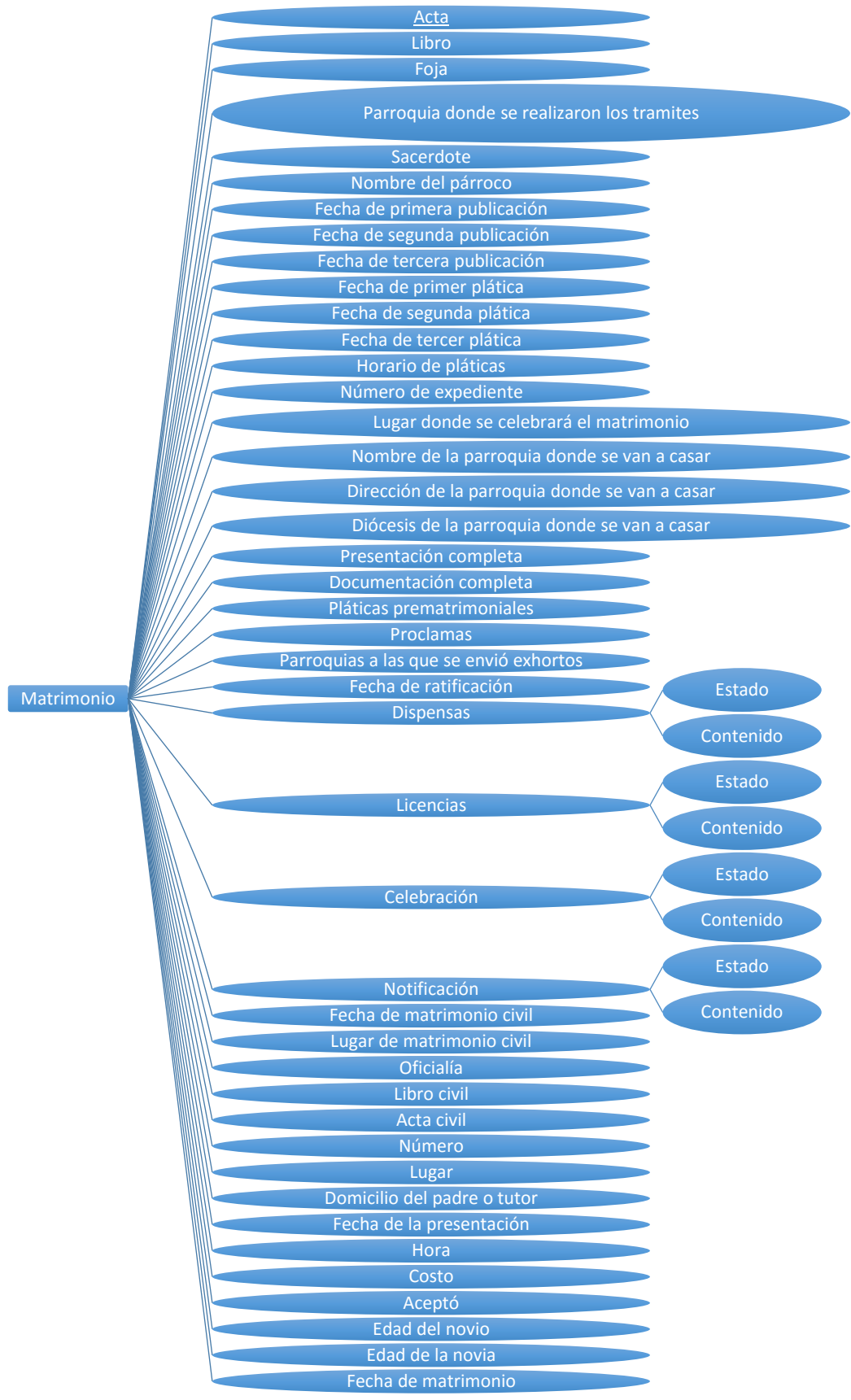
A.11 Diagrama E-R estilo Chen de la entidad cripta



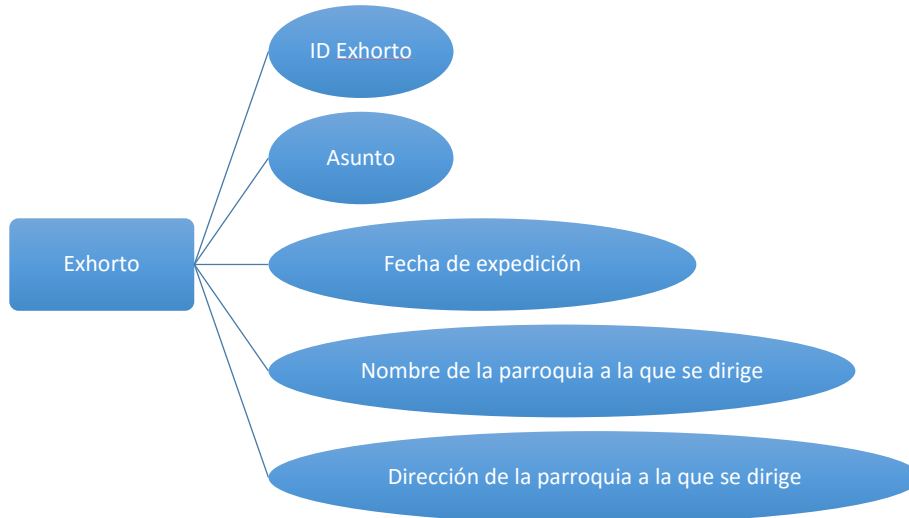
A.12 Diagrama E-R estilo Chen de la entidad pagos de cripta



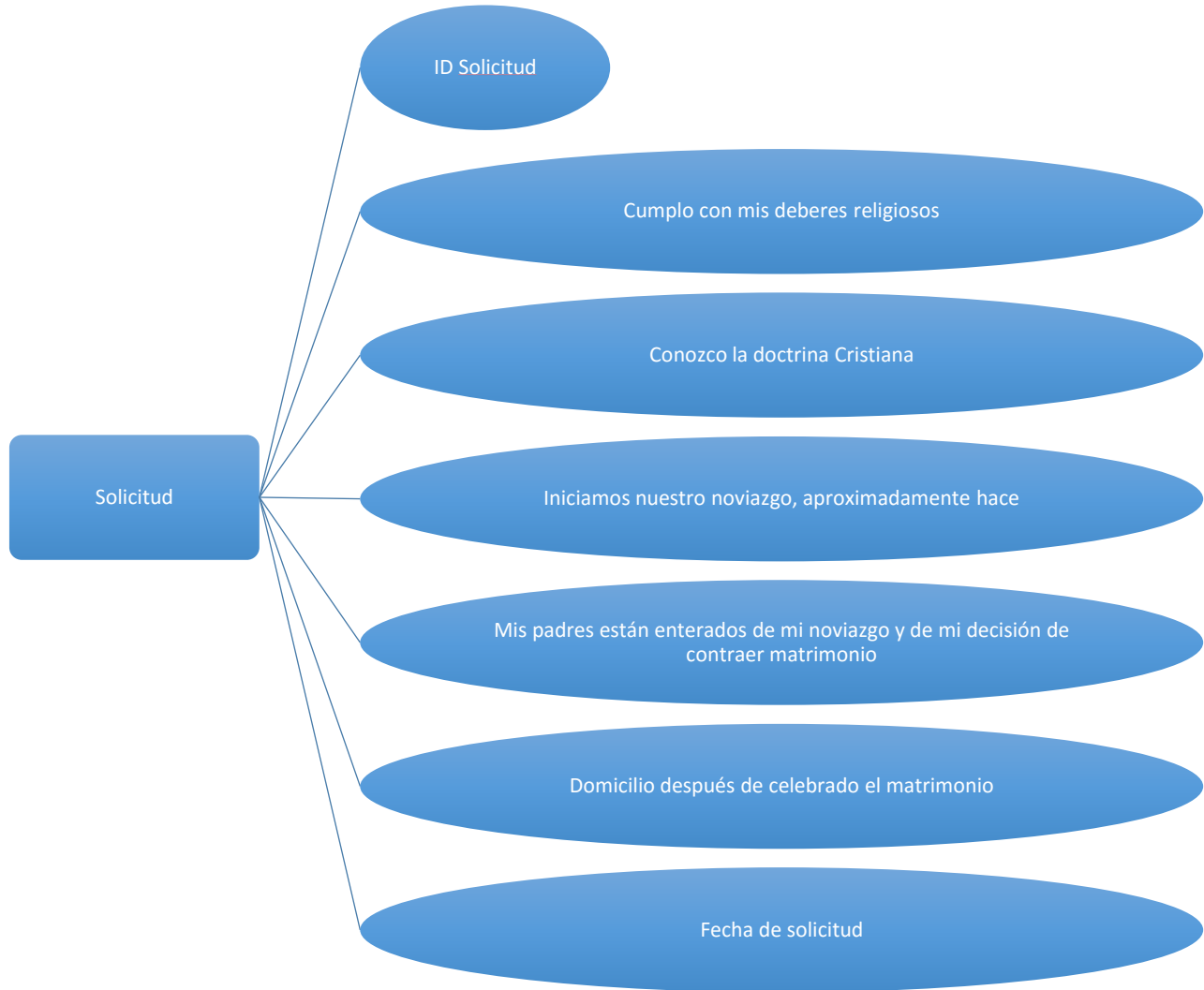
A.13 Diagrama E-R estilo Chen de la entidad matrimonio



