

Implementación del patrón arquitectónico MVC en aplicaciones web para la arquitectura del software del sistema de capellanía de la UM

Pablo Yair Castillo Sánchez
Facultad de Ingeniería y Tecnología
Universidad de Morelos

El presente trabajo de investigación tiene como propósito fundamental implementar una aplicación web para el servicio de capellanía de la Universidad de Morelos, con la funcionalidad principal de que los alumnos agenden citas con los capellanes y ellos a su vez puedan gestionar la información de las entrevistas con los alumnos de la facultad que representan. En el Compromiso Educativo de la Universidad de Morelos, se hace referencia al servicio de capellanía, tutoría y consejería y esto se resalta como una parte importante en el legado institucional de la universidad, dado que los pastores delegados en esta área, denominados capellanes se dedican a velar por la vida espiritual de la facultad que representan, a profesores y alumnos por igual, además, una función primordial consiste en el conocimiento de las necesidades y problemas que los estudiantes pudieran experimentar. Se desarrolló un modelo de software para el departamento de capellanía de la Universidad de Morelos, tomando en cuenta que una parte de vital importancia para esta división es la de relacionarse con sus alumnos, acercándose a ellos y detectando sus necesidades. Todo esto mediante encuestas y entrevistas que se realizan periódicamente.

Keywords: Patrón arquitectónico, capellanía, arquitectura de software

Introducción

El presente trabajo de investigación tiene como propósito fundamental implementar una aplicación web para el servicio de capellanía de la Universidad de Morelos, con la funcionalidad principal de que los alumnos agenden citas con los capellanes y ellos a su vez puedan gestionar la información de las entrevistas con los alumnos de la facultad que representan. En el Compromiso Educativo de la Universidad de Morelos (2017) se hace referencia al servicio de capellanía, tutoría y consejería y esto se resalta como una parte importante en el legado institucional de la universidad [1], dado que los pastores delegados en esta área, denominados capellanes se dedican a velar por la vida espiritual la, facultad que representan, a profesores y alumnos por igual, además, una función primordial consiste en el conocimiento de las necesidades y problemas que los estudiantes pudieran experimentar. Para ello, los alumnos deben acudir

a distintas entrevistas con su capellán, que eventualmente genera algunas complicaciones. Esto debido a que, a pesar de que hay un capellán para cada facultad en la universidad, son muchos los alumnos para entrevistar. Teniendo en cuenta que la entrevista debe ser tanto presencial, como privada con el capellán y, que su duración es de 15 a 30 minutos por entrevista. Resulta difícil para los alumnos encontrar un horario disponible para la cita y confuso en la organización de éstas para los capellanes. Además de esto, los capellanes también han expresado que les gustaría tener la información obtenida de esas entrevistas en un formato más accesible, eficiente, organizado e inclusive ecológico debido a la automatización de la captura de datos que sustituye el uso de hojas de papel. El desarrollo del proyecto se divide en dos partes: La arquitectura de software, que a su vez se subdivide en dos elementos fundamentales: Interfaces y Reglas del Negocio; y la Arquitectura de Datos. En este documento, se expondrá la parte de Reglas del Negocio, específicamente hablando de la

aplicación del patrón arquitectónico de arquitectura de software denominado MVC.

Antecedentes

MVC, o Modelo-Vista-Controlador, es un patrón de diseño de software que utiliza y a la vez mantiene separados los tres componentes descritos en esta metodología. Se trata de una arquitectura importante puesto que se utiliza tanto en componentes gráficos básicos, como en sistemas empresariales. Además, es muy utilizado para la arquitectura en la mayoría de los frameworks modernos y uno de los más usados en la industria para poder crear proyectos escalables y modulares [2]. La idea detrás de MVC es que cada uno de los componentes de su código tenga un propósito, y que esos propósitos sean diferentes. Además de que la forma en que se relacionan estas partes ayuda a los desarrolladores a crear código con la ventaja de realizar un mejor mantenimiento en el futuro. Principalmente, tiene como objetivo dar soporte a los modelos funcionales y mapas mentales de la información relevante para los usuarios, permitiendo un modelo que facilite la consulta y manejo de los mismos [3].

