

Implementación de la arquitectura Modelo Vista Controlador con el framework Razor .Net Core en el sistema web de actividades complementarias

Flores-Luevanos, M.G.¹, Moreno-Núñez, E.¹, Rodríguez-Lozano, K.V.¹

Datos de Adscripción:

¹ Tecnológico Nacional de México, Campus Lerdo / Instituto Tecnológico Superior de Lerdo. Av. Tecnológico No. 1555 Sur. Periférico Gómez - Lerdo. Km. 14.5, Ciudad Lerdo, Estado de Durango. C.P. 35150. México.

*maría.fl@itslerdo.edu.mx

Resumen - La arquitectura Modelo-Vista-Controlador, es un patrón de diseño de software de fácil organización que se puede adaptar a diferentes frameworks. Este modelo permite escalar fácilmente las versiones de un software y facilita el trabajo en equipo. El Sistema Web para el control de Actividades Complementarias, utilizado en el Tecnológico Superior de Lerdo, ha tenido constantes cambios, lo que ha derivado en la publicación de diferentes versiones en corto tiempo. La arquitectura utilizada en sus inicios ha quedado obsoleta, por lo que ha sido necesaria su actualización. El objetivo de esta investigación aplicada es la actualización de la arquitectura y entorno de trabajo de ese sistema institucional, para la corrección y optimización de la ejecución de las funciones existentes y el cumplimiento de los requisitos administrativos del lineamiento de actividades complementarias del TecNM. Se documenta el proceso de implementación del Modelo Vista Controlador con el Framework ASP .NET 6 Razor Core, en el Sistema Web de Actividades Complementarias. Se determinaron marcos de trabajo para la gestión ágil del proceso de desarrollo del proyecto. Como resultado del proyecto, el Sistema Web SAC cumplirá con los objetivos y requerimientos establecidos, facilitando el uso del mismo para los jefes de carrera, docentes y estudiantes.

Palabras Clave - *Actividades complementarias, Arquitectura, MVC, Razor,*

Abstract - The Model-View-Controller architecture is an easily organized software design pattern that can be adapted to different frameworks. This model allows easy scalability of software versions and facilitates teamwork. The Web System for the control of Complementary Activities, used at Tecnológico Superior de Lerdo, has undergone constant changes, which has resulted in the publication of different versions in a short time. This fact has generated that the architecture used for its development has become obsolete, so it has been necessary to update it. The objective of this applied research is to update the architecture and framework of this institutional system, for the correction, optimization and execution of the existing functions and the fulfillment of the administrative requirements of the TecNM's complementary activities guidelines. The implementation process of the Controller View Model with ASP Framework .Net 6 Razor Core, in the SAC System (Complementary Activities System) is documented. Frameworks for the agile management of the project development process were

determined. As a result of the project, the SAC Web System will comply with the established objectives and requirements, facilitating its use for career managers, teachers and students.

Keywords - *Architecture, Complementary Activities, MVC, Razor,*

I. INTRODUCCIÓN

En el Instituto Tecnológico Superior de Lerdo se implementan los lineamientos del Tecnológico Nacional de México, entre los que figura el de la acreditación de actividades complementarias; que especifica que los alumnos deben cumplir con una formación integral y acreditar las siguientes actividades: tutoriales, culturales, deportivas, de lectura e investigación. Debido a la creciente cantidad de población estudiantil en el tecnológico, se desarrolló un sistema web en el año 2015, el SAC (Sistema de Actividades Complementarias) para llevar el control de las actividades, generar reportes e imprimir la carta de liberación de cada actividad. El sistema mencionado ha tenido cambios importantes debido a nuevos requerimientos o a la mejora de los procesos, que se traducen en nuevas funcionalidades. Sin embargo, a través del tiempo la tecnología ha evolucionado a pasos agigantados, por ello se tuvo la necesidad de crear una nueva versión con nuevas tecnologías y mejorando notablemente la interfaz gráfica. Para implementar adecuadamente los cambios pertinentes se tuvo que realizar una gestión adecuada, reajustar el sistema para ubicar las modificaciones en los procesos de trabajo, integrar nuevas funcionalidades y mejorar ya las existentes, actualizar el código y solucionar incidencias detectadas. En el proceso de desarrollo se utilizó la metodología agile o Agile Management, que permitió adecuar el proyecto al equipo de trabajo y a los tiempos demandados, y su desarrollo en las condiciones específicas del entorno. La tecnología seleccionada para el proyecto es la arquitectura MVC Razor Pages en ASP.NET Core, que facilita la programación de escenarios centrados en páginas y hace más productivos los controladores y las vistas. (Anderson, 2023). En la sección de resultados del artículo se muestran los productos del diseño y análisis, se incluyen los principales diagramas UML, el mapa de navegación, la arquitectura MVC y las principales interfaces del sistema web desarrollado.

