Implementación de la arquitectura de datos usando el modelo relacional y MySQL Community Edition para el diseño de la base de datos del sistema de capellanía UM

Abner Alejandro Contreras Sánchez
Ingeniería en Tecnologías de la Información y Comunicación/ Facultad de Ingeniería y tecnología
Universidad de Montemorelos

Este proyecto se realizó con el fin de facilitar la gestión de información de datos de estudiantes para el servicio de capellanía en las facultades de la UM y para mejorar algunas actividades esenciales en servicio, tales como, gestión de las entrevistas, la captura, recolección de datos y, mantenimiento de la información por parte de los capellanes. Para cubrir esta necesidad se ha desarrollado una aplicación web que se conecta a un servidor de datos MySQL con una base de datos diseñada con las normas y criterios del modelo relacional para almacenar los datos del alumno y agendar las actividades propias del servicio de capellanía.

Keywords: Arquitectura cliente servidor, arquitectura de datos, bases de datos relacionales, modelo relacional.

Introducción

Este proyecto tiene como propósito diseñar la arquitectura de los datos para la base de datos del Sistema de capellanía de la UM, además se creará usando el modelo relacional de las bases de datos SQL [1], para ejecutar las operaciones de almacenamiento, acceso, mantenimiento y recuperación de los datos en un sistema de gestión de bases de datos de licencia libre para reducir costos de operación.

Sabemos que la tecnología evoluciona constantemente, tiene un crecimiento exponencial, un claro ejemplo de ello es la reducción del uso de papel. Las organizaciones optan por digitalizar el contenido de sus documentos, por esta razón se realizará una base de datos para brindar un servicio más eficiente en las capellanías de las facultades y mantener a disposición esta información todo el tiempo. Explicado lo anterior, se ha optado por almacenar la información de los alumnos en bases de datos relacionales con el propósito de facilitar algunas actividades del capellán en turno.

Otro punto importante es el diseño y la normalización de la base de datos, debido a que la aplicación girará en torno a aspectos específicos de la capellanía y que están relacionados con aspectos académicos, personales, de financiamiento estudiantil, espirituales, etc., se necesitará que el modelo diseñado de la base de datos sea eficiente y útil para extraer reportes los cuales se analizaran para obtener información que puede generar estadísticas sobre el alumnado en la Universidad de Montemorelos.

El departamento de capellanía es un área de vital importancia para las facultades y escuelas de la Universidad de Montemorelos y busca apoyar de forma sistemática a los estudiantes a través de la consejería de carácter personal por parte del capellán, incluso mediante una ayuda efectiva a los estudiantes más necesitados. Para ello necesitan información actualizada para que su colaboración tenga mejores resultados. Frecuentemente, a través de la mentoría espiritual y las entrevistas estipuladas por el consejo de capellanes entablan conversaciones para dar seguimientos a las situaciones de cualquier índole que el estudiante esté experimentando.

Antecedentes

Durante cada semestre del ciclo escolar las asignaturas del currículo del legado institucional que manifiestan la cosmovisión y los valores ideales de la UM son impartidas en todas las carreras, estas materias son aprovechadas para tener una entrevista entre el alumno y el capellán de su facultad. En el proceso se llena de forma manual un formulario impreso se archiva para extraer información en el futuro, dependiendo del tiempo con el que se cuente durante la entrevista, el formulario será llenado por el alumno y el capellán, todo de manera manual, cabe resaltar que cada facultad tiene su propio diseño de formulario y hasta antes de iniciado el presente proyecto no existía un consenso para determinar qué datos son realmente útiles.

Los datos que se almacenarán pueden servir como bases de datos para realizar proyectos de investigación que ayuden a la capellanía y a la UM para tomar decisiones que beneficien el entorno estudiantil.