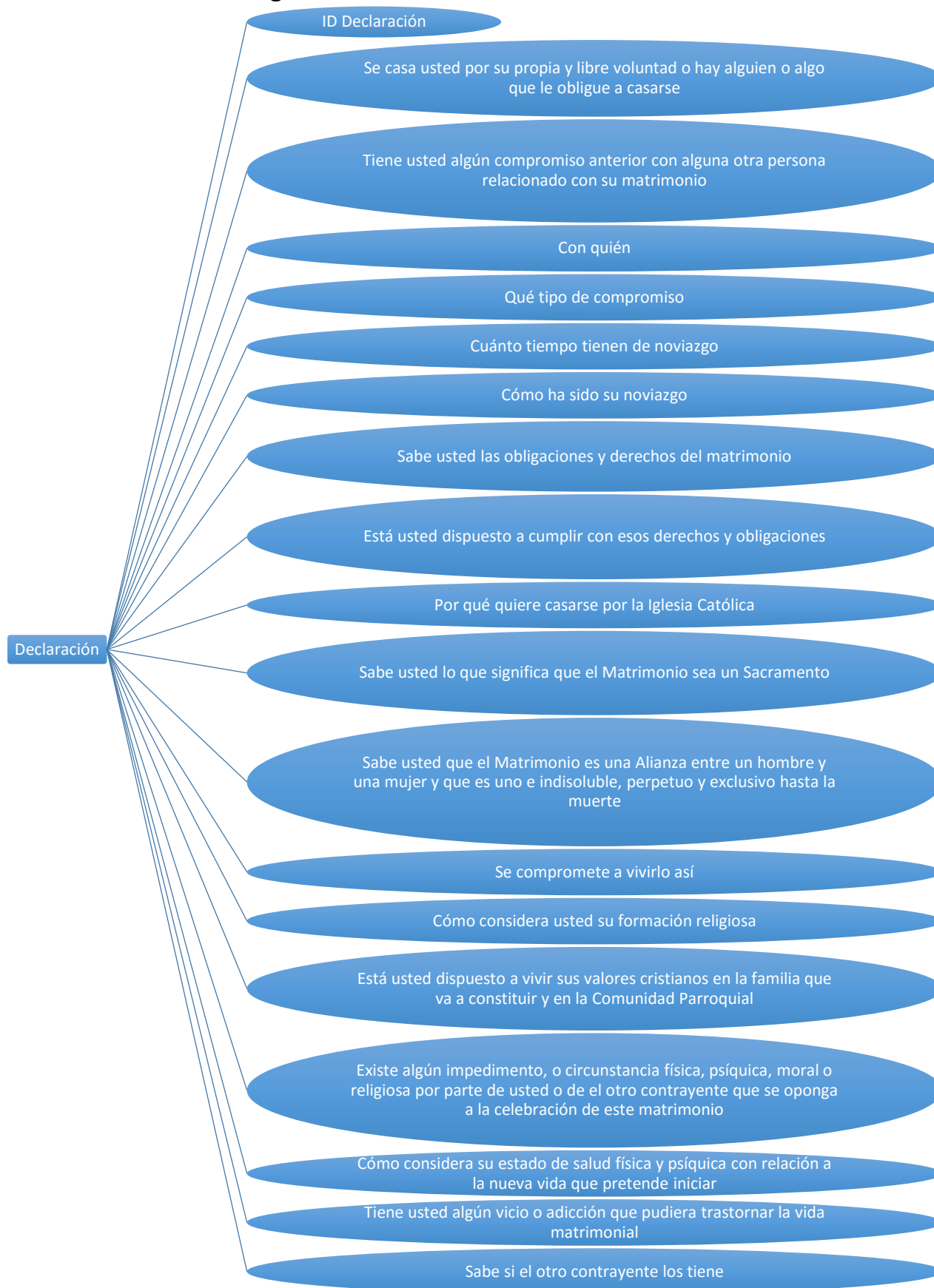
A.14 Diagrama E-R estilo Chen de la entidad exhorto



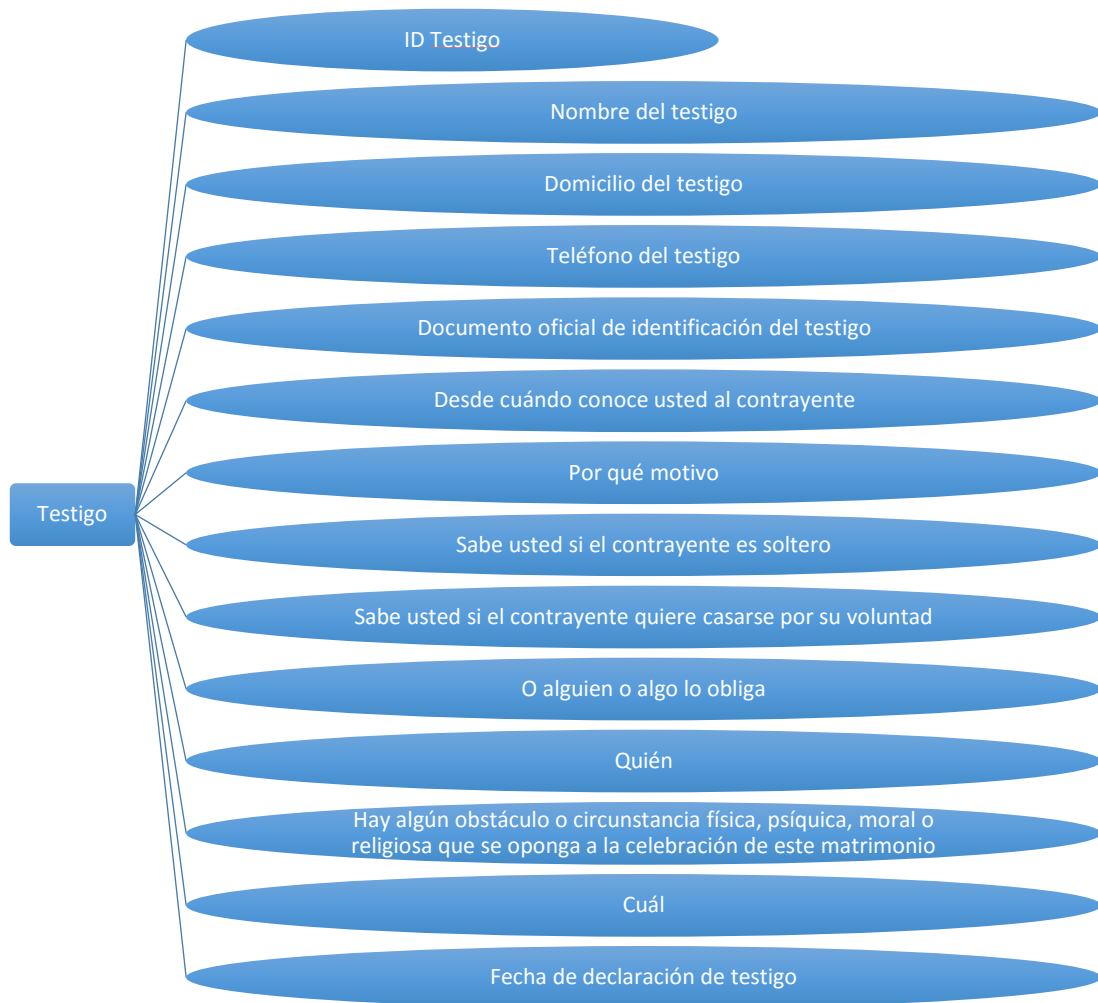
A.15 Diagrama E-R estilo Chen de la entidad solicitud