II. PARTE TÉCNICA DEL ARTÍCULO

A. Metodología

Los métodos ágiles son un método de trabajo dirigido a resultados que extiende el proceso de desarrollo, integrando nuevas funcionalidades y actualizaciones de forma continua, asegurando el lanzamiento en plazos razonables. Las metodologías ágiles buscan adaptar pequeños cambios a las rutinas de trabajo teniendo como punto esencial las personas y el trabajo en equipo orientado a resultados. Existen diversas metodologías ágiles que se pueden implementar, cada una de estas se enfoca en ciertos aspectos del desarrollo de software que la hace ideal para cubrir necesidades específicas de un proyecto.

Para la realización de este proyecto, fue necesario establecer una metodología que permitiera la implementación de un entorno de trabajo colaborativo, definiendo roles y tareas con el objetivo de tener avances constantes en el desarrollo del sistema. La metodología ágil utilizada es la SCRUM, la cual se define como un marco de trabajo leve que permite definir roles, eventos y reglas para lograr el éxito en el desarrollo de un proyecto.

El proyecto gestiona los procesos para poder seguir la metodología SCRUM en el desarrollo de los productos de software. Para ello se creará una aplicación web orientada a tecnologías .NET que facilita la programación de los escenarios, incrementando la productividad de los controladores y las vistas.

B. Arquitectura Modelo-Vista-Controlador

El Modelo-Vista-Controlador es un patrón arquitectónico de tres capas, para la construcción de software. Se enfoca al desarrollo de software en partes, con los objetivos de facilitar la implementación, actualización y mantenimiento independientemente de sus elementos y reducir el esfuerzo requerido en la fase de codificación, entre otros. (Ávila, 2019).

En esta arquitectura, el modelo (representado por su letra inicial M en el acrónimo) se encarga de controlar los datos fuente de la aplicación, por medio de la definición de reglas de negocio y notifica a las vistas los cambios provocados por un agente externo; la vista (V) es el conjunto de ítems visibles a los usuarios por medio de las interfaces, recibe los datos de su controlador asociado y tiene un registro de éste; el controlador (C) recibe los eventos de entrada, contiene las reglas de gestión de las peticiones al modelo o a la vista y manipula los datos del modelo para mostrarlos a los usuarios.

Las principales ventajas de la implementación de frameworks con el patrón MVC son: la separación física y lógica de los componentes de las aplicaciones, la optimización de la actualización y del mantenimiento del software. (Díaz y Fernández, 2012).

El marco de ASP.NET MVC posee generadores de los archivos base, en un entorno de poca complejidad y fácil de probar.

ASP.NET Core MVC representa, desde el 2015, año de su presentación, un cambio radical para los desarrolladores web que utilizan la plataforma de Microsoft. Este framework enfatiza la arquitectura limpia, los patrones de diseño, la capacidad de

prueba; y no oculta su funcionamiento en entornos web. (Freeman, 2022).

ASP.NET Core es un marco de trabajo completamente nuevo, al estar basado en .NET Core, es compatible con aplicaciones web en una gama de plataformas, provee la funcionalidad de ASP.NET MVC construido en la nueva plataforma ASP.NET Core.

ASP.NET Core MVC sigue la arquitectura modelo-vista-controlador, para conformar las aplicaciones web ASP.NET y las interrelaciones de sus elementos.

El patrón MVC data de 1978, pero ha ganado popularidad al día de hoy como un patrón para aplicaciones web por dos razones principales: 1) la interacción del usuario en una aplicación con esta arquitectura es lógica, cuando el usuario demanda una acción, la aplicación modifica su modelo de datos para responder con una vista actualizada; este ciclo se ajusta convenientemente a las aplicaciones web, con una secuencia de peticiones y respuestas en http; 2) las app web combinan tecnologías diferentes, como lo son bases de datos, HTML y código ejecutable, que usualmente se dividen en capas. Estas combinaciones de patrones se mapean de forma natural en los conceptos del patrón MVC, así, ASP.NET Core MVC implementa una variante del patrón MVC especialmente adecuada para aplicaciones web. (Freeman, 2022)

C. Razor .NET Core

Razor es un framework de desarrollo web para .NET que se utiliza para crear aplicaciones web modernas y escalables.