Identificación del problema

La captura de datos en la forma actual es bastante lenta, y se corre el riesgo de que no sean compartidos los datos en su totalidad, esto limita la disposición de los capellanes para ayudar a sus estudiantes. En el diálogo y con los capellanes para la recolección y selección de requerimientos este proyecto, comparten que los datos recolectados por medio del formulario no se pueden actualizar con facilidad debido a que los alumnos pueden cambiar su información de acuerdo con las condiciones, tanto del ámbito personal, como el estudiantil, por ejemplo (número telefónico, residencia, carrera, etc.); otra situación que comúnmente se experimenta es que como la información está en archivos basados en papel estos pueden extraviarse debido a la falta de sitios seguros y privados donde los documentos puedan mantenerse sin que se deterioren.

Hipótesis

El diseño de la arquitectura de datos para el software de gestión de datos del departamento de capellanía se realizará de acuerdo con los parámetros de la Arquitectura de cliente servidor y el diseño de la base de datos mediante la aplicación del modelo relacional, lo cual facilitará almacenamiento, recuperación y mantenimiento de los datos que se recopilan durante la entrevista.

Propósito de la propuesta

El propósito de este proyecto es hacer eficiente el manejo de información de estudiantes al que tienen acceso los capellanes y facilitar a los estudiantes la captura de esta información. La base de datos estará disponible en cualquier momento, así como en
cualquier plataforma y sistema operativo. Disminuirá radicalmente el uso del papel, mantendrá la información actualizada y organizada, agilizará el proceso de agendar una entrevista con el capellán, estará protegida mediante los recursos a los que tenemos acceso contra robo de información y poder tener un mantenimiento fácil y sencillo, así como la capacidad de migrar de servidores.

Objetivo general

Implementar el uso de MySQL como gestor de la base de datos en la que se almacena la información adquirida por medio de la aplicación web a través de los formularios HTML previamente enlazados con las tablas creadas en la base de datos. Actuando como el modelo relacional [2].

Objetivos específicos

Desarrollar la base de datos para centralizar los datos en un único servidor para que por la vía de una aplicación web puedan ser accedidos sin mayor problema por los capellanes.

Permitir el acceso a la base de datos al administrador, siendo que la información a manejar será confidencial. Mantener una correcta conexión con la aplicación web para evitar caídas en el sistema y pérdida de información. Mantener y monitorear la normalización de la base respecto a los requerimientos que vayan surgiendo de parte los datos del usuario.

Limitaciones

Existen varios obstáculos para la realización de este proyecto, entre los cuales se encuentra el tiempo ya que el diseño en el modelo relacional requiere de un trabajo minucioso, para que a partir

de este diseño la creación de la base de datos sea correcta.

Conseguir un servidor, montarlo y configurarlo en las computadoras de los participantes de este proyecto también requiere tiempo.

El software que utilizaremos para crear la aplicación y la conexión de la base de datos puede variar dependiendo de su versión, así como su sintaxis, lo cual supondrá un obstáculo al momento de empezar a crear código.

La compatibilidad de los distintos frameworks y sus lenguajes de programación representan un reto al momento de estar combinándolos para obtener el mejor rendimiento.

Delimitaciones

Durante la creación del proyecto podrían surgir nuevas ideas y requisitos, no obstante, por la duración de nuestra participación en este proyecto solo se trabajará con los siguientes elementos de requerimientos:

- ☐ El sistema será de uso exclusivo para capellanes y alumnos.
- ☐ Se enfocará únicamente en la gestión y control de los datos de los alumnos.
- ☐ La fase de pruebas se hará con datos propios, esperando la aprobación de parte de los administrativos.
- ☐ Solo se cumplirá con los requerimientos establecidos por el periodo de tiempo de trabajo.
- ☐ Los nuevos requerimientos y labores de mantenimiento se dejarán para futuros desarrollos.

Definición de términos

Modelo relacional

El modelo relacional está basado en el principio matemático de relación [2]. Que para estos fines se representa gráficamente como una tabla.

Este modelo almacena todos los datos en las relaciones, cada relación es un conjunto de datos. La información en este modelo no tiene un orden de almacenamiento, la organización de los datos se efectúa al hacer las consultas. De esta manera la información no se hace confusa [2].