El patrón de arquitectura MVC, en su forma original, fue introducido por primera vez con Smalltalk-76 (uno de los primeros lenguajes orientados a objetos que aparecieron en la historia de la computación y uno de los primeros en implementar MVC) por su creador Trygve Reenskaug (1979) y luego implementado en Smalltalk-80 en la década 1980 [4], pero se explicó formalmente por primera vez en un artículo en The Journal of Object Technology (1988). En ese tiempo las cosas eran claramente diferentes a las actuales, dado que MVC había nacido como un estilo de arquitectura GUI (interfaz gráfica de usuario) y como un patrón dedicado exclusivamente al desarrollo de aplicaciones de escritorio. Sin embargo, gracias al crecimiento de las aplicaciones web, se consolidó como uno de los patrones más usados y populares para el desarrollo de aplicaciones web.

Actualmente, se trata de un modelo maduro y el patrón de diseño orientado a objetos más ampliamente difundido para el desarrollo de software, que ha ido demostrando su importancia a lo largo de los años en todo tipo de aplicaciones,

sobre todo las aplicaciones web, y sobre multitud de lenguajes y plataformas de desarrollo, dado que este patrón permite una separación muy clara de los datos de la aplicación que consta de tres partes interconectadas: modelo, vista y controlador.

□ Vista: este elemento hace referencia a la parte de una aplicación que considera la interfaz gráfica. Es decir, cada elemento gráfico que interactúa con el usuario forma parte de la Vista. Su función es obtener la información que requiere el usuario, lo cual se denomina como, evento. Esta capa siempre muestra la información proporcionada por el Modelo [5].

□ Modelo: esta capa tiene la función de relacionar y gestionar los datos con los cuales la aplicación va a operar, como consultas, actualizaciones, creación de información o eliminación. Todo esto se le denomina como, Lógica de Negocio. En otras palabras, esta parte se refiere a la transformación de las actividades del mundo real a la forma en la que se va a modificar la información [5].

□ Controlador: este componente responde ante eventos o acciones que realiza el usuario a través de la Vista para poder solicitar una operación de la información. Además, tiene la tarea de la elección de la Vista que se mostrará al usuario de acuerdo con la solicitud recibida, por lo que es el vínculo entre el Modelo y la Vista [5].

Todo esto da como resultado una gran lista de ventajas al desarrollar software. Entre las más frecuentes de la comunidad se encuentran [6]:

- La reducción de la complejidad del código
- La reutilización de código
- Mayor flexibilidad
- Código desacoplado

Identificación del problema

El Ministerio de Capellanía Adventista fue creado en 1985. Es parte de una corporación entre la iglesia adventista y la División Norteamericana, principalmente, porque tiene que ver con los capellanes militares. Sin embargo, durante los últimos años la iglesia a nivel mundial ha estado promoviendo la necesidad de dicho ministerio. Ello, porque envuelve cientos de capellanes que son pastores en hospitales, instituciones educativas y facilidades correccionales, de acuerdo con Leito [7], presidente para la iglesia en Interamérica. El departamento de capellanía de la Universidad de Morelia ha sido desde sus inicios una parte fundamental en dicha institución. Mediante este

departamento, los capellanes de cada facultad pueden estar al tanto de los problemas que pudiera experimentar algún alumno en su estancia como estudiante inscrito en la institución y los aspectos que pudieran estar relacionados con diferentes ámbitos de su vida cotidiana. Entre las funciones que ejerce un capellán dentro de la UM está el ministrar al personal y los estudiantes, realizar entrevistas con cada uno de los alumnos, llevar a cabo cultos, retiros, vigilias, el servicio comunitario y todo lo relacionado con el ámbito espiritual.

Como parte del cuidado de la vida espiritual de los alumnos, el capellán debe realizar un seguimiento de caso de cada uno de ellos. Para cumplir con esto, es necesario realizar encuestas cada cierto tiempo y registrar la información resultante. Actualmente, este proceso se ha estado efectuando mediante las herramientas más básicas: papel para capturar información junto con carpetas y cajas para almacenarla; haciendo que llevar a cabo este seguimiento sea mucho más difícil y tedioso. Sobre todo, porque, como es de esperarse, la información recuperada al cabo de algunas entrevistas es mucha y la manera de conservarla tampoco ayuda al momento de querer consultarla, modificarla o desecharla. Es por ello que, en un mundo donde la tecnología avanza a pasos agigantados, es imprescindible cambiar de métodos y herramientas a otras más apropiadas para este proceso. No obstante, se debe determinar qué tecnologías son las más adecuadas para un proyecto como este. Con esto en mente, se decidió primero obtener los requisitos y requerimientos que necesitaría el sistema.