En una app ASP.NET Core MVC, un componente llamado motor de vista (*view engine*) se usa para producir el contenido enviado a los clientes. El motor de vista default es el Razor, responsable de incorporar los datos en los documentos HTML. La habilidad de generar contenido dinámicamente es esencial para poder escribir apps web. Razor provee características que facilitan el trabajo, usando declaraciones C# (Freeman y Sanderson, 2012)

Las expresiones Razor son agregadas al código HTML estático en los archivos de las vistas, estas expresiones son evaluadas para generar las respuestas a las solicitudes de los clientes. (Freeman, 2022)

Algunas de las características ventajosas de usar páginas Razor es que se facilita la implementación de los patrones que comúnmente se utilizan con exploradores web, permite la reutilización de plantillas; es intuitiva para los programadores en el entorno .NET, tiene una sintaxis comprensible en la creación de las vistas y brinda un alto nivel de protección contra código malicioso; las validaciones se declaran en la clase del modelo y son aplicables en todo el sistema web; entre sus desventajas principales están las limitaciones en su integración con herramientas de otros sistemas operativos distintos a Windows u otras plataformas; otra es que debido a su alto nivel de personalización, los sistemas web pueden ser altamente complejos y grandes en cuanto a tamaño, lo que se podría reflejar en la afectación del tiempo de carga de las páginas web y el rendimiento general de la aplicación. (Anderson, Brock y Larkin, 2023)

D. SQL Server 2019

SQL Server es el sistema gestor para bases de datos relacionales, seleccionado para el desarrollo de esta nueva versión del sistema web de AC. SQL Server, en su versión 2019 (15.0) incluye mejoras a las anteriores versiones que repercuten

en el rendimiento de las cargas de trabajo de las BD, sin necesidad de modificar su diseño o el de las aplicaciones. Específicamente, el procesamiento inteligente de consultas, que permite la ejecución de cargas de trabajo paralelas y la compatibilidad con dispositivos de memoria persistente. SQL Server 2019 utiliza conjuntos de macrodatos de SQL Server y proporciona funciones y mejoras adicionales para el motor de base de datos de SQL Server, SQL Server Analysis Services, SQL Server Machine Learning Services, SQL Server en Linux y SQL Server Master Data Services. (Microsoft Learn, 2023)

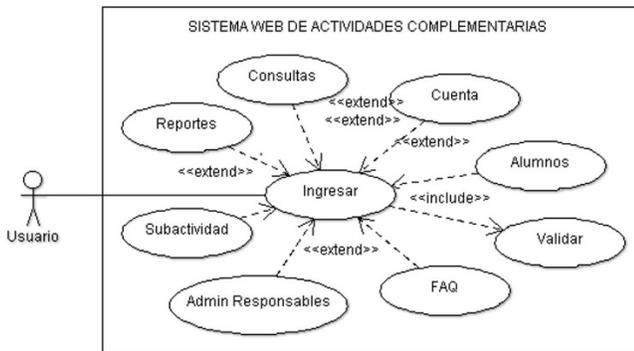
III. RESULTADOS Y DISCUSIÓN

3.1 Alcance

Para determinar y priorizar los requisitos para la nueva versión del Sistema web de Actividades Complementarias, fue necesario elaborar los modelos de casos de uso y de clases, para lo que se utilizó la herramienta de software orientada al lenguaje unificado de modelado ArgoUML®. A continuación, en las figuras 1 a la 3, se presentan los diagramas de casos de uso de las funciones sustantivas del sistema.