La tabla contiene filas y columnas, cada columna se conoce como “campo” y cada fila es un “registro”.

matricula	usuario	nombre	apellidos	fechanacimiento
1140583	yair	Yair	Castillo	1999-10-06
1160004	eunice	Eunice	Lopez	1997-11-09
1160592	Alejandro	Alejandro	Contreras	1997-12-02
7654321	Yair Castillo	Pablo Yair	Castillo Sanchez	1999-10-06

Fig. 1. Partes de una tabla relacional.

MVC

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa una aplicación en tres componentes principales: modelos, vistas y controladores.

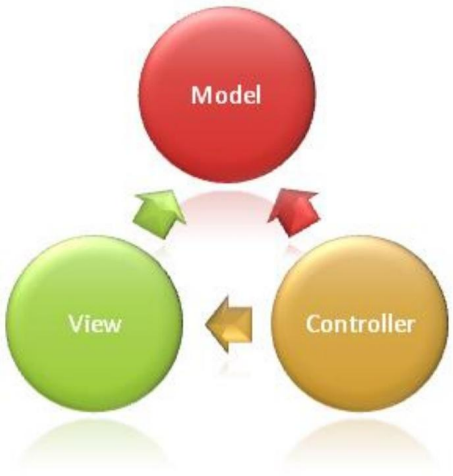

Fig. 2. Partes del MVC. [3].

En la imagen se muestran los tres componentes y cual hace referencia a los demás componentes. El acceso a la base de datos es proporcionado por el controlador el cual crea la conexión, y el modelo es el que administra la información recibida y enviada, siendo este último el que genera las peticiones y sentencias SQL a la base de datos.

Arquitectura de datos

La Arquitectura de datos es la integración de modelos, políticas y reglas que rigen qué datos serán recopilados, cómo serán almacenados, ordenados y puestos en uso mediante determinada infraestructura tecnológica [4].

Arquitectura cliente servidor

Es un modelo de trabajo de software, en la que participan dos partes, un cliente y un servidor, interactuando mayormente por una red. Los clientes solicitan un servicio al servidor y el servidor devuelve la petición realizada [5][6].

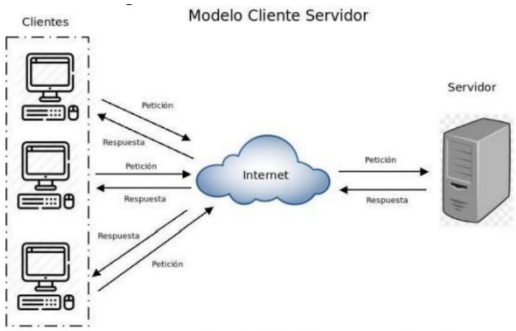

Fig. 3. Representación del Modelo Cliente-Servidor

SQL

El lenguaje de consulta estructurada (SQL) es un lenguaje creado para administrar y recuperar información de los sistemas de gestión de las bases de datos.

Es el lenguaje estándar en las bases de datos relacionales. SQL se considera un lenguaje de no procesamiento, aun así, un es lenguaje completo para crear y mantener objetos en una base de datos, asegurar esos objetos y manipular la información dentro de los objetos [7][8][9][10][11].

Cabe resaltar la importancia de este lenguaje, debido a que se asocia de una manera inmejorable con las bases de datos relacionales, la razón es porque SQL se basa en el modelo relacional.

SQL contiene precompiladores los cuales crean una implementación específica para el lenguaje host y otra implementación específica para SQL.

Las instrucciones SQL están categorizadas en tres tipos de instrucciones [12], [11]:

Lenguaje de definición de datos (DDL, Data Definition Language)

Estas instrucciones son utilizadas para crear, modificar o eliminar objetos de en una base de datos, tales como tablas, vistas, esquemas, dominios, activadores y almacenar procedimientos. Y para efectuar su función se valen de palabras clave:

- CREATE:
  Permite crear bases de datos o tablas.
- ALTER:
  Permite cambiar las propiedades de la base de datos o tablas existentes.
- DROP:
  Permite borrar bases de datos, tablas o campos.