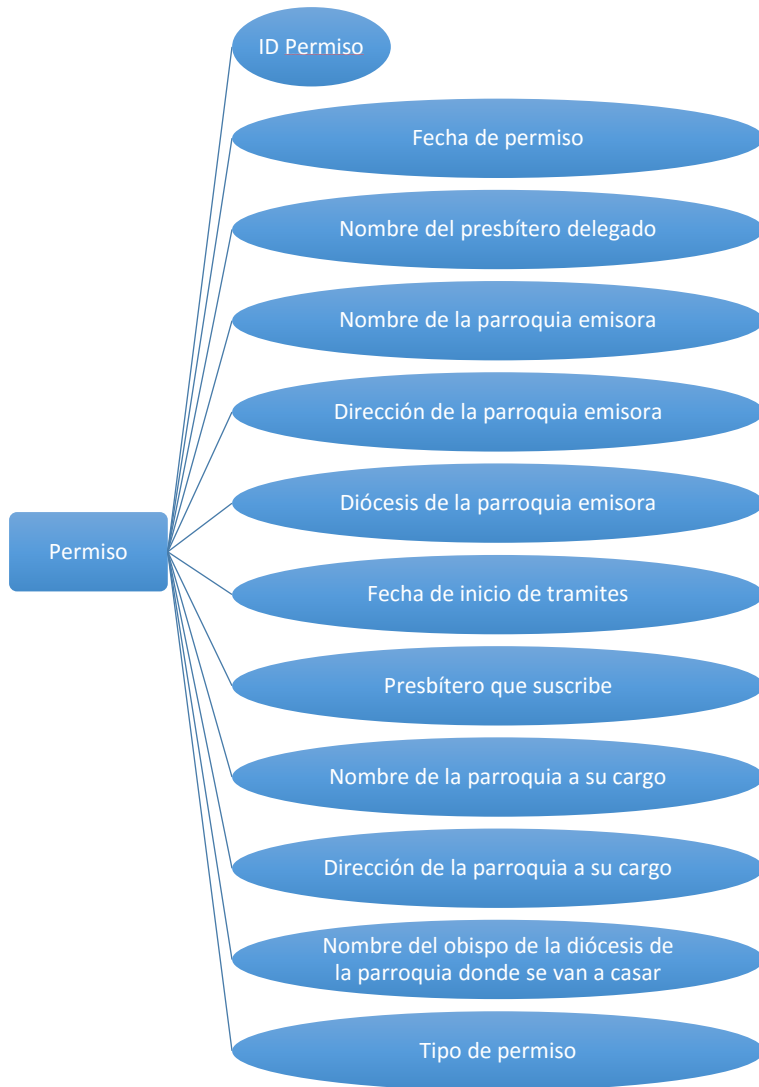
A.16 Diagrama E-R estilo Chen de la entidad declaración



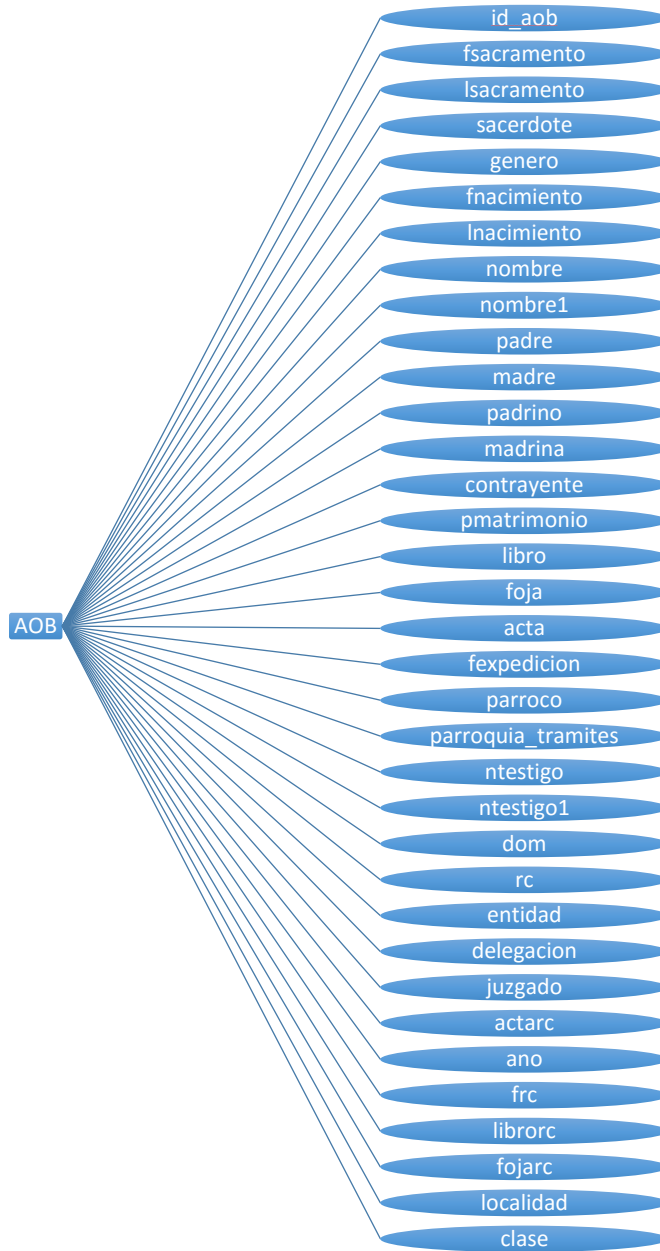
A.17 Diagrama E-R estilo Chen de la entidad testigo



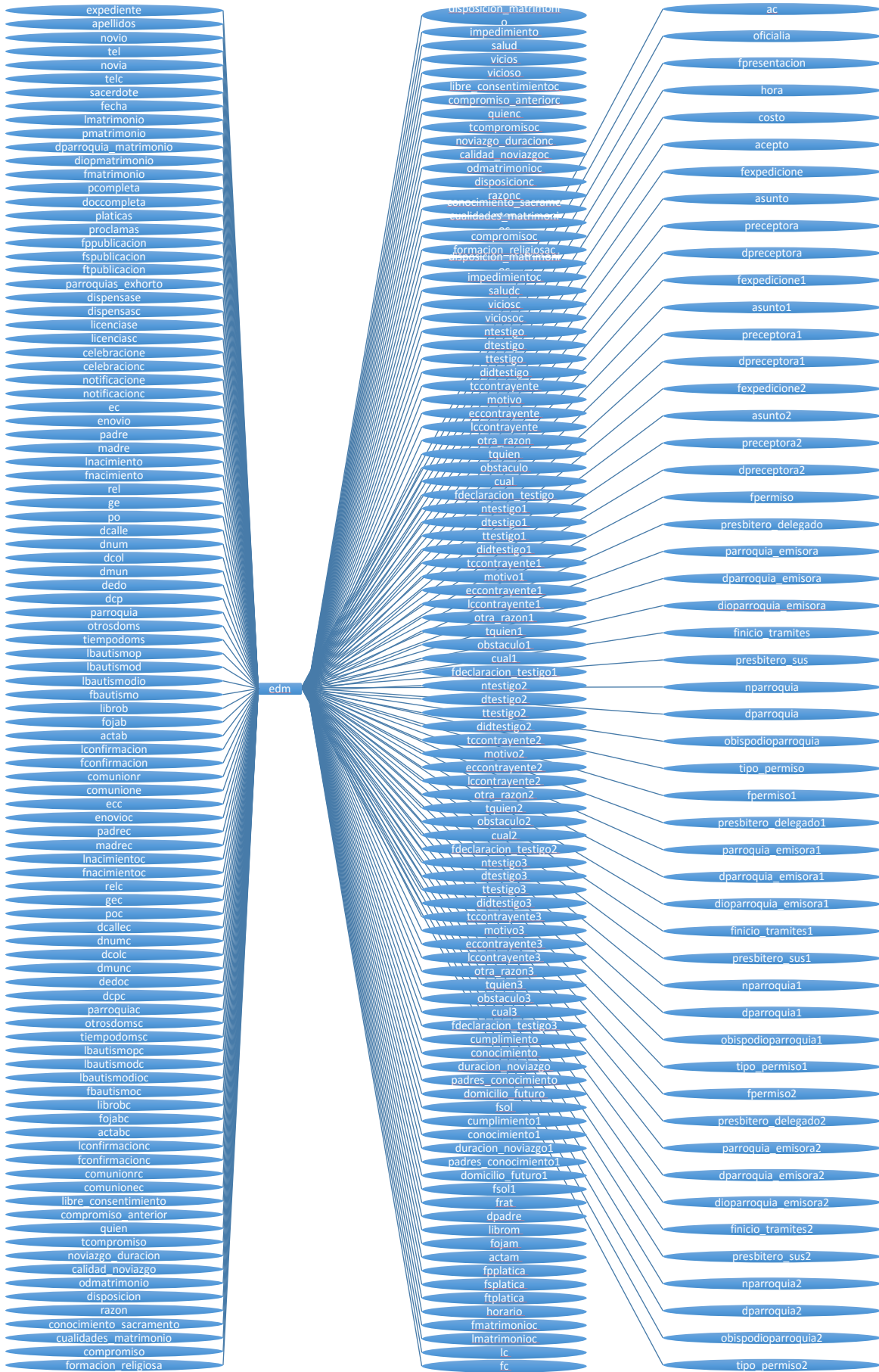
A.18 Diagrama E-R estilo Chen de la entidad permiso