Para ello, se realizó una encuesta a los capellanes con las siguientes preguntas:

- ¿Cuáles son las principales problemáticas a las que se enfrenta en su área?
- ¿Qué aspectos espera que se automaticen con el sistema?
- ¿Algunas características que espera que cuente el sistema?

De acuerdo con esta encuesta, cada uno de ellos opina que las principales problemáticas a las que ellos se enfrentan son:

- Seguimiento a los alumnos de forma más práctica
- Organización

Y en general coinciden en:

- Gasto en hojas de papel,
- Generación de documentos físicos extensos,
- Dificultades en la accesibilidad a la información de los alumnos

En cuanto a los aspectos que ellos esperan que se automaticen se encuentran:

- Bitácoras para un mejor seguimiento de los alumnos
- El registro de asistencia a las actividades espirituales
- La planificación
- Autoevaluaciones

Y en general coinciden en:

- Registros de entrevistas
- Información de los alumnos

Además de esto, los capellanes esperan que el sistema también cuente con las siguientes funciones:

- Mostrar a los alumnos que se encuentren en problemas o en riesgo, tales como problemas económicos, familiares, emocionales, de salud; o riesgo de reprobar una materia o tener alguna deuda pendiente.
- Un espacio para compartir información entre los capellanes
- Alumnos que cumplen años

Una vez expuestos estos requisitos y formulados ciertos requerimientos, se concluyó que la mejor propuesta de solución consiste en la creación de una aplicación web, debido a sus beneficios ante este tipo de proyecto a comparación con otros softwares. A continuación, algunas de las ventajas que vale la pena resaltar [8]:

- Compatibilidad multiplataforma: de esta forma, no importa el sistema operativo que los capellanes utilicen, la experiencia del usuario no se verá afectada.
- Inmediatez de acceso: dado que se encuentran en un servidor, no hace falta descargar, instalar ni configurar. Está lista para ser usada al momento.
- Múltiples usuarios concurrentes: las aplicaciones web realmente pueden ser utilizadas por múltiples usuarios al mismo tiempo, lo cual es perfecto para un sistema como este.

Aún llegados a este punto, hace falta determinar el patrón de arquitectura de software

que se utilizará, el cual brindará un esquema más amplio de la organización estructural necesaria para el sistema.

Propósito de la propuesta

La presente investigación es una propuesta de un sistema para el departamento de capellanía de la Universidad de Morelia, utilizando el patrón de arquitectura de software Modelo-Vista-Controlador, cuya principal característica sea realizar encuestas a los alumnos, y agendar citas con los capellanes. Y de esta manera, agilizar y mejorar la organización en comparación al proceso efectuado en la actualidad.

La realización de encuestas y entrevistas regulares a los alumnos permite a los capellanes un conocimiento más detallado del perfil social, académico y espiritual del estudiante para tener mejor percepción acerca del comportamiento y actitudes del estudiante y mediante esa información ofrecer un servicio de consejería eficiente y acertado. Este proyecto permite al capellán reducir: el trabajo de aplicación manual de las encuestas, el volumen de expedientes, y la dificultad de realizar búsquedas y análisis de datos en papel. También, proporcionará una aplicación web intuitiva, capaz de administrar y coordinar toda la información obtenida mediante estos sondeos. Sin embargo, estas no son las únicas características con las que se plantea que cuente la aplicación. Pero, debido al limitante del tiempo, se dejarán otras funcionalidades como trabajo a futuro, y a diferentes desarrolladores. Esta es una ventaja del desarrollo mediante el patrón MVC debido a que se puede desarrollar el mantenimiento de las aplicaciones debido a la de arquitectura y organización de los componentes del sistema [9]. Una de las principales ventajas de MVC frente a otros patrones arquitectónicos es la clara separación de intereses, lo cual, además, es la forma en la que están creadas la mayoría de las aplicaciones y sitios web actuales [10].