Lenguaje de control de datos (DCL, Data Control Languague)

Las instrucciones DCL gestionan el acceso a la base de datos. Con DCL se otorga o restringe el acceso usando las siguientes instrucciones:

- GRANT:
  Otorga permisos a un usuario.
- REVOKE:
  Quita los permisos a un usuario.

Lenguaje de manipulación de datos (DML, Data Manipulation Language)

Las instrucciones DML recuperan, agregan, modifican y borran datos almacenados en los objetos de una base de datos. Las palabras claves son:

- INSERT:
  Permite insertar datos en una tabla
- UPDATE:
  Permite modificar los datos existentes.
- DELETE:
  Permite borrar los datos.
- SELECT:
  Permite hacer consultas de los datos que se encuentran en las tablas.

Base de datos relacionales

Una base de datos relacional es un tipo de base de datos basada en el modelo relacional, proporcionando acceso a puntos de datos relacionados entre ellos. Cada registro (fila)

contiene un ID único llamado clave [13]. Las columnas de las tablas contienen atributos, y cada registro (columna) generalmente tiene un valor para cada atributo. Como se ilustra en la Figura 1.

La base de datos de MySQL cuenta con un mecanismo de almacenamiento de datos de código abierto, InnoDB, un motor de almacenamiento transaccional [14], [15]. La característica principal es que soporta transacciones de tipo ACID [16] y bloqueo de registros e integridad referencial [17] [18]. InnoDB realiza bloqueos en el nivel de fila y también proporciona funciones de lectura consistente sin el bloqueo de Oracle en las sentencias SELECT. MySQL contiene cinco tipos índices [19], [20], [21] [22]:

- INDEX:
Es un índice normal, por lo cual admite valores duplicados para las columnas que componen el índice. No genera ninguna restricción especial a los datos de las columnas. Es empleado para mejorar el tiempo de ejecución de las consultas.
- UNIQUE:
En este índice usado cuando todas las columnas deben tener un valor único, por lo que no admite valores duplicados para las columnas que sean parte del índice.
- PRIMARY
En este índice todas las columnas deben tener un valor único (como el índice UNIQUE), pero con la limitación que solo puede haber un índice PRIMARY por cada tabla.
- FULLTEXT
Este índice es empleado para hacer búsquedas sobre campos de texto (CHAR, VARCHAR y TEXT). No tienen restricción.
- SPATIAL
Estos índices son usados para realizar búsquedas sobre datos que sean partes de figuras geométricas representadas en el espacio.

Las llaves o claves primarias son identificadores únicos o de varias columnas NOT NULL que identifica de forma única una fila de una tabla. Una tabla sólo puede tener una PRIMARY KEY, y se recomienda que cada tabla tenga una. InnoDB crea automáticamente una en caso de no contar con una [15].

Ahora, las llaves foráneas (FOREIGN KEY) son una limitación referencial entre dos tablas. La clave foránea identifica un campo en una tabla (tabla hija o referendo) que se refiera a un campo de otra tabla (tabla padre o referenciada). El campo en la tabla padre debe ser una clave primaria [23].

Las llaves tanto primarias como foráneas tienen una estrecha relación con la normalización, ya que, al normalizar una tabla, esta tiende a dividirse, por lo cual es necesario mantener una relación entre las tablas creadas, aquí ocupan un papel esencial las llavestanto primarias como foráneas, ya que el correcto funcionamiento y el éxito de la normalización dependen en gran medida que estas llaves estén correctamente relacionadas. Al cumplir estos requerimientos se debe tener en cuenta la dependencia funcional y la integridad referencial, las cuales hacen alusión directamente a parámetros de relación creados por las llaves primarias y foráneas [24].

MySQL

El software MySQL es un sistema de gestión de bases de datos relacional basado en el lenguaje SQL. El servidor MySQL está diseñado para entornos intensivos de lectura de datos, lo quelo hace ideal para aplicaciones web.