Figura 1

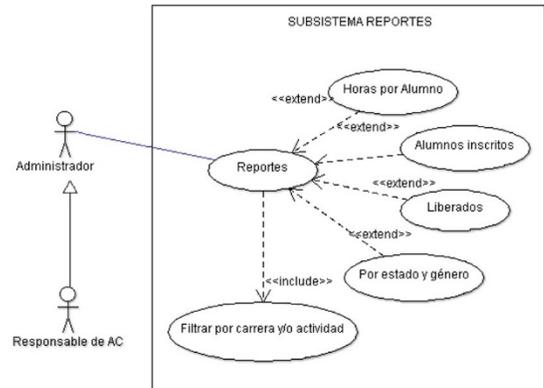
Diagrama general de casos de uso del sistema web



En la figura 1, se aprecian las funcionalidades del SAC: el usuario, que es cualquier miembro de la comunidad escolar del TecNM campus Lerdo, podrá ingresar al sistema web, mediante la validación de su nombre de usuario y una contraseña particular; a continuación, dependiendo del rol que se identifique mediante sus credenciales de acceso, podría tener las facultades de configurar su cuenta, efectuar consultas, gestionar a los alumnos, a los responsables o a las subactividades, generar reportes y consultar la sección de preguntas frecuentes.

Figura 2

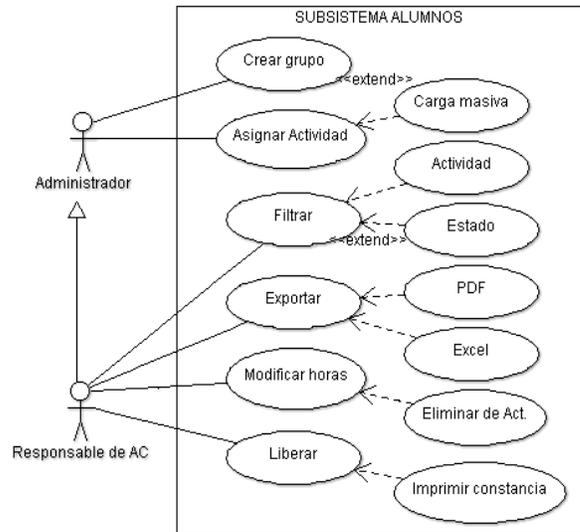
Diagrama de casos de uso del subsistema de reportes



En la figura 2 se muestra la diversidad de opciones para generar los reportes del sistema, en cualquiera que se ejecute, se deberá filtrar a los estudiantes por la carrera en que están inscritos o bien por actividad complementaria; las opciones de reporte son: el listado de los estudiantes inscritos en cualquier actividad complementaria; el número de horas que tiene registradas el estudiante en la actividad; la relación de alumnos que ya acreditaron la actividad, por lo que ya fueron liberados y un reporte especial que cuenta a los estudiantes de cada género y por estado del alumno en la actividad: "en curso", "liberado" o "completado".

Figura 3

Diagrama general de casos de uso Alumnos

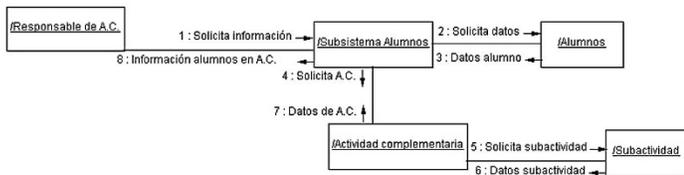


La figura 3 representa a los servicios del subsistema Alumnos, que están disponibles para el administrador del sistema o el profesor responsable de cualquier actividad complementaria. Estos actores tienen la opción de crear un grupo de estudiantes y asignarle una actividad complementaria a ese grupo o bien a un alumno. Es posible cargar los datos de los alumnos que conforman un grupo en forma masiva, obteniendo los datos de un archivo en formato de Excel. También es posible filtrar la información de los alumnos por actividad complementaria o por

cualquiera de los estados descritos anteriormente, exportar esos datos en un archivo con formato PDF o Excel, modificar las horas del estudiante o del grupo, eliminar su suscripción a una actividad, liberarlos cuando ya completaron sus horas e imprimir su constancia de cumplimiento.

En la figura 4, se muestra el diagrama de colaboración entre las clases del sistema, que contempla las acciones desencadenadas cuando el subsistema de alumnos recibe una solicitud de tratamiento de información emitida por el responsable de alguna actividad complementaria (actor) y están presentes todas las condiciones para que la consulta tenga buen término.