Cada una de sus capas es responsable de un aspecto claramente definido de la aplicación. Esto permite que los elementos puedan ser flexibles y adaptables a diferentes situaciones. Permite diseñar, implementar y probar cada parte por separado y aun así mantener el código organizado. Es útil también para encontrar lo que se necesita rápidamente, probar características y agregar funcionalidades con facilidad. Todo esto resulta en

un código más eficiente y una forma efectiva de reutilizar código para aplicaciones más rápidas [11].

MVC despliega un ámbito que permite el razonamiento en la lógica de programación y la organización de los archivos del programa. Indica la idea de que el código debe estar organizado por su función. Brinda un punto de partida para interpretar las ideas en código y, así mismo, al identificar qué hace cada elemento, facilita su retorno a este y la colaboración de otros desarrolladores. Pensar en cómo interactúa un código con otro código es una parte importante de la programación, y aprender a colaborar con otros desarrolladores es una habilidad importante [12].

Esto permitirá de una manera más cómoda la implementación de nuevas funcionalidades para el sistema por parte de otros desarrolladores en el futuro.

Objetivos de la investigación

La presente investigación tiene la intención de definir la utilidad y aprovechar las bondades y recursos del patrón arquitectónico MVC en el sistema para el departamento de capellanía de la Universidad de Morelia, de manera que el código redactado sea fácilmente entendible por todos los desarrolladores involucrados en el proyecto, además de utilizar herramientas que puedan hacer el sistema escalable para nuevas funcionalidades y que la experiencia del usuario final sea favorable; que tenga el potencial de convertirse en una fuente rentable, viable y confiable de datos para el cuidado cercano de los estudiantes al permitir la obtención eficaz de la información relevante y oportuna.

Lo que se desea con este proyecto es posibilitar el mejoramiento de la calidad general de la atención al alumno por parte del departamento de capellanía; además de hacer el proceso de encuestas y entrevistas de los capellanes más eficaz.

El sistema almacenará información completa y pertinente de los estudiantes y permitirá a los pastores capellanes un acceso fácil y seguro, así como, y por efecto indirecto encuentros más productivos en el uso del tiempo de las entrevistas y mejor comunicación entre capellanes y alumnos.

En esencia, se desea apoyar a los capellanes mediante este plan a asistir a los estudiantes y que realicen una relación personal más cercana con

cada uno de ellos, con el fin de determinar las necesidades y proyectar el potencial del estudiante para ayudarlo en el desarrollo de su proyecto educativo personal, a partir de ayudar al alumno a descubrir la misión que Dios le ha dado, para diseñar el mejor camino de desarrollo completo. Así mismo ejerciendo la disciplina preventiva y correctiva de modo que el carácter del estudiante se desarrolle de modo armonioso y fiel a Dios [1, p. 56-57].

Limitaciones

□ La primera limitación es la disponibilidad de horas para el desarrollo del sistema debido a que el tiempo, siendo es un factor crítico de éxito en proyectos de ingeniería de software.

□ Por ahora, el sistema solamente tendrá como función principal el almacenamiento de datos útiles para capellanía y la información de las encuestas que sirven al departamento para realizar las entrevistas con los estudiantes. Otra función importante consiste en agendar entrevistas con los capellanes y que estos puedan organizarlas de manera más eficiente y sin pérdida de información.

□ Las pruebas necesarias para el desarrollo del sistema serán realizadas localmente, hasta que esté listo para migrar a un servidor de la universidad.

□ Debido al confinamiento ocasionado por la pandemia actual del COVID-19 [13], es posible que las tres partes en las que está dividida la investigación no se acoplen a la perfección, pues cada participante es responsable de una de estas secciones.

□ Una vez terminada esta fase del sistema, los capellanes y los alumnos son la clave para el éxito o fracaso del cambio en la metodología en el departamento de capellanía que se propone en este trabajo. Se espera que puedan adaptarse y darle un buen empleo al sistema, evidenciando así el beneficio que tiene utilizarlo. Así mismo, se espera que futuros desarrolladores puedan seguir actualizándolo y mejorándolo.

Delimitaciones

La presente investigación contará con los siguientes parámetros:

1. El estudio solamente abarcará el análisis de la utilización del patrón arquitectónico MVC en

el sistema, siendo esta una de las tres partes en las que está dividido el proyecto.