La versión phpMyAdmin (entorno gráfico de MySQL) 4.8.5 fue la que se usó en el sistema de capellanía. El servidor web para generar la conexión es Apache/2.4.39 (Win64) PHP/7.3.5. La versión utilizada de MySQL fue la 5.7.26 - MySQL Community Edition. La cual es una de las versiones más recientes de uso gratuito, y contiene características interesantes [25]:

- Está desarrollado en C/C++.
- Se distribuyen ejecutables para diecinueve plataformas diferentes.
- La API se encuentra disponible en C, C++, Eiffel,
Java, Perl, PHP, Python, Ruby y TCL.
- Está optimizado para equipos de múltiples procesadores.
- Es muy destacable su velocidad de respuesta.
- Se puede utilizar como cliente-servidor o incrustado en aplicaciones.
- Cuenta con un rico conjunto de tipos de datos.

- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el SGBD a cada caso concreto.
- Su administración se basa en usuarios y privilegios.
- Se tiene constancia de casos en los que maneja cincuenta millones de registros, sesenta mil tablas y cinco millones de columnas.
- Sus opciones de conectividad abarcan TCP/IP, sockets UNIX y sockets NT, además de soportar completamente ODBC.
- Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.
- Es altamente confiable en cuanto a estabilidad se refiere. En consultas sobre el rendimiento de MySQL respecto a otras bases de datos se puede observar un comportamiento promedio y estable [26], [27].

PHP

PHP es un procesador de hipertexto, un lenguaje de programación de código abierto, el cual es muy amigable con MySQL teniendo una gran capacidad de interacción [28].

La versión utilizada para hacer la conexión a la base de datos fue PHP 7.3.5. El programa se crea enteramente en archivos php, dentro de los cuales se inserta html y JS (JavaScript). PHP es muy versátil respecto a su desempeño.

Se trabaja en el sistema de acuerdo con el modelo vista- controlador, y es en las secciones de modelo y controlador donde el lenguaje interactúa con la base de datos a través de un controlador que otorgue una conexión a la base de datos, y de varios modelos que mandan consultas SQL para ingresar información, borrarla y actualizarla.

Más concretamente existe el archivo datoseventos.php donde se realiza la lógica del CRUD [29] para las citas, también en las consultas SQL de modificar y borrar los datos se implementa la condición de borrar los cambios que solo pertenecen al usuario.

Normalización

La normalización es el proceso de organizar los datos de una base de datos, designando y aplicando una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación. Esto genera la creación de más tablas y establecer más relaciones entre esas tablas con el objetivo de minimizar la redundancia de los datos, reducir los problemas de la actualización de los datos en las tablas y proteger la integridad de estos [30].

Metodología

En esta sección se describen los procesos realizados para la construcción de la base de datos y su acoplamiento con el Sistema de capellanía de la UM.

Diseño de la base de datos

Al plantearse el diseño de la base de datos se contemplaron las diferentes formas de crear la base de datos, y diferentes gestores de bases de datos, y se optó por los gestores que operaban con sentencias SQL, después de ello se decidió usar el gestor MySQL el cual cuenta con una interfaz de trabajo el cual es MySQL.

Debido a que es una base de datos SQL, se basa en el modelo relacional, por lo cual facilita en gran medida la interacción de los datos, ya que el programa exige muchas combinaciones de tablas para resultados específicos.

La base de datos debe almacenar el registro inicial en la aplicación, las citas agendadas y los datos personales correspondientes al usuario (tipo de usuario, capellán o alumno). Por lo que se hizo un diagrama de relación para representar cómo funcionarán las tablas de la base de datos al relacionarse entre ellas.

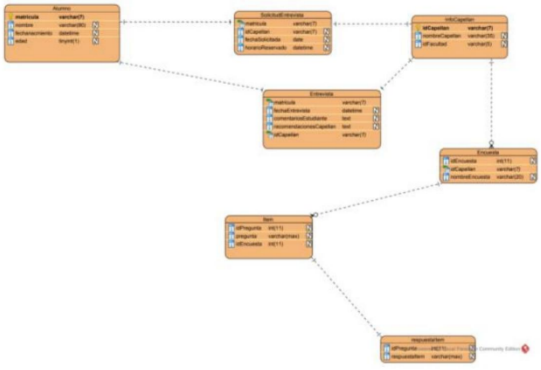

Fig. 8. Diagrama de relación.