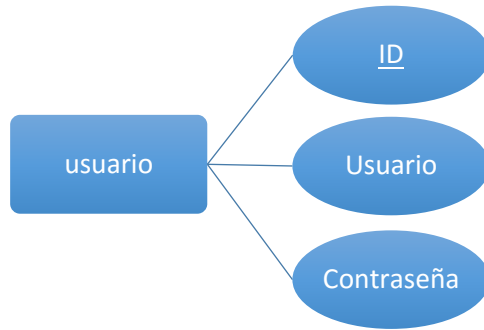
A.21 Diagrama E-R estilo Chen de la entidad aob



A.22 Diagrama E-R estilo Chen de la entidad edm

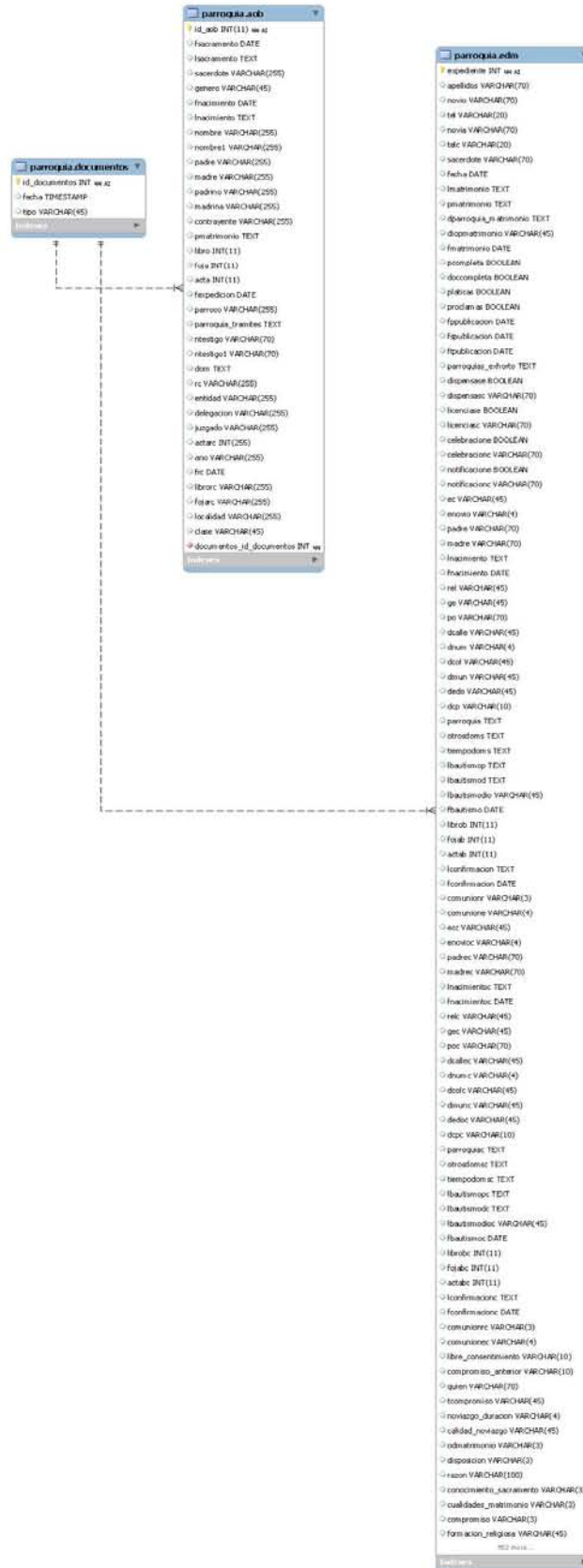


A.22 Diagrama E-R estilo Chen de la base de datos usuarios



ANEXO B: DIAGRAMAS GENERADOS EN MYSQL WORKBENCH DEL CAPÍTULO 5

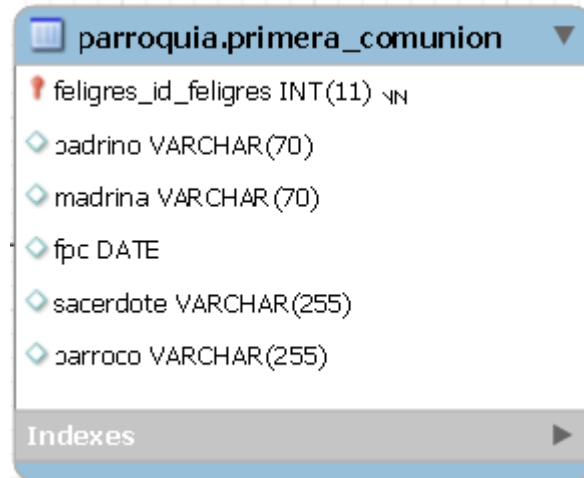
B.1 Diagrama E-R de la base de datos parroquia



B.2 Diagrama E-R estilo de la entidad bautismo



B.3 Diagrama E-R de la entidad primera comunión



B.4 Diagrama E-R de la entidad confirmación

parroquia.confirmacion	
feligres_id_feligres	INT(11) \checkmark NN
padrino	VARCHAR(70)
madrina	VARCHAR(70)
lconfirmacion	TEXT
fconfirmacion	DATE
sacerdote	VARCHAR(255)
parroco	VARCHAR(255)

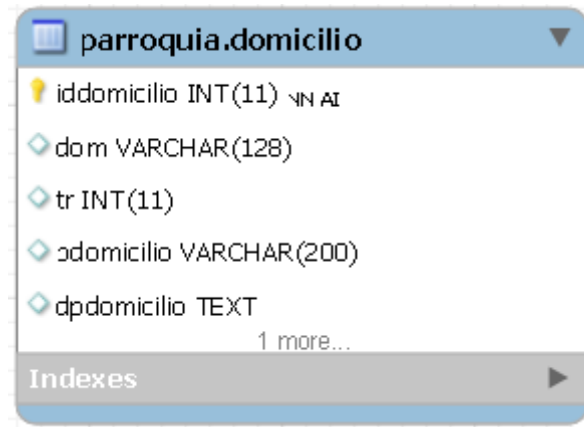
Indexes

B.5 Diagrama E-R de la entidad pago

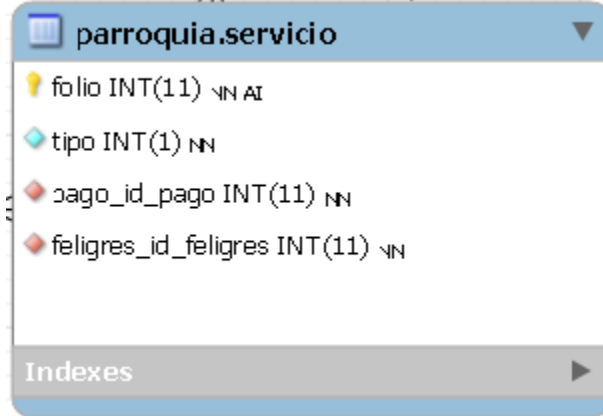
parroquia.pago	
id_pago	INT(11) \checkmark NN AI
fecha	DATE
cintencion	VARCHAR(255)
cmatrimonio	VARCHAR(255)
cxv	VARCHAR(255)
caniversario	VARCHAR(255)
cpresentacion	VARCHAR(255)
carchivo	VARCHAR(255)
cotro	TEXT
fmisa	DATE
gmisa	TIME
recibio	VARCHAR(70)
total	VARCHAR(45)
ac	VARCHAR(45)
resta	VARCHAR(45)
fliquidacion	DATE
feligres_id_feligres	INT(11) \checkmark NN

Indexes

B.6 Diagrama E-R de la entidad domicilio



B.7 Diagrama E-R de la entidad servicio



B.8 Diagrama E-R de la entidad feligres

parroquia.feligres	
id_feligres	INT(11) PK AI
nombre	VARCHAR(255)
apellido	VARCHAR(255)
apellido2	VARCHAR(255)
nacimiento	DATE
bautismo	TEXT
bautismo2	TEXT
bautismo3	TEXT
bautismo4	VARCHAR(255)
padre	VARCHAR(255)
madre	VARCHAR(255)
ic	VARCHAR(255)
entidad	VARCHAR(255)
delegacion	VARCHAR(255)
jurisdic	VARCHAR(255)
actvic	INT(255)
ano	VARCHAR(255)
fic	DATE
fidvic	VARCHAR(255)
fajvic	VARCHAR(255)
localidad	VARCHAR(255)
clase	VARCHAR(45)
genero	VARCHAR(255)
ec	VARCHAR(255)
tel	VARCHAR(20)
rel	VARCHAR(255)
ge	VARCHAR(255)
pa	VARCHAR(255)
parroquia	TEXT
comunion1	VARCHAR(255)
comunion2	VARCHAR(255)
rfdef	DATE
id_conyugate	INT(11) FK