2. El sistema propuesto será una aplicación web, implementada en un web hosting con una base de datos relacional.

3. El sistema será exclusivamente para el departamento de capellanía de la Universidad de Montemorelos.

4. Los tipos de usuario se delimitarán a capellanes y a cualquier alumno perteneciente a la universidad que precise de los servicios de capellanía.

5. Cabe mencionar que la información requerida en las encuestas y entrevistas del sistema será solamente la proporcionada y solicitada por los capellanes, los cuales, previamente tuvieron una sesión para determinar la más importante.

6. El sistema no tendrá relación con otros sistemas ajenos o pertenecientes a la universidad, como la plataforma e42 o el sistema académico virtual.

7. El sistema es de uso exclusivo para proporcionar información de los alumnos y de interés para los capellanes y estos datos no estarán disponibles para otros fines.

Definición de términos

□ Reglas del Negocio. “Una regla de negocio es una condición que se debe satisfacer cuando se realiza una actividad de negocio. Una regla puede imponer una política de negocio, tomar una decisión o inferir nuevos datos de existentes” [14]. “Forma parte de la lógica de negocios, la cual es una capa que ocupa un lugar primordial en la construcción de una infraestructura de software que permitirá el crecimiento y la extensión de servicios para todas las aplicaciones existentes y futuras” [15].

□ Framework. “En los sistemas informáticos, un framework, o marco de trabajo, es a menudo una estructura en capas que indica qué tipo de programas pueden o deben ser construidos y cómo se interrelacionan. Algunos de ellos también incluyen programas reales, especifican interfaces de programación u ofrecen herramientas de programación para usar los marcos” [16].

□ Arquitectura de Software. “Describe los componentes básicos de un sistema de software y su combinación interna. En el marco de desarrollo de software, representa la decisión de diseño más

temprana. Es determinada básicamente por criterios de calidad como la modificabilidad, mantenibilidad, seguridad y el rendimiento” [17].

□ Aplicación Web. “Una aplicación web es un sitio web que contiene páginas con contenido sin determinar, parcialmente o en su totalidad. El contenido final de una página se determina sólo cuando el usuario solicita una página del servidor web. Dado que el contenido final de la página varía de una petición a otra en función de las acciones del visitante, este tipo de página se denomina página dinámica. Las aplicaciones web se crean en respuesta a diversas necesidades o problemas” [18].

Metodología

Este proyecto tiene como un fin resolver un problema relacionado con el departamento de capellanía, utilizando las tecnologías de la información.

La solución que se busca conduce a un producto de software que agende, organice y muestre la información proveniente de las entrevistas y encuestas realizadas regularmente a los alumnos por los representantes de este departamento a través de un navegador web.

En base a la experiencia y el conocimiento de los capellanes, determinaron información clave para llevar un seguimiento en la calidad de vida de sus alumnos en varios aspectos, pero sobre todo en el espiritual. Esta valiosa información fue lo primero que se tomó en cuenta para el almacenamiento de los datos, que combinado con las metodologías para el desarrollo de software, han permitido la creación de este sistema.

Análisis de requerimientos

Un requerimiento se puede definir como una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal [19]. Sin embargo, un requerimiento de software también puede ser definido como una condición o necesidad del software para que el usuario pueda resolver un problema o alcanzar un objetivo, en este caso la resolución de las cuestiones descritas anteriormente.

Los requerimientos se han agrupado desde tres perspectivas:

- Requerimientos del capellán
- Capturar datos del formato oficial de la entrevista inicial definida por todos los capellanes.
- Capacidad de agendar entrevistas.
- Capacidad de mostrar los datos de los alumnos obtenidos a través de la entrevista inicial
 - Requerimientos del sistema.
 - Capacidad de que el usuario pueda registrarse con una cuenta en el sistema.
 - Capacidad de que el usuario pueda iniciar sesión con su cuenta en el sistema.
 - Requerimientos arquitectónicos:
 - Seguir el patrón arquitectónico MVC, el cual implica separar el código por su función principal, ya sea mostrar una vista, hacer redirecciones o trabajar con los datos.
 - Creación de una base de datos adecuada para el sistema y la información requerida para la entrevista inicial.
 - Emplear una interfaz de usuario agradable e intuitiva para una mejor experiencia de usuario.