La interacción de las tablas a través de las llaves primarias y foráneas es la clave de un buen funcionamiento en la base de datos.

Normalización de la base de datos

La normalización es un proceso necesario, si se desea optimizar la base de datos, por ello se llevó acabo la normalización de la base de datos a su forma 3FN (tercera forma normal) [31].

Al crear la base de datos se encontraba en la forma inicial, por ello, para ascender al primer nivel (1FN, primera forma normal) se debe [32]:

- Eliminar los campos repetitivos de las tablas individuales.
- Generar una tabla separada por cada campo de datos relacionados.
- Identificar cada campo de datos relacionados con una PRIMARY KEY.

Terminado el proceso de la primera forma normal se procede con el segundo nivel. Los requisitos son [33]:

- Crear tablas separadas para los campos que se aplican a varios registros.
- Relacionar las claves mediante una FOREIGN KEY.

Completado el proceso del nivel dos pasaremos al nivel tres de normalización. En este punto solo hay un paso a seguir [34]:

- Eliminar aquellos campos que no dependan de la clave.

Finalizado el tercer nivel ahora si estaremos en la 3FN. Hay que tener en cuenta que existen más niveles de normalización, pero eso no garantiza un correcto funcionamiento, si una base de datos sobrepasa el nivel ideal de normalización podría descomponer su base de datos en demasiadas tablas más pequeñas, lo que llevará inevitablemente a crear consultas más complejas para unir los datos dispersos [35].

Eliminación de la redundancia

La redundancia en una base de datos es el almacenamiento de los mismos datos en diferentes campos. Si no se elimina la redundancia puede ocasionar problemas [36], [37]:

- Incremento del trabajo: Ya que el mismo dato está almacenado en varios lugares, cuando se graban o actualizan los datos, deben hacerse en todos los lugares a la vez.
- Desperdicio del almacenamiento: Al ser los mismos datos repetidos y almacenados en varios lugares, ocupan más bytes del medio de almacenamiento.
- Inconsistencias de datos:

  Si el dato de un lugar es actualizado pero eldato duplicado que está en otro lugar no se actualiza, generará problemas al momento deser usado.

Selección de la arquitectura cliente servidor

Es un modelo de trabajo de software, en la que participan dos partes, un cliente y un servidor, interactuando mayormente por una red. Conocido también como programación por capas. Los clientes solicitan un servicio al servidor y el servidor devuelve la petición realizada. Usualmente este modelo está dividido en capas [6] [38].

Capa del cliente o capa de presentación

Aquí se crean los módulos que llamaran a la capa de negocio solicitando información y desplegará el resultado en pantalla a través de una interfaz gráfica, que se caracteriza por ser amigable para el usuario. Esta capa se comunica únicamente con la capa de negocio [38].

(1) Interfaz de usuario

La interfaz de usuario es el código creado en html, es lo que el usuario mira y con lo que interactúa.

(2) Lógica de interfaz de usuario

Es el código que acompaña a la interfaz de usuario y muchas veces también es html, esta lógica es la que le da la funcionalidad a la interfaz de usuario, debido a esta lógica el usuario interactúa con la interfaz.

Capa de la aplicación o capa de negocio

Aquí se crean los archivos que tendrán las consultas SQL. Esta capa se comunica con la capa de presentación, para recibir solicitudes y devolver los resultados, también se conecta con la capa de datos, para interactuar con el gestor de la base de datos (MySQL) [38].

(*1) Lógica de negocio (controladores)*

Aquí se crean los archivos php donde se controlan las interfaces y se hacen peticiones a la base de datos.

Capa de datos

En esta capa se crea la conexión a la base de datos, recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio [38].

(1) Acceso a servicios

Esta sección se puede ejemplificar cuando un usuario se registra en el sistema y seguidamente ingresa a su cuenta. Desde este punto se validan sus datos y se le da autorización de entrar.

(2) Acceso a datos

El acceso a datos permite el acceso a los mismos, permitiendo la realización del CRUD, si el usuario cuenta con los permisos correspondientes.