Indexes

B.9 Diagrama E-R de la entidad decreto

decreto	
servicio_folio	INT(11) NN
idDecreto	INT(11) NN AI
error	TEXT
parroco	VARCHAR(70)
dparroquia	VARCHAR(200)
lmodificar	INT(11)
ralibro	INT(11)
rafoja	INT(11)
raacta	INT(11)
nrlibro	INT(11)
nrfoja	INT(11)
nracta	INT(11)
ftranscripcion	DATE
fentrega	DATE
correcciones	TEXT
parroquia	VARCHAR(100)

Indexes

B.10 Diagrama E-R de la entidad intención

parroquia intencion	
servicio_folio	INT(11) NN
idIntencion	INT(11) NN AI
enunciado	TEXT
tintencion	VARCHAR(45)
fecha	DATE
fcelebracion	DATETIME(0)

Indexes

B.11 Diagrama E-R de la entidad cripta

parroquia.cripta	
servicio_folio	INT(11) NN
expediente	INT(11) NN
propietario	VARCHAR(70)
dpropietario	VARCHAR(128)
tpropietario	VARCHAR(20)
beneficiario	VARCHAR(70)
dbeneficiario	VARCHAR(128)
tbeneficiario	VARCHAR(20)
nicho	INT(5)
seccion	VARCHAR(45)
nivel	VARCHAR(1)
pasillo	VARCHAR(45)
Indexes	

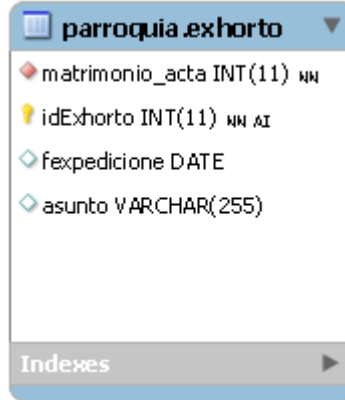
B.12 Diagrama E-R de la entidad pagos de cripta

parroquia.pagos_de_criptas	
cripta_servicio_folio	INT(11) NN
idpagoc	INT(11) NN AI
tpago	VARCHAR(255)
cantidad	DECIMAL(7,2)
cantidadl	TEXT
donativoi	DECIMAL(7,2)
mensualidades	INT(2)
spendiente	DECIMAL(7,2)
numero_recibo	INT(11)
fecha	DATE
reservacion	VARCHAR(255)
finicio_pago	DATE
felaboracion	DATE
dparroquia	TEXT
tparroquia	VARCHAR(20)
concepto	VARCHAR(255)
recibo	VARCHAR(70)
Indexes	

B.13 Diagrama E-R de la entidad matrimonio

Field Name	Data Type	Constraints
acta	INT(11)	PK
libro	INT(11)	
foja	INT(11)	
parroquia_tramites	TEXT	
sacerdote	VAR-CHAR(255)	
parroco	VAR-CHAR(255)	
fpublicacion	DATE	
fspublicacion	DATE	
ftpublicacion	DATE	
fppublica	DATE	
fsplatica	DATE	
ftplatica	DATE	
horario	VAR-CHAR(45)	
expediente	VAR-CHAR(45)	
Imatrimonio	TEXT	
pmatrimonio	TEXT	
dparroquia_matrimonio	TEXT	
dioparroquia_matrimonio	VAR-CHAR(255)	
pcompleta	BOOLEAN	
doccompleta	BOOLEAN	
platicas	BOOLEAN	
proclamas	BOOLEAN	
parroquias_exhorto	TEXT	
frat	DATE	
dispensase	BOOLEAN	
dispensasc	VAR-CHAR(255)	
licenciase	BOOLEAN	
licenciasc	VAR-CHAR(255)	
celebracione	BOOLEAN	
celebracionc	VAR-CHAR(255)	
notificacione	BOOLEAN	
notificacionc	VAR-CHAR(255)	
fmatrimonioc	DATE	
Imatrimonioc	TEXT	
oficialia	VAR-CHAR(45)	
lc	INT(11)	
fc	INT(11)	
ac	INT(11)	
dpadre	TEXT	
fpresentacion	DATE	
hora	TIME	
costo	VAR-CHAR(45)	
acepto	VAR-CHAR(70)	
enovia	VAR-CHAR(45)	
enovia	VAR-CHAR(45)	
feligres_id_feligres	INT(11)	FK
fmatrimonio	DATE	

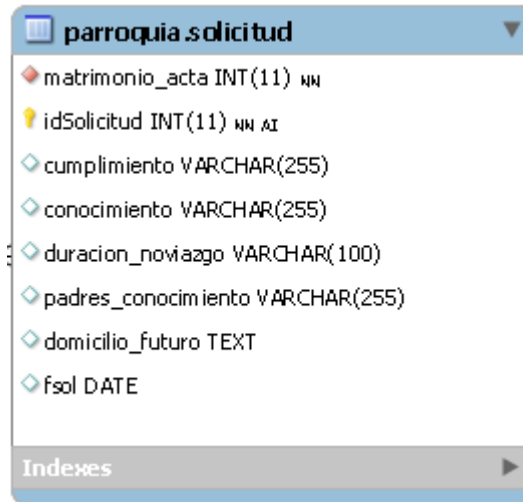
B.14 Diagrama E-R de la entidad exhorto



The screenshot shows a table definition for 'parroquia.exhorto'. It lists four columns: 'matrimonio_acta' (INT(11) NN), 'idExhorto' (INT(11) NN AI), 'fexpedicion' (DATE), and 'asunto' (VARCHAR(255)). There is an 'Indexes' section at the bottom with a right-pointing arrow.

Column Name	Data Type	Constraints
matrimonio_acta	INT(11)	NN
idExhorto	INT(11)	NN AI
fexpedicion	DATE	
asunto	VARCHAR(255)	

B.15 Diagrama E-R de la entidad solicitud



The screenshot shows a table definition for 'parroquia.solicitud'. It lists eight columns: 'matrimonio_acta' (INT(11) NN), 'idSolicitud' (INT(11) NN AI), 'cumplimiento' (VARCHAR(255)), 'conocimiento' (VARCHAR(255)), 'duracion_noviazgo' (VARCHAR(100)), 'padres_conocimiento' (VARCHAR(255)), 'domicilio_futuro' (TEXT), and 'fsol' (DATE). There is an 'Indexes' section at the bottom with a right-pointing arrow.

Column Name	Data Type	Constraints
matrimonio_acta	INT(11)	NN
idSolicitud	INT(11)	NN AI
cumplimiento	VARCHAR(255)	
conocimiento	VARCHAR(255)	
duracion_noviazgo	VARCHAR(100)	
padres_conocimiento	VARCHAR(255)	
domicilio_futuro	TEXT	
fsol	DATE	

B.16 Diagrama E-R de la entidad declaración

Field Name	Data Type	Constraints
matrimonio_acta	INT(11)	PK, NN
idDeclaracion	INT(11)	PK, NN, AI
libre_consentimiento	VARCHAR(255)	
compromiso_anterior	VARCHAR(255)	
quien	VARCHAR(70)	
tcompromiso	VARCHAR(255)	
noviazgo_duracion	VARCHAR(255)	
calidad_noviazgo	VARCHAR(255)	
odmatrimonio	VARCHAR(255)	
disposicion	VARCHAR(255)	
razon	VARCHAR(255)	
conocimiento_sacramento	VARCHAR(255)	
cualidades_matrimonio	VARCHAR(255)	
compromiso	VARCHAR(255)	
formacion_religiosa	VARCHAR(255)	
disposicion_matrimonio	VARCHAR(255)	
impedimento	VARCHAR(255)	
salud	VARCHAR(255)	
vicios	VARCHAR(255)	
vicioso	VARCHAR(255)	

Indexes

B.17 Diagrama E-R de la entidad testigo

Field Name	Data Type	Constraints
matrimonio_acta	INT(11)	PK, NN
idTestigo	INT(11)	PK, NN, AI
ntestigo	VARCHAR(70)	
dtestigo	TEXT	
ttestigo	VARCHAR(20)	
didtestigo	VARCHAR(255)	
tccontrayente	VARCHAR(45)	
motivo	TEXT	
eccontrayente	VARCHAR(255)	
lccontrayente	VARCHAR(255)	
otra_razon	TEXT	
tquien	VARCHAR(70)	
obstaculo	VARCHAR(100)	
cual	TEXT	
fdeclaracion_testigo	DATE	

Indexes

B.18 Diagrama E-R de la entidad permiso

parroquia permiso	
matrimonio_acta	INT(11) NN
idPermiso	INT(11) NN AI
fpermiso	DATE
presbitero_delegado	VARCHAR(255)
parroquia_emisora	TEXT
dparroquia_emisora	TEXT
dioparroquia_emisora	VARCHAR(255)
finicio_tramites	DATE
presbitero_sus	VARCHAR(255)
nparroquia	TEXT
dparroquia	TEXT
obispodioparroquia	VARCHAR(255)
tipo_permiso	VARCHAR(255)