Diseño y definición de la arquitectura

Actualmente, no existe una definición única para el concepto de arquitectura de software, no obstante, se reconoce como la definición más completa la dada por la IEEE: “La arquitectura de software es la organización fundamental de un sistema enmarcada en sus componentes, las relaciones entre ellos, y el ambiente, y los principios que orientan su diseño y evolución” [20].

El concepto de arquitectura multicapa es un conjunto ordenado de subsistemas, cada uno de cuales está constituido en términos de los que tiene por debajo y proporciona la base de la implementación de aquellos que están por encima de él [21].

Se escogió la arquitectura multicapa con 3 capas iniciales pues está basada en el bajo costo de administración de clientes, alta accesibilidad, alta flexibilidad de crecimiento, alta disponibilidad y tolerancia a fallos, alta escalabilidad e

independencia de los repositorios y bases de datos. Esta arquitectura particiona todo el sistema en distintas unidades funcionales. Esto asegura una división clara de responsabilidades y hace que el sistema sea más mantenible y extensible. Los sistemas con tres o más capas se han probado como más escalables y flexibles que un sistema cliente-servidor, en el que no existe la capa central de lógica de negocios [22]. A continuación se explica detalladamente las capas de nuestra arquitectura:

Capa de datos: MySQL

Esta capa incluye tanto el servidor web como el servidor de base de datos. Para el servidor de base de datos se utilizó el motor MySQL Community Edition. Se eligió el estándar SQL ya que la información de este proyecto es altamente relacional.

El modelo inicial de la base de datos incluía una tabla para alumnos, una para capellanes, siete para almacenar la información de la entrevista inicial y cinco para agendar y almacenar una entrevista. Esto claramente fue cambiando a lo largo del desarrollo, empezando por la creación de dos tablas, una donde se almacenaran los eventos creados por los usuarios y los capellanes y otra para los eventos predefinidos solamente por los capellanes. Además se creó una tabla llamada usuarios, que incluyera a todos los usuarios registrados en el sistema, pero con un identificador del tipo de usuario para diferenciarlos. Otro cambio importante fue la adición de un campo para almacenar las rutas de las imágenes de perfil de los capellanes. Debido a que en MySQL no es posible guardar una imagen como tal, sino que tiene que ser convertida en otros tipos de datos lo cual puede llegar a ser muy fastidioso y, además, no es factible dado que afecta la eficiencia de la base de datos por ser un tipo de objeto muy grande [23], las imágenes son subidas y guardadas en una carpeta en el servidor, y en la base de datos simplemente se guarda la ruta a esa imagen.

Las otras modificaciones a la base de datos fueron menores, como la modificación de un tipo de dato por otro más adecuado o la adición de un identificador necesario. El modelo final incluye una tabla para todos los usuarios, una tabla para alumnos, una para capellanes, ocho para almacenar la información de la entrevista inicial, dos para

agendar las entrevistas con los capellanes, y cinco más utilizadas como tablas de catálogos.

El servidor web utilizado para el desarrollo de esta aplicación fue el servidor virtual de Apache proporcionado por WAMP/XAMP, esto hasta que esté listo para migrar a un servidor perteneciente a la Universidad de Montemorelos.

Capa de negocios: PHP

Se decidió utilizar el lenguaje PHP debido al provecho brindado para este proyecto. PHP es un lenguaje apropiado para el desarrollo de páginas web, que se ejecuta del lado del servidor. Además, puede ser incrustado dentro de HTML y realizar diversas funciones, de las cuales, las más destacables para este proyecto son el procesamiento de información en formularios y las páginas dinámicas.

Otra de las funciones es que cuenta con un amplio soporte para bases de datos, entre las cuales pueden mencionarse InterBase, mSQL, Oracle, Informix, PostgreSQL y por supuesto, MySQL, entre otras [24]. PHP es la opción natural para los programadores en máquinas con Linux que ejecutan servidores web con Apache, pero funciona igualmente bien en cualquier otra plataforma de UNIX o de Windows.

Para visualizar, modificar y eliminar los eventos creados, llenar dinámicamente los selects utilizados para la entrevista inicial y además poder guardar todas las secciones de la entrevista sin tener que recargar la página se utilizó una tecnología llamada AJAX. Asynchronous JavaScript and XML es un conjunto de técnicas de desarrollo web que permiten que las aplicaciones web funcionen de manera asíncrona, procesando cualquier solicitud al servidor en segundo plano [25]. En palabras más simples, AJAX permite acceder a datos existentes en el servidor sin necesidad de recargar la página.