Resultados

El programa se realizó con éxito, logrando cumplir los objetivos y los requerimientos del cliente. Yo era el encargado de la sección de conexión a la base de datos, así como la creación de las bases de datos, aun así, el equipo no se limitó a sus tareas. Aporte la lógica en la capa de presentación en la sección de lógica de interfaz de usuario. También trabajamos en la creación de más tablas sin perder el punto de la normalización, ya que como comenté en la sección
de delimitaciones, en el proceso salieron tareas que nunca fueron contempladas y eran necesarias para el funcionamiento del sistema.

Se generó el calendario donde se agendan las citas, el calendario permite, crear, borrar, eliminar y mostrar sus citas, las citas agendadas por un estudiante de una facultad no pueden ser
visualizadas por un usuario de otra facultad. De igual manera las citas creadas por el mismo usuario no pueden ser alteradas por ningún otro usuario, ya que cada cita al ser agendada obtiene el nombre del usuario que crea la cita y es comparado con el nombre de usuario que tiene cada cita, al no coincidir la modificación no procede.

El capellán puede visualizar los datos de los alumnos que pertenecen a su facultad y también tiene la opción de crear eventos (citas) predefinidas, a diferencia de los alumnos.

Conclusiones

Fue un proyecto fácil de plantear, pero complicado al ser ejecutado, debido a la cantidad de detalles que merecen la atención. Pero en el proceso conseguimos confirmar el funcionamiento
de las tecnologías aplicadas. Se comprobó el rendimiento del gestor de la base de datos (MySQL), el cual prometía respuestasrápidas a consultas continuas. Logramos obtener una mejor comprensión de cómo funcionan las bases de datos y que podemos lograr con una consulta bien estructurada. Se obtuvo un poco más de lo planeado. De igual manera el sistema puede ser mejorado, creando una interfaz más atractiva, también se debería analizar el código en la capa de negocio para disminuirlo y reciclar código, haciendo más eficiente el flujo de datos y servicios.

Referencias

[1] S. L. Reference, «Oracle,» 2020. [En línea]. Available: https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/History-of-SQL.html#GUID-4DD5E1B6-BEC7-4E9B-B369-1466F93ACA28. [Último acceso: 20 Abril 2020].

[2] U. N. d. Córdoba, «Ofimática,» [En línea]. Available: http://oftgu.eco.catedras.unc.edu.ar/unidad-3/sistemas-de-gestion-de-base-de-datos/modelo-relacional-conceptos-basicos-y-fundamentos/. [Último acceso: 29 Abril 2020].

[3] S. Smith, «Microsoft Docs,» 12 Febrero 2020. [En línea]. Available: https://docs.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-3.1. [Último acceso: 29 Abril 2020].

[4] A. Osorio Rodríguez y M. Martinez Lopez, «LACCEI,» 14 Agosto 2013. [En línea]. Available:

http://www.laccei.org/LACCEI2013-Cancun/RefereedPapers/RP032.pdf. [Último acceso: 29 Abril 2020].

[5] S. Alvarez, «desarrolloweb.com,» 30 Agosto 2007. [En línea]. Available: https://desarrolloweb.com/articulos/arquitectura-cliente-servidor.html. [Último acceso: 29 Abril 2020].

[6] Oracle, «Docs Oracle,» 2010. [En línea]. Available: https://docs.oracle.com/cd/E19528-01/820-0888/aaubb/index.html. [Último acceso: 29 Abril 2020].

[7] C. E. P. Prado, «DevCode,» 2020. [En línea]. Available: https://devcode.la/blog/que-es-sql/. [Último acceso: 29 Abril 2020].

[8] H. H. L., «Sistema de bibliotecas y bibloteca central,» 2002. [En línea]. Available: http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/indata/v05_n1/dise%C3%B1o.htm. [Último acceso: 29 Abril 2020].