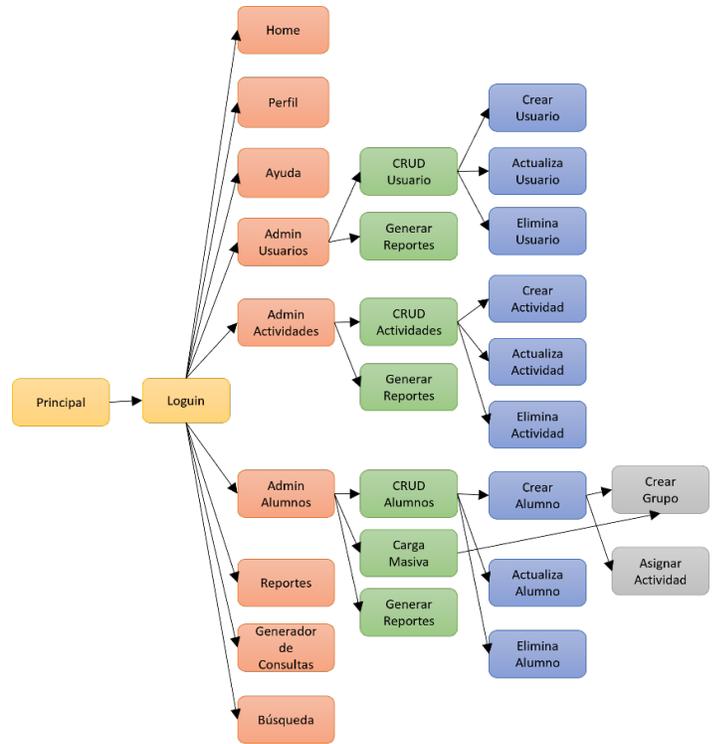
Figura 4
Diagrama de colaboración de tratamiento de información de los alumnos en actividad complementaria



3.2 Construcción

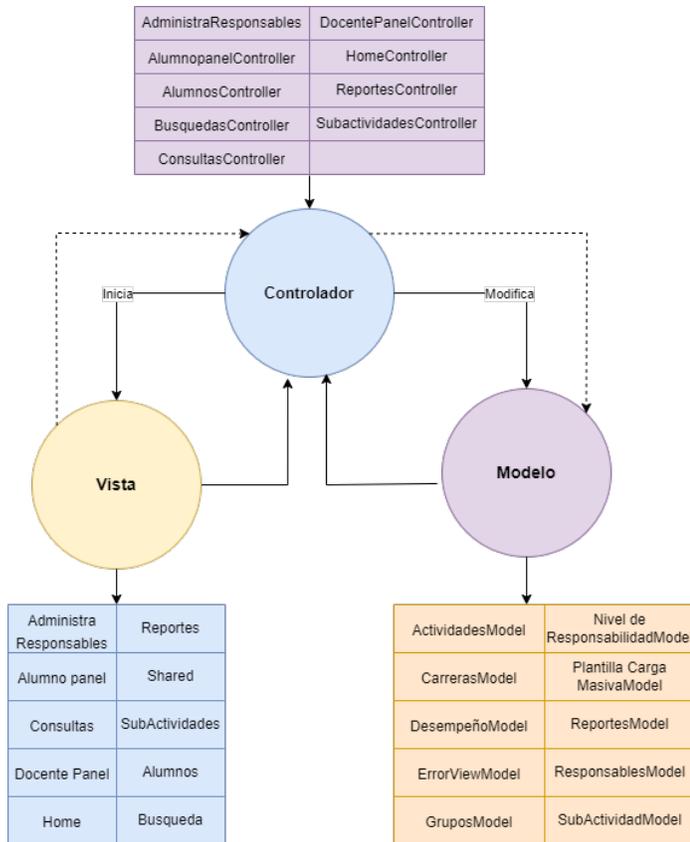
El mapa de navegación web, también conocido como mapa del sitio o *sitemap*, es la representación visual de todas las páginas que conforman el sitio web, proporcionando una visión general de la forma en que está organizado un sitio, lo que facilita la navegación intuitiva al usuario. La figura 5, muestra la estructura jerárquica de las páginas del sitio web (SAC) y la relación entre ellas. Dentro de algunas figuras en el diagrama se escriben las siglas CRUD, que significa "Create-Crear, Read-Leer, Update-Actualizar y Delete-Borrar".