Indexes

B.19 Diagrama E-R de la base de datos documentos



B.20 Diagrama E-R de la entidad documento

parroquia.documentos	
id_documentos	INT PK AI
fecha	TIMESTAMP
tipo	VARCHAR(45)
Indexes	

B.21 Diagrama E-R de la entidad aob

parroquia.aob	
id_aob	INT(11) PK AI
fechaemita	DATE
basamento	TEXT
sacerdote	VARCHAR(255)
genero	VARCHAR(45)
fnacimiento	DATE
lnacimiento	TEXT
nombre	VARCHAR(255)
nombre1	VARCHAR(255)
padre	VARCHAR(255)
madre	VARCHAR(255)
padrina	VARCHAR(255)
madrina	VARCHAR(255)
contayente	VARCHAR(255)
pmatrimonio	TEXT
fbrn	INT(11)
fbrj	INT(11)
acta	INT(11)
fe expedicion	DATE
parroco	VARCHAR(255)
parroquia_camices	TEXT
resugo	VARCHAR(70)
resugo1	VARCHAR(70)
dam	TEXT
rc	VARCHAR(255)
entidad	VARCHAR(255)
delegacion	VARCHAR(255)
jugado	VARCHAR(255)
actarc	INT(255)
ano	VARCHAR(255)
fic	DATE
fbrarc	VARCHAR(255)
fbrjrc	VARCHAR(255)
localidad	VARCHAR(255)
clase	VARCHAR(45)
documentos_id_documentos	INT PK
Indexes	

B.22 Diagrama E-R de la entidad edm

parroquia.edm		
expediente INT NN AI	notificacionc VARCHAR(70)	comunionr VARCHAR(3)
apellidos VARCHAR(70)	ec VARCHAR(45)	comunione VARCHAR(4)
novio VARCHAR(70)	enovia VARCHAR(4)	ecc VARCHAR(45)
tel VARCHAR(20)	padre VARCHAR(70)	enovioc VARCHAR(4)
novia VARCHAR(70)	madre VARCHAR(70)	padrec VARCHAR(70)
telc VARCHAR(20)	Inacimiento TEXT	madrec VARCHAR(70)
sacerdote VARCHAR(70)	fnacimiento DATE	Inacimientoc TEXT
fecha DATE	rel VARCHAR(45)	fnacimientoc DATE
Imatrimonio TEXT	ge VARCHAR(45)	relc VARCHAR(45)
pmatrimonio TEXT	po VARCHAR(70)	gec VARCHAR(45)
dparroquia_m atrimonio TEXT	dcalle VARCHAR(45)	poc VARCHAR(70)
diopmatrimonio VARCHAR(45)	dnum VARCHAR(4)	dcallec VARCHAR(45)
fmatrimonio DATE	dcol VARCHAR(45)	dnum c VARCHAR(4)
pcompleta BOOLEAN	dmun VARCHAR(45)	dcolc VARCHAR(45)
doccompleta BOOLEAN	dedo VARCHAR(45)	dmunc VARCHAR(45)
platicas BOOLEAN	dcp VARCHAR(10)	dedoc VARCHAR(45)
proclamas BOOLEAN	parroquia TEXT	dcpc VARCHAR(10)
fppublicacion DATE	otrosdoms TEXT	parroquiac TEXT
fspublicacion DATE	tiempodom s TEXT	otrosdomsc TEXT
ftpublicacion DATE	lbautismop TEXT	tiempodom sc TEXT
parroquias_exhorto TEXT	lbautismod TEXT	lbautismopc TEXT
dispensase BOOLEAN	lbautismodio VARCHAR(45)	lbautismodc TEXT
dispensasc VARCHAR(70)	fbautismo DATE	lbautismodioc VARCHAR(45)
licenciase BOOLEAN	librob INT(11)	fbautismoc DATE
licenciasc VARCHAR(70)	fojab INT(11)	librob c INT(11)
celebracione BOOLEAN	actab INT(11)	fojab c INT(11)
celebracionc VARCHAR(70)	lconfirmacion TEXT	actab c INT(11)
	fconfirmacion DATE	lconfirmacionc TEXT

- ◇ fconfirmacionc DATE
- ◇ comuniorc VARCHAR(3)
- ◇ comunione c VARCHAR(4)
- ◇ libre_consentimiento VARCHAR(10)
- ◇ compromiso_anterior VARCHAR(10)
- ◇ quien VARCHAR(70)
- ◇ tcompromiso VARCHAR(45)
- ◇ noviazgo_duracion VARCHAR(4)
- ◇ calidad_noviazgo VARCHAR(45)
- ◇ odm matrimonio VARCHAR(3)
- ◇ disposicion VARCHAR(3)
- ◇ razon VARCHAR(100)
- ◇ conocimiento_sacramento VARCHAR(3)
- ◇ cualidades_matrimonio VARCHAR(3)
- ◇ compromiso VARCHAR(3)
- ◇ formacion_religiosa VARCHAR(45)
- ◇ disposicion_matrimonio VARCHAR(3)
- ◇ impedimento VARCHAR(3)
- ◇ salud VARCHAR(45)
- ◇ vicios VARCHAR(45)
- ◇ vicioso VARCHAR(45)
- ◇ libre_consentimientoc VARCHAR(3)
- ◇ compromiso_anteriorc VARCHAR(3)
- ◇ quienc VARCHAR(70)
- ◇ tcompromisoc VARCHAR(45)
- ◇ noviazgo_duracionc VARCHAR(4)
- ◇ calidad_noviazgoc VARCHAR(45)
- ◇ odm matrimonioic VARCHAR(3)
- ◇ disposicionc VARCHAR(3)

- ◇ razonc VARCHAR(70)
- ◇ conocimiento_sacramentoc VARCHAR(3)
- ◇ cualidades_matrimonioc VARCHAR(3)
- ◇ comprom isoc VARCHAR(3)
- ◇ formacion_religiosac VARCHAR(45)
- ◇ disposicion_matrimonioc VARCHAR(3)
- ◇ impedimientoc VARCHAR(3)
- ◇ saludc VARCHAR(3)
- ◇ viciosc VARCHAR(45)
- ◇ viciosoc VARCHAR(45)
- ◇ ntestigo VARCHAR(70)
- ◇ dtestigo TEXT
- ◇ ttestigo VARCHAR(20)
- ◇ didtestigo VARCHAR(45)
- ◇ tcontrayente VARCHAR(45)
- ◇ motivo TEXT
- ◇ econtrayente VARCHAR(45)
- ◇ lcontrayente VARCHAR(3)
- ◇ otra_razon TEXT
- ◇ tquien VARCHAR(70)
- ◇ obstaculo VARCHAR(3)
- ◇ cual TEXT
- ◇ fdeclaracion_testigo DATE
- ◇ ntestigo1 VARCHAR(70)
- ◇ dtestigo1 TEXT
- ◇ ttestigo1 VARCHAR(20)
- ◇ didtestigo1 VARCHAR(45)
- ◇ tcontrayente1 VARCHAR(45)

- ◇ motivo1 TEXT
- ◇ econtrayente1 VARCHAR(45)
- ◇ lcontrayente1 VARCHAR(3)
- ◇ otra_razon1 TEXT
- ◇ tquien1 VARCHAR(70)
- ◇ obstaculo1 VARCHAR(3)
- ◇ cual1 TEXT
- ◇ fdeclaracion_testigo1 DATE
- ◇ ntestigo2 VARCHAR(70)
- ◇ dtestigo2 TEXT
- ◇ ttestigo2 VARCHAR(20)
- ◇ didtestigo2 VARCHAR(45)
- ◇ tcontrayente2 VARCHAR(45)
- ◇ motivo2 TEXT
- ◇ econtrayente2 VARCHAR(45)
- ◇ lcontrayente2 VARCHAR(3)
- ◇ otra_razon2 TEXT
- ◇ tquien2 VARCHAR(70)
- ◇ obstaculo2 VARCHAR(3)
- ◇ cual2 TEXT
- ◇ fdeclaracion_testigo2 DATE
- ◇ ntestigo3 VARCHAR(70)
- ◇ dtestigo3 TEXT
- ◇ ttestigo3 VARCHAR(20)
- ◇ didtestigo3 VARCHAR(45)
- ◇ tcontrayente3 VARCHAR(45)
- ◇ motivo3 TEXT
- ◇ econtrayente3 VARCHAR(45)