Además de esto, se creó y utilizó un archivo htaccess para la utilización de las denominadas URLs semánticas o amigables. Un archivo htaccess es un fichero de configuración utilizada por servidores web basados en Apache. Este tipo de archivos configura los ajustes iniciales de un programa o un servidor. Esto significa que se puede utilizar para que el servidor se comporte de una forma determinada [26]. Por otro lado, las URLs amigables son aquellas direcciones que son

fáciles de entender por los usuarios gracias a su estructura y forma. Tratan siempre de arrojar información clave para que cualquiera pueda entender el contenido que albergan en la página a la que dirigen [27].

Capa de presentación: Bootstrap

Para un mejor aspecto de la aplicación, se decidió utilizar los siguientes componentes:

□ Bootstrap. Un conjunto de herramientas de código abierto para desarrollar con HTML, CSS y JS que facilita la maquetación de sitios web. Provee características imprescindibles en un proyecto como este. Además ofrece herramientas para que el sitio web se vea bien en toda clase de dispositivos, sin tener que rediseñar todo el sitio [28].

□ Google Fonts. El mayor catálogo de fuentes web gratuitas y de código abierto [29].

□ Plugins de Javascript como DataTables, FullCalendar y ClockPicker, ideales para mostrar y almacenar datos. A continuación se describe cada uno:

1. DataTables: es una herramienta altamente flexible para jQuery, construida sobre los cimientos de la mejora progresiva, que agrega múltiples características avanzadas a cualquier tabla HTML [30].

2. FullCalendar: es una herramienta ideal para mostrar eventos, pero no una solución completa. El agregar opciones como la modificación del nombre depende del programador [31].

3. ClockPicker: es un selector de hora estilo reloj para Bootstrap o jQuery [32].

Así mismo, para mejorar la lógica de la entrevista inicial se utilizó JavaScript para modificar la vista según las acciones del usuario.

Seguridad y autenticación

La seguridad es un tema muy importante en el área del desarrollo web debido a que muchas veces se maneja información sensible y crítica. Es por esto que, desde el punto de vista de un administrador de un sitio web, la seguridad de una aplicación web no puede quedar desatendida. Las decisiones concernientes a la seguridad deben ser tomadas durante todo el ciclo de vida del proyecto: desde la fase de diseño hasta la puesta en producción [33].

La seguridad del proyecto en el nivel de la capa de negocios y de datos, está basada en la autenticación por contraseña. Para ello, se utiliza la función de PHP `password_hash()`, la cual mediante un algoritmo encripta la contraseña al crear una nueva cadena de caracteres, llamada hash. Esta se almacena en la base de datos y se recupera mediante otra función de PHP llamada `password_verify()`, la cual comprueba que la contraseña coincida con el hash.

El hashing de contraseñas es una de las consideraciones de seguridad más básicas que deben realizarse al diseñar cualquier aplicación que acepte contraseñas de los usuarios. Sin el hashing, cualquier contraseña almacenada en la base de datos de la aplicación puede ser robada si la base de datos llegara a estar comprometida, y luego puede ser usada inmediatamente para comprometer no solo la aplicación, sino también las cuentas de los usuarios en otros servicios, si no usan contraseñas únicas. Al aplicar un algoritmo de hash a las contraseñas de sus usuarios antes de almacenarlas en su base de datos, hace imposible que cualquier atacante determine la contraseña original, al tiempo que puede comparar el hash resultante con la contraseña original en el futuro [34].

En la capa de presentación, también es necesario un método de seguridad. Es necesario determinar los sitios a los que un tipo de usuario puede acceder o no. Para ello, cada vista contiene archivos de verificación que, entre otras cosas, comprueban que el usuario haya iniciado sesión, que esté validado, que haya contestado los datos iniciales correspondientes, etcétera.

Conclusiones

En este proyecto se propuso desarrollar un modelo de software para el departamento de capellanía de la Universidad de Morelos, tomando en cuenta que una parte de vital importancia para esta división es la de relacionarse con sus alumnos, acercándose a ellos y detectando sus necesidades. Todo esto mediante encuestas y entrevistas que se realizan periódicamente.