Figura 5
Mapa de navegación sitio web



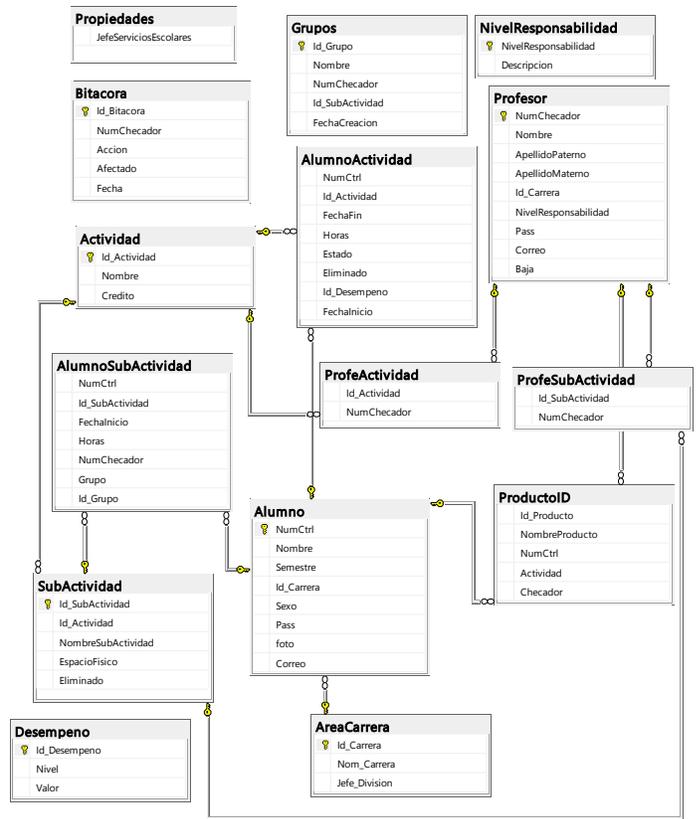
En la Figura 6, se muestra la arquitectura MVC del Sistema de Actividades Complementarias, esta arquitectura permite una mayor modularidad y reutilización de código. También facilita el mantenimiento y la evolución del software, ya que los cambios en una capa no afectan directamente a las otras capas. La imagen describe cómo desde el controlador se extraen, establecen y modifican los datos, que son actualizados en las vistas. El sistema web migró desde Aspx a Asp.Net, con Framework 6 y arquitectura MVC.

Figura 6
Arquitectura MVC



El modelo relacional ha sido ampliamente adoptado en la gestión de bases de datos debido a su simplicidad, flexibilidad y eficiencia, proporcionando una estructura organizada para almacenar y gestionar datos relacionales. El diagrama relacional utilizado en el sistema SAC, está compuesto por 16 tablas, que permiten relacionar al alumno con las actividades complementarias que está realizando dentro de un grupo o de manera individual y supervisado por un profesor. Esta base de datos también cuenta con procedimientos almacenados para ejecutar sentencias que agilicen la interacción con los datos del lado del servidor; es por este motivo, que en la nueva versión con MVC, no se incluyó el uso de *EntityFramework ni Linq* y se optó por crear objetos que sirven como intermediarios entre la base de datos y los modelos del sistema.

Figura 7
Modelo Relacional



3.3 Producción

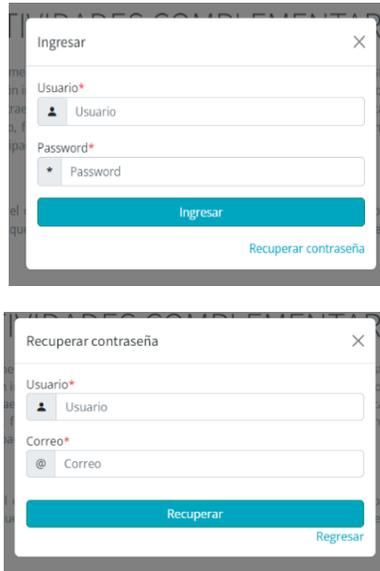
La figura 8 muestra la pantalla de inicio del sistema, aquí se pueden encontrar las opciones “Acerca de AC”, “Actividades” y “Contacto”, así como el botón para ingresar al sistema. La opción “Acerca de AC”, muestra el objetivo de las actividades complementarias, la opción de Actividades proporciona al usuario cuales son las actividades y una breve descripción de ellas, en la sección de contacto se despliegan los nombres y correos electrónicos de las personas encargadas de las actividades.

Figura 8
Interfaz principal de acceso al sistema web SAC



La figura 9 muestra la interfaz gráfica de login, en la que el usuario ingresa su usuario y contraseña. En caso de haber sido olvidada por el usuario, podrá recuperarla mediante su cuenta de correo electrónico.

Figura 9
Interfaz de acceso al sistema y Recuperación de contraseña.