- ◇ lcontrayente3 VARCHAR(45)
- ◇ otra_razon3 TEXT
- ◇ tquien3 VARCHAR(70)
- ◇ obstaculo3 VARCHAR(3)
- ◇ cual3 TEXT
- ◇ fdeclaracion_testigo3 DATE
- ◇ cumplimiento VARCHAR(3)
- ◇ conocimiento VARCHAR(3)
- ◇ duracion_noviago VARCHAR(45)
- ◇ padres_conocimiento VARCHAR(3)
- ◇ domicilio_futuro TEXT
- ◇ fsol DATE
- ◇ cumplimiento1 VARCHAR(3)
- ◇ conocimiento1 VARCHAR(3)
- ◇ duracion_noviago1 VARCHAR(45)
- ◇ padres_conocimiento1 VARCHAR(3)
- ◇ domicilio_futuro1 TEXT
- ◇ fsol1 DATE
- ◇ frat DATE
- ◇ dpadre TEXT
- ◇ librom INT(11)
- ◇ fojam INT(11)
- ◇ actam INT(11)
- ◇ fpplatica DATE
- ◇ fsplatica DATE
- ◇ ftplatica DATE
- ◇ horario VARCHAR(45)
- ◇ fmabrimonioc DATE

- ◇ Imabrimonioc TEXT
- ◇ lc INT(11)
- ◇ fc INT(11)
- ◇ ac INT(11)
- ◇ oficialia VARCHAR(45)
- ◇ fpresentacion DATE
- ◇ hora TIME
- ◇ costo VARCHAR(45)
- ◇ acepto VARCHAR(70)
- ◇ fexpedicion DATE
- ◇ asunto VARCHAR(255)
- ◇ preceptora VARCHAR(255)
- ◇ dpreceptora VARCHAR(255)
- ◇ fexpedicion1 DATE
- ◇ asunto1 VARCHAR(255)
- ◇ preceptora1 VARCHAR(255)
- ◇ dpreceptora1 VARCHAR(255)
- ◇ fexpedicion2 DATE
- ◇ asunto2 VARCHAR(255)
- ◇ preceptora2 VARCHAR(255)
- ◇ dpreceptora2 VARCHAR(255)
- ◇ fpermiso DATE
- ◇ presbitero_delegado VARCHAR(255)
- ◇ parroquia_emisora TEXT
- ◇ dparroquia_emisora TEXT
- ◇ dioparroquia_emisora VARCHAR(255)
- ◇ finicio_tramites DATE
- ◇ presbitero_sus VARCHAR(255)
- ◇ nparroquia TEXT

- ◇ dparroquia TEXT
- ◇ obispodioparroquia VARCHAR(255)
- ◇ tipo_permiso VARCHAR(255)
- ◇ fpermiso1 DATE
- ◇ presbitero_delegado1 VARCHAR(255)
- ◇ parroquia_emisora1 TEXT
- ◇ dparroquia_emisora1 TEXT
- ◇ dioparroquia_emisora1 VARCHAR(255)
- ◇ finicio_tramites1 DATE
- ◇ presbitero_sus1 VARCHAR(255)
- ◇ nparroquia1 TEXT
- ◇ dparroquia1 TEXT
- ◇ obispodioparroquia1 VARCHAR(255)
- ◇ tipo_permiso1 VARCHAR(255)
- ◇ fpermiso2 DATE
- ◇ presbitero_delegado2 VARCHAR(255)
- ◇ parroquia_emisora2 TEXT
- ◇ dparroquia_emisora2 TEXT
- ◇ dioparroquia_emisora2 VARCHAR(255)
- ◇ finicio_tramites2 DATE
- ◇ presbitero_sus2 VARCHAR(255)
- ◇ nparroquia2 TEXT
- ◇ dparroquia2 TEXT
- ◇ obispodioparroquia2 VARCHAR(255)
- ◇ tipo_permiso2 VARCHAR(255)
- ◇ documentos_id_documentos INT

Indexes ▶

ANEXO C: GLOSARIO

Amonestación

Las Amonestaciones son documentos que se dan a partir del primer domingo después de la presentación matrimonial.

Se publican de 3 a 4 domingos en la parroquia con los mismos datos del exhorto.

Aplicación multinivel

Aplicación compuesta por niveles o tiers, por lo general un frontend, un middle tier y un backend (Gupta 2013, traducción propia).

Archivo parroquial

Conjunto de libros donde están registros los bautizados que han recibido los sacramentos.

Atributo

“Propiedad utilizada para describir una entidad o relación”(Bagui 2011, 327, traducción propia).

Atributo atómico

“Atributo que no puede ser descompuesto o dividido” (Bagui 2011, 75, traducción propia).

Atributo compuesto

“Un atributo compuesto de multiples componentes, cada uno con una existencia independiente”(Bagui 2011, 327, traducción propia).

Atributo derivado

“Un atributo que obtiene un valor calculado o derivado de la base de datos”(Bagui 2011, 327, traducción propia).

Atributos de valores múltiples

“Un atributo que podría tener muchos valores para una entidad”(Bagui 2011, 329, traducción propia).

Backend

El nivel de una aplicación multinivel, también, conocido como backend tier donde se realizan las conexiones a bases de datos o sistemas legados(Gupta 2013, traducción propia).

Base de datos

“Una colección compartida de datos relacionados o lógicamente asociados”(Bagui 2011, 237, traducción propia).

Celebración

“Es un encuentro de los hijos de Dios con su Padre, en Cristo y en el Espíritu Santo, y este encuentro se expresa como un diálogo a través de acciones y de palabras. Ciertamente, las acciones simbólicas son ya un lenguaje, pero es preciso que la Palabra de Dios y la respuesta de fe acompañen y vivifiquen estas acciones, a fin de que la semilla del Reino dé su fruto en la tierra buena. Las acciones litúrgicas significan lo que expresa la Palabra de Dios: a la vez la iniciativa gratuita de Dios y la respuesta de fe de su pueblo” (Librería Editrice Vaticana, 2015).

Clave

“Un atributo o un objeto de datos que identifica únicamente una instancia registro o tupla en una relación” (Bagui 2011, 329, traducción propia).

Clave candidata

“Un atributo o conjunto de atributos que identifican de forma única ocurrencias de un tipo de entidad” (Bagui 2011, 327, traducción propia).

Clave primaria

“Un identificador único para un registro en una tabla en una base de datos relacional; una clave candidata seleccionada de una entidad” (Bagui 2011, 330, traducción propia).

Codificación (codificar, code)

“Forma coloquial de decir programar (es decir, escribir código fuente)” (QuinStreet 2015, traducción propia)

Consagración

Condición de un bautizado que ha optado por la vida consagrada.

Controlador

Es la ejecución o llamada de las funciones o módulos que forman el “Modelo” (Jones 2014, traducción propia).

Cripta

Espacio donde se colocan urnas con difuntos que pertenecen a la parroquia.

CSS

“Abreviatura de Cascading Style Sheets (Hojas de Estilo en Cascada), una nueva característica agregada a HTML que da tanto a los desarrolladores y usuarios de sitios Web más control sobre como las páginas son mostradas. Con CSS, diseñadores y usuarios pueden crear Hojas de Estilo que definen como diferentes elementos, tales como, header (títulos) y links (enlaces), aparecen. Estas hojas de estilo puede ser aplicadas a cualquier página web. El término *en cascada* deriva del hecho de que múltiples hojas de estilo pueden ser aplicadas sobre la misma página Web” (QuinStreet 2015, traducción propia)

DBMS

Sistema de administración de bases de datos.

Dependencia funcional

“Una relación entre dos atributos en una relación (relation). Atributo Y es funcionalmente dependiente de X si el atributo X identifica al atributo Y. Para cada único valor de X, el mismo valor de Y siempre será encontrado” (Bagui 2011, 328, traducción propia).

Diagrama Entidad-Relación

“Un **diagrama Entidad-Relación** es una herramienta gráfica que facilita el modelado de datos” (Bagui 2011, 67)

Dispensa

Gracia concedida por el Romano Pontífice para indicar que le matrimonio se celebró pero no está consumado (Librería Editrice Vaticana, 2015).

Entidad

“Algo en el mundo real que es de importancia para un usuario y que necesita ser representado en una base de datos de tal forma que información sobre la entidad pueda ser registrada. Una entidad puede tener existencia física (tal como un estudiante o un edificio) o podría tener existencia conceptual (como un curso)” (Bagui 2011, 328, traducción propia).

Entidad débil

“Una entidad que depende de otra para su existencia”(Bagui 2011, 331, traducción propia).

Entidad fuerte

“Una entidad que no depende de otra para su existencia”(Bagui 2011, 331, traducción propia).

Entidad intersección

Entidad que se desarrolló de una relación (Bagui 2011, 193, traducción propia).

Especialización

“El proceso de maximizar las diferencias entre miembros de una entidad superclase por identificar sus características distintivas” (Bagui 2011, 330, traducción propia).

Esquema

“Un **esquema** implica una descripción fija y permanente de la estructura de los datos” (Bagui 2011, 68, traducción propia).

Esquema de modelado de datos

“Un **esquema de modelado de datos** es un método que nos permite modelar o representar una base de datos “(Bagui 2011, 67).

Exhorto

Es un documento que se manda cuando algún contrayente pertenece a otra parroquia, es decir, vive fuera del territorio parroquial.