[9] «Support Office,» 2020. [En línea]. Available:https://support.office.com/es-es/article/access-sql-conceptos-b%C3%A1sicos-vocabulario-y-sintaxis-444d0303-cde1-424e-9a74-e8dc3e460671#bm1. [Último acceso: 29 Abril 2020].

[10] «Docs Oracle,» 2020. [En línea]. Available:https://docs.oracle.com/en/database/oracle/orcle-database/19/sqlrf/History-of-SQL.html#GUID-4DD5E1B6-BEC7-4E9B-B369-1466F93ACA28. [Último acceso: 29 Abril 2020].

[11] A. Oppel y R. Sheldon, «PEDRO BELTRAN CANESSA -BIBLIOTECA DE PROGRAMACIÓN Y CIENCIAS,» McGraw-Hill, 2009. [En línea]. Available:https://pedrobeltrancanessa-biblioteca.weebly.com/uploads/1/2/4/0/12405072/fundamentos_de_sql_3edi_oppel.pdf. [Último acceso: 29 Abril 2020].

[12] Universidad de Antioquia, «Aprende en línea,» 30 Abril 2016. [En línea]. Available: http://aprendeenlinea.udea.edu.co/lms/moodle/mod/page/view.php?id=75248. [Último acceso: 29 Abril 2020].

[13] Oracle, «Oracle,» 2020. [En línea]. Available:https://www.oracle.com/mx/database/what-is-a-relational-database/. [Último acceso: 29 Abril 2020].

[14] MySQL, «MySQL,» 2020. [En línea]. Available:https://dev.mysql.com/doc/refman/8.0/en/innodb-introduction.html. [Último acceso: 29 Abril 2020].

[15] Arsys, «arsys,» 4 Abril 2012. [En línea]. Available:https://www.arsys.es/blog/programacion/bases-de-datos/myisam-o-innodb-elige-tu-motor-de-almacenamiento-mysql/. [Último acceso: 29 Abril 2020].

[16] B. A. P. RALDA, «BASE DE DATOS USAC PUERTOBARRIOS,» 7 Agosto 2015. [En línea]. Available:https://usacdatospb.files.wordpress.com/2015/09/reglas-acid-y-codd-gestores-base-de-datos.pdf. [Último acceso: 29 Abril2020].

[17] aulaClic, «aulaClic,» 2020. [En línea]. Available:https://www.aulaclic.es/sql/b_8_1_1.htm. [Último acceso: 29 Abril 2020].

[18] Support Office, «Support Office,» 2020. [En línea]. Available:https://support.office.com/es-es/article/mantener-la-integridad-referencial-en-diagramas-de-modelo-de-base-de-datos-80f60e10-1238-48f7-ab59-2bd31b2f047a. [Último acceso: 29bril 2020].

[19] R. V. Amor, «Adictos al trabajo,» 11 Septiembre 2015. [En línea]. Available:

https://www.adictosaltrabajo.com/2015/09/11/introduccion-a-indices-en-mysql/. [Último acceso: 29 Abril 2020].

[20] S. Mendez, «Salvador Mendez,» 4 Octubre 2011. [En línea].

Available: http://www.sgmendez.com/2011/10/04/tipos-

indices-mysql/. [Último acceso: 29 Abril 2020].

[21] Programacion.net, «Programacion.net,» 2020. [En línea].
Available:
https://programacion.net/articulo/indices_y_optimizacion_de_c
onsultas_305/2. [Último acceso: 29 Abril 2020].
[22] A. Barrena, «Aner Barrena,» 24 Junio 2016. [En línea].

Available:
https://www.anerbarrena.com/mysql-create-index-
5281/. [Último acceso: 29 Abril 2020].

[23] P. Freitas, «RIP tutorial,» 2020. [En línea]. Available:

https://riptutorial.com/es/mysql/example/2725/creacion-de-
tablas-con-clave-primaria. [Último acceso: 29 Abril 2020].

[24] H. Sulbaran, «helisulbaransistemas,» 15 Mayo 2014. [En
línea]. Available:

https://helisulbaransistemas.blogspot.com/2014/05/como-crear-
claves-foraneas-en-mysql.html. [Último acceso: 29 Abril 2020].