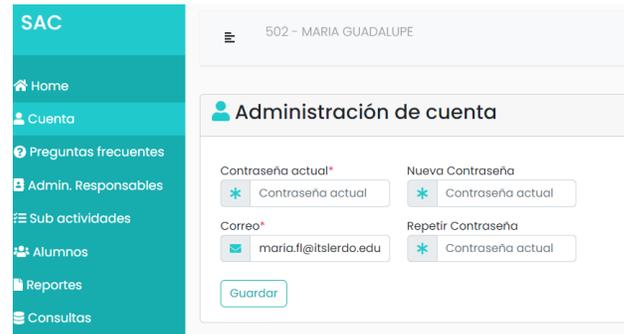
En la figura 10 se muestra el menú principal, desplegado a la izquierda de la pantalla: las opciones que tiene el usuario son: Home, que regresa a la pantalla inicial; Cuenta, para modificación de la contraseña o el correo electrónico registrados; la sección de Preguntas Frecuentes; Administración de responsables; Alumnos, para inscribir a los estudiantes a las subactividades, modificar sus horas de trabajo y acreditarles la actividad; Reportes, para la creación de diversos reportes del sistema; Consultas y Búsqueda, que permiten buscar a los estudiantes por número de matrícula o por actividad complementaria.

Figura 10
Interfaz del menú de opciones



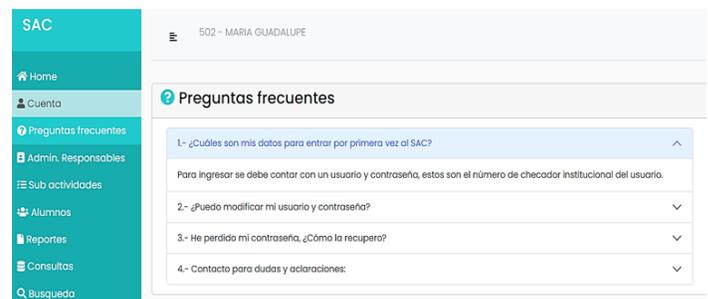
En la figura 11 se aprecian las opciones para la administración de la cuenta, que son básicamente la modificación de la contraseña o el correo electrónico registrados por el usuario. Cabe mencionar que el email registrado es el medio para la recuperación de la contraseña, en caso de olvido.

Figura 11
Interfaz de administración de cuenta.



Se incluyó una sección de FAQ (*Frequently Asked Questions*), como se muestra en la figura 12, en respuesta a la experiencia con las anteriores versiones del sistema; se le nombró "Preguntas Frecuentes" y se presentan en forma de una lista desplegable de preguntas y sus respuestas; también se le recuerda al usuario que en cada encabezado de las diferentes secciones aparece el icono de la ayuda (un signo cerrado de interrogación (?)) para que la ayuda hipertextual correspondiente se despliegue, en forma de cuadros de texto con indicaciones específicas acerca de la tarea en la que se encuentra el usuario.

Figura 12
Interfaz de preguntas frecuentes



En la figura 13, muestra la administración de responsables, en el sistema se puede asignar cualquiera de cinco tipos de roles a los usuarios permitiendo la seguridad, la eficiencia y adaptabilidad a diferentes contextos dependiendo del actor; a cada usuario se le asignan y permiten funcionalidades y características específicas. Los roles que maneja el SAC son: Administrador, Institucional, Académico, Jefe de Carrera y Servicios Escolares. Los Administradores tienen control total del sistema, el Institucional es el encargado de cada actividad complementaria, el académico es quien realiza la actividad con los alumnos, el jefe de carrera puede ver y consultar diferentes tipos de reportes. El rol de Servicios Escolares fue agregado para

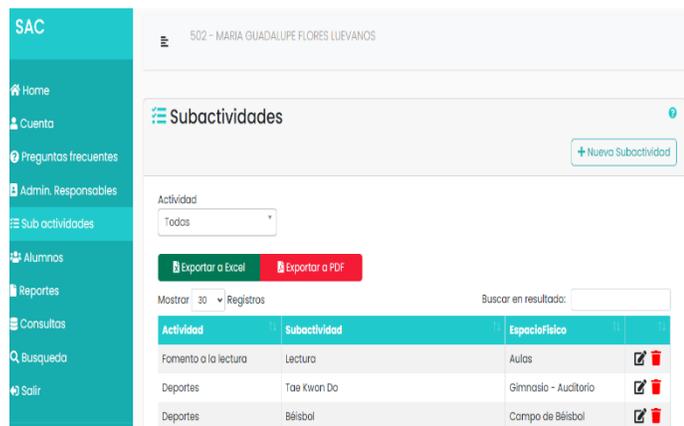
que este departamento pueda consultar o generar las cartas de término y/o liberación de las actividades complementarias.