Se considera que la importancia de este proyecto no puede ser medido sólo por los paradigmas de programación que fueron empleados para su desarrollo, sino el impacto que tendrá en los usuarios del sistema. En el mundo

actual, existe una infinidad de aplicaciones web. Sin embargo, ninguna en específico para un servicio como este. Los autores creen que es necesario reconocer la importancia del trabajo de los capellanes como guías y guardianes espirituales encomendados por Dios. Por ello, se tiene planeado a futuro, que este sistema no sea únicamente de la Universidad de Montemorelos, si no que en cada escuela, universidad o institución adventista cuente con con él y pueda llegar a ser una pequeña ayuda en el crecimiento de la vida espiritual para cualquier persona que lo utilice.

Referencias

- [1] Plan Institucional de Desarrollo, Universidad de Montemorelos, Montemorelos, NL, 2017, p.69
- [2] Manisha J., Ashok K., Manisha A., Isha S., “Behavior of MVC (Model View Controller) based Web Application developed in PHP and .NET framework” IEEE, Indore, India, Nov.2016.
- [3] X. Wang, B. Xu, R. Gu, “The Application of Code Reuse Technology Based on the MVC Framework” International Conference on Computer Sciences and Applications, pp.534-537, 2013.
- [4] Krasner, “A cookbook for using the model-view-controller user interface paradigm in smalltalk-80” J. Object Oriented Program., vol. 1, no. 3, Aug. 1988.
- [5] Jin P., Yao J., “Design and realization of college service center system based on MVC”, IEEE Ottawa, ON, Sept. 2014
- [6] Mohammad R., Indhi F., “Design an MVC Model using Python for Flask Framework Development”, IEEE, Indonesia, Indonesia, Sept. 2019.
- [7] L. Stevens (2008, Mayo 9). “Capellanía, el ministerio que en Interamérica va hacia adelante” [Online] Disponible :<https://www.interamerica.org/es/2008/05/capellania-ministerio-que-en-interamerica-va-hacia-adelante/>
- [8] Internet Ya (2016, Nov. 14) “Ventajas y beneficios de las aplicaciones Web” [Online]Disponible: <https://www.internetya.co/ventajas-y-beneficios-de-las-aplicaciones-web/>
- [9] Álvarez M. (2014, Ene. 2) “Qué es MVC” [Online]Disponible:<https://desarlloweb.com/articulos/que-es-mvc.html>
- [10] Sandoval M., (2014, Ago. 29) “Qué es MVC y por qué es tan usado en el desarrollo Web” [Online] Disponible: <https://blog.michelletorres.mx/mvc-y-su-importancia-en-la-web/>
- [11] Siya Infotech (2017, Ene. 16) “Top 6 Most Important Benefits of MVC Architecture forWeb Application Development Process” [Online] Disponible: <http://siyainfo.com/2017/01/16/top-6-important-benefits-mvc-architecture-web-application-development-process/>
- [12] Codecademy. “MVC: Model, View, Controller” [Online] Disponible: <https://www.codecademy.com/articles/mvc>
- [13] Organización Mundial de la Salud (2020) “Brote de enfermedad por coronavirus (COVID-19): orientaciones para el público” [Online] Disponible: <https://www.who.int/es/emergencias/diseases/novel-coronavirus-2019/advice-for-public>
- [14] IBM. “Reglas del negocio” [Online] Disponible: https://www.ibm.com/support/knowledgecenter/es/SSFTN5_8.5.6/com.ibm.wbpm.wid.bpel.doc/busrules/topics/cundbus.html
- [15] Junta de Andalucía. “Funcionalidad de la capa de negocio” [Online] Disponible: <http://www.juntadeandalucia.es/servicios/madeja/sites/default/files/historico/1.3.0/contenido-libro-pautas-12.html>
- [16] M. Rouse (Dic. 2016) “Framework” [Online] Disponible: <https://searchdatacenter.techtarget.com/es/definicion/Framework>
- [17] Voigtmann. “Arquitectura de software” [Online] Disponible: <https://www.voigtmann.de/es/desarrollo-de-software/arquitectura-de-software/>
- [18] Adobe. “Aspectos básicos de las aplicaciones web” [Online] Disponible: <https://helpx.adobe.com/es/dreamweaver/using/web-applications.html>