Lo llevan a la parroquia que pertenece donde lo publican de 3 a 4 domingos, para ver si no hay ningún impedimento, se publica nombres de los novios, papas, lugar de nacimiento, edad, dirección y sus fotos.

Feligrés

Miembro de una parroquia.

Framework

Una definición muy completa propuesta por TechTarget es la siguiente:

En general, un framework es una estructura real o conceptual que tiene la intención de servir como un soporte o guía para la construcción de algo que expande la estructura en algo más útil.

En sistemas de computadoras, un framework frecuentemente es una estructura en capas que indican que tipo de programas pueden o deberían ser construidas y como podrían estar interrelacionadas. Algunos framework de sistemas de computadoras también incluyen programas, interfaces de programación de aplicaciones u ofrecen

herramientas de programación para usar los frameworks. Un framework podría ser un conjunto de funciones dentro de un sistema y como están interrelacionados; las capas de un sistema operativo; las capas de un subsistema aplicación; como la comunicación debería ser estandarizada en algún nivel de la red; y así sucesivamente (2015, traducción propia).

Esta definición es extensa y completa tanto hablando en general como en el área de tecnologías de la información, por eso consideré adecuada integrarla en el glosario.

Frontend

El nivel de una aplicación multinivel, también, conocido como frontend tier donde el usuario interactúa con la aplicación mediante una interfaz desarrollada en uno o varios frameworks web(Gupta 2013, traducción propia).

Función

“En programación, se llama a una sección de un programa que realiza una tarea específica. En este sentido, una función es un tipo de procedimiento o rutina” (QuinStreet 2015, traducción propia)

Generalización

“El proceso de minimizar las diferencias entre entidades por identificar sus características en común y removerlas dentro de una entidad superclase” (Bagui 2011, 328, traducción propia).

HTML5

“Es una especificación de W3C que define la quinta mayor revisión del Lenguaje de Etiquetado de Hipertexto (HTML)” (QuinStreet 2015, traducción propia)

Intención

Deseos expresados por los feligreses a ofrecer a Dios en la Sagrada Eucaristía.

Jerarquía de clases

Se le denomina al hecho de que una clase principal define e implementa el comportamiento común para un conjunto de clases, a su vez otras se derivan de esta y muchas otras de estas (Oracle 2015, traducción propia)

Libro de bautismo

Libro donde se registran los nombres de los bautizados en el territorio parroquial.

Libro de confirmación

Libro donde se registran los nombres de los confirmados en el territorio parroquial.

Libro de matrimonio

Libro donde se registran los nombres de los feligreses que celebran el matrimonio.

Libro de primera comunión

Libro donde se registran los nombres de aquellos que reciben la Sagrada Eucaristía por primera vez en el territorio parroquial.

Mapeo

“**Mapeo** es el proceso de elegir un modelo lógico y entonces pasar de un modelo conceptual (el diagrama E-R) a un sistema de archivos de bases de datos físico” (Bagui 2011, 327, traducción propia).

Método

“ En programación orientada a objetos, un procedimiento que es ejecutado cuando un objeto recibe un mensaje. Un método es realmente lo mismo como un procedimiento, función o rutina, en lenguajes de programación procedimental. La única diferencia es en programación orientada a objetos, un método está siempre asociado con una clase” (QuinStreet 2015, traducción propia)

Middle tier

El nivel de una aplicación multinivel donde se realizan las transacciones u operaciones con la información obtenida de las bases de datos (Gupta 2013, traducción propia).

Modelo

Hace alusión a las clases, métodos, funciones o módulos que generan la infraestructura de la aplicación (Jones 2014, traducción propia).

Modelo conceptual

“Se considera un **modelo conceptual** a la descripción abstracta de los datos” (Bagui 2011, 68).

Modulo

“Es algo que puede ser cargado con require() en un programa Node.js”(npm 2015, traducción propia)

MongoDB

“Base de datos open source escalable y de alto rendimiento diseñada para manejar almacenamiento orientado a documentos” (QuinStreet 2015, traducción propia)

MySQL

“Es un RDBMS(Sistema de administración de bases de datos relacional) que utiliza SQL para procesar los datos en la base de datos” (QuinStreet 2015, traducción propia)

Notificación

Documento que se envía a las parroquias donde fueron bautizados los contrayentes una vez celebrado el matrimonio. Y que se responde con otro documento proveniente de dichas parroquias, para asegurar que esos datos ya fueron asentados en los libros de bautismo.

Paquete

Un paquete es:

- a) Un directorio que contiene un programa descrito por un archivo package.json.
- b) Un tarball comprimido con gzip que contiene (a)
- c) Una url que resuelve a (b)
- d) Un <nombre>@<versión> que es publicado en el registro con (c)
- e) Un <nombre>@<etiqueta> que apunta (d)

- f) Un <nombre> que tiene una última etiqueta que satisface (e)
- g) Una url de git, cuando es clonada, que resulta en (a)

npm(2015, traducción propia)

Parroquia

“La parroquia es una determinada comunidad de fieles constituida de modo estable en la Iglesia particular, cuya cura pastoral, bajo la autoridad del Obispo diocesano, se encomienda a un párroco, como su pastor propio” (Librería Editrice Vaticana, 2015).

Patrón de diseño

La mejor forma en la que se aborda o soluciona un problema específico, es decir, un problema se presentó y se resolvió de muchas maneras, la mejor fue adoptada más y más hasta hacerla popular, en la popularidad revisada, optimizada y estandarizada para su implementación (Jones 2014, traducción propia).

PHP

“Abreviación de Preprocesador de hipertexto, es un lenguaje de scripts incrustados en HTML, open source y del lado del servidor utilizado para crear páginas web dinámicas” (QuinStreet 2015, traducción propia)

Proclama

Publicación para dar a conocer a la gente que se contraerá matrimonio y para que la comunidad parroquial notifique ante algún impedimento (como ya haber estado casado por la iglesia).

Programación orientada a objetos

“Un tipo de programación en la cual los programadores definen no solo el tipo de dato de una estructura de datos, sino también los tipos de operaciones (funciones) que pueden ser aplicadas el tipo de estructura de datos. De esta forma, la estructura de datos se convierte en un objeto que incluye tanto datos como funciones. Además, los programadores puede crear relaciones entre los objetos” (QuinStreet 2015, traducción propia)

Queries

Plural de query.

Query

“Una solicitud de información desde una base de datos” (QuinStreet 2015, traducción propia)

Relación (relation)

“Una tabla que contiene registros de valor-único y no renglones duplicados. El significado de las columnas es el mismo en cada renglón, y el orden de los renglones y columnas no es importante. Frecuentemente una relación es definida como una tabla poblada” (Bagui 2011, 330, traducción propia).

Relación (relationship)

“Una asociación entre entidades” (Bagui 2011, 330, traducción propia).

Relación binaria

“Relación entre dos entidades” (Bagui 2011, 327, traducción propia).

Relación ternaria

Relación necesaria cuando una relación binaria no es suficiente para describir exactamente la semántica de una asociación entre tres entidades (Bagui 2011, 230, traducción propia).

Sacramentos

“son acciones de Cristo y de la Iglesia, son signos y medios con los que se expresa y fortalece la fe, se rinde culto a Dios y se realiza la santificación de los hombres, y por tanto contribuyen en gran medida a crear, corroborar y manifestar la comunión eclesial” (Librería Editrice Vaticana, 2015).

Sistema de información

“Es un sistema que colecta y almacena datos” (QuinStreet 2015, traducción propia)

Stack (Web stack)

“Término utilizado para referirse a pilas (conjuntos) de software en ambientes de desarrollo Web. La pila de software, principalmente esta compuesta de software open source, contendrá un sistema operativo, un servidor Web, un servidor de bases de datos, y un lenguaje de programación” (QuinStreet 2015, traducción propia)

Template

“Algo que se establece o sirve como patrón para referencia” (QuinStreet 2015, traducción propia)

Vista

Se refiere interfaz de interacción o presentación de datos obtenidos mediante el uso de un controlador (Jones 2014, traducción propia).

ÍNDICE

A

ANEXO A	
DIAGRAMAS ESTILO CHEN DEL CAPÍTULO 4.....	101
ANEXO B	
DIAGRAMAS GENERADOS EN MYSQL WORKBENCH DEL	
CAPÍTULO 5	119
ANEXO C	
GLOSARIO.....	135

C

CAPÍTULO 1 INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE .	1
---	---

CAPÍTULO 2 PLANTEAMIENTO DEL PROBLEMA	20
CAPÍTULO 3 REQUERIMIENTOS DEL SISTEMA.....	23
CAPÍTULO 4 ESPECIFICACIÓN DEL SISTEMA	25
CAPÍTULO 5 DISEÑO DEL SISTEMA	66
CAPÍTULO 6 DESARROLLO DEL SISTEMA	70
CAPÍTULO 7 IMPLEMENTACIÓN DEL SISTEMA	93
CONCLUSIONES	97

I

INTRODUCCIÓN.....	iv
-------------------	----