Figura 13
Interfaz de Administración de responsables



La figura 14 corresponde a la sección de Subactividades, mediante la cual el profesor responsable de cada una de éstas puede registrar una nueva subactividad, modificarla o eliminarla. Cabe mencionar que las subactividades son las diversas acciones que se asignan a los estudiantes para que las realicen y de esta manera acumulen sus horas de trabajo, hasta completar las necesarias para que se les acredite una actividad complementaria.

Figura 14
Interfaz de Asignación de Subactividades



IV. CONCLUSIONES

Se concluye que la utilización del Framework Razor en el desarrollo de aplicaciones web dinámicas, aportó al sistema web una sintaxis fácil de leer y escribir, mezclando de forma fluida el código del lado del servidor (Backend) con la estructura de las páginas componentes. Razor, que es uno de los motores de vistas compatibles en ASP.NET Core, se utilizó en la generación del contenido HTML que se envía al navegador y permitió a los desarrolladores tener un mayor control y flexibilidad en la generación del contenido de las páginas que componen el sistema web.

La implementación de la arquitectura MVC (Modelo-Vista-Controlador), fomenta la escalabilidad del sistema web al permitir agregar nuevas vistas o modelos sin afectar el funcionamiento de los existentes.

La aplicación de código para serializar y deserializar a objetos .NET es de gran utilidad para la comunicación cliente-servidor y la visualización de los datos en las vistas. En el sistema web, las consultas se realizan en la base de datos y el resultado se regresa serializado en un objeto; en la vista éste se deserializa y los datos se presentan al usuario en la estructura requerida.

Las ventajas obtenidas al migrar el SAC de la plataforma ASP4x a una ASP.NET Core es el uso de las páginas Razor en la creación de las interfaces de usuario web, se aprecia un mayor rendimiento del sistema en el entorno de ejecución .NET CORE que en su versión anterior en ASP.NET 4.x

V. AGRADECIMIENTOS

Las autoras agradecen en primera instancia al titular de la Subdirección de Investigación y Desarrollo Tecnológico del ITSL, Ing. Jesús Alejandro Valdés Nieblas, por su apoyo en este proyecto. El agradecimiento se extiende a los estudiantes que a lo largo de los años han colaborado con su dedicación y esfuerzo en la mejora del sistema.

VI. REFERENCIAS

Anderson, R; Brock, D & Larkin, K (2023). *Introducción a Razor Pages en ASP.NET Core*. Microsoft Learn. <https://learn.microsoft.com/es-es/aspnet/core/razor-pages/?view=aspnetcore-7.0&tabs=visual-studio-code#razor-pages>

Ávila Garzón, C. (2019). Modelo Vista Controlador. Repositorio Digital Konrad Lorenz Fundación Universitaria. <https://repositorio.konradlorenz.edu.co/handle/001/1528>

Brind, M..(2022) ASP.NET Core Razor Pages in Action. Manning

Díaz González, Y., & Fernández Romero, Y. (2012). Patrón Modelo-Vista-Controlador. Telemática, 11(1), 47–57. <https://revistatelematica.cujae.edu.cu/tele/article/view/15>

Freeman, A., & Sanderson, S. (2013). Pro Asp. net Mvc 4. Apress

Freeman, A. (2022) Pro ASP.NET Core 6: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages 9th ed. Apress.

Lasa Gómez, C., Álvarez García, A., De las Heras del Dedo, R. (2017). Métodos Ágiles. Scrum, Kanban, Lean. Comercial Grupo Anaya, S.A.

Microsoft Learn (2023) Novedades de SQL Server 2019 (15.x). <https://learn.microsoft.com/es-es/sql/sql-server/what-s-new-in-sql-server-2019?view=sql-server-ver15>

Schwaber, K., y Sutherland, J. (2013). La Guía Definitiva de Scrum: Las Reglas del Juego.

Sutherland, J., y Sutherland, J. J. (2015). Scrum: El revolucionario método para trabajar el doble en la mitad de tiempo. Grupo Planeta Spain.