

Prototipo web como herramienta para la gestión de conocimiento y divulgación de resultados de investigación en CEGAFE

Claudia Lorena Quintero Gutiérrez

Área de teleinformática, SENA - CEGAFE, clquintero79@misena.edu.co

Manuel Enrique Saavedra Melo

Área de teleinformática, SENA - CEGAFE, manuelsaavedra@misena.edu.co

Resumen

Desde el año 2014 en el centro de gestión administrativa y fortalecimiento empresarial del SENA se han venido desarrollando proyectos de investigación en las diferentes áreas de formación, sin embargo, los productos generados en el proceso no han tenido la divulgación que tan importante actividad refiere, esto debido a que el centro no dispone de medios para realizar esta labor. Por otra parte, existen documentos formales que soportan cada proyecto de investigación, los cuales requieren un repositorio central para garantizar la disponibilidad de los mismos y ser gestionados por la líder Sennova del centro, en la actualidad este proceso se realiza en su mayoría a través de correo electrónico y eventualmente con almacenamiento en la nube.

Desde sus inicios el proceso de investigación ha contado con recursos tanto Sennova como de centro para la ejecución de los proyectos de investigación, esto ha permitido disponer de los equipos necesarios para el logro de los objetivos, y en este caso puntual, para el desarrollo de un prototipo de plataforma web que permita la gestión de conocimiento y divulgación de resultados de investigación. Cabe agregar que en el centro se forman aprendices con competencias relacionadas con el desarrollo de software y diseño web, de tal suerte que disponemos de recurso humano capacitado para ejecutar las tareas técnicas que fueron necesarias para construir la plataforma web.

El objetivo general que se planteó fue desarrollar un prototipo de una plataforma tecnológica para la gestión del conocimiento y divulgación de resultados de investigación e Innovación en el SENA – CEGAFE, gracias a la consecución del proyecto se obtuvieron como productos: prototipo de software, manuales de funcionamiento de la plataforma, del modelo conceptual y de clases y un *working paper* de la estructura del modelo físico de la base de datos.

Teniendo en cuenta que la gestión del proceso de investigación en el centro está en cabeza de la líder Sennova, y considerando que la entrevista es el instrumento más apropiado para la recolección de datos en este caso, se seleccionó el enfoque cualitativo como el más apropiado para el desarrollo del proyecto. Para la codificación de la plataforma web se prefirió Scrum como marco de trabajo, esto gracias a que existe experiencia previa y miembros del equipo capacitados.

Palabras clave

Plataforma web, gestión de investigación, gestión de conocimiento, divulgación.

Introducción

En el transcurso de los 3 últimos años en el centro de gestión administrativa y fortalecimiento empresarial se han propuesto desarrollar proyectos de investigación en las diferentes áreas temáticas que maneja el centro, en su formación titulada y complementaria. Como resultado del proceso se han originado diferentes productos de investigación entre los que podemos mencionar: artículos, estados de arte, software, ponencias, manuales, cartillas, *working paper*. Sin embargo, la gestión de los productos y del proceso en sí se ha realizado con herramientas ofimáticas, correo electrónico y eventualmente con almacenamiento en la nube, esto debido a que en el centro no se cuenta con una herramienta tecnológica que permita gestionar estos contenidos y centralizar la información.

Teniendo en cuenta la necesidad de disponer de una herramienta tecnológica que apoye la gestión del proceso de investigación para la divulgación de resultados y el almacenamiento de los archivos digitales propios del proceso, y con el ánimo dar solución a esta situación problemática se propone la construcción de una plataforma tecnológica para la gestión del conocimiento y divulgación de resultados de investigación e innovación en el centro de gestión administrativa y fortalecimiento empresarial.

Actualmente los espacios para dar a conocer los resultados de investigación son los eventos de transferencia al sector productivo, eventos promovidos por entidades externas o sencillamente voz a voz. Sin embargo, en estos escenarios el tiempo es limitado y depende de factores complejos (logística, tiempo, recursos) que no permiten la divulgación inmediata de los productos desarrollados, y más aún, en algunos casos no dependen de la gestión directa por parte del SENA.

Teniendo en cuenta que gran parte de los productos se evidencian en documentos que pueden ser manejados en medios digitales, la forma idónea de darlos a conocer sería un sistema propietario que permita realizar publicaciones web ajustadas a las condiciones del proceso investigativo; sin embargo actualmente no se cuenta con herramientas eficientes para tal fin ni con la vinculación a una plataforma especializada en este tipo de contenidos para que todos los actores del proceso puedan conocer de primera mano los resultados de los proyectos, es por esto que surge la pregunta de investigación:

¿En el SENA se cuenta con el talento humano, la capacidad y el conocimiento para construir una plataforma tecnológica que permita dar a conocer los proyectos de investigación?

Referentes teóricos

MVC 5 (Modelo, Vista, Controlador)

Es un patrón de diseño de software que divide el proyecto en tres componentes principales: modelo del dominio, presentación y las acciones basadas entre el usuario, en tres clases separadas.

- Modelo: Administra el proceso y los datos de la aplicación.
- Vista: Administra la visualización de la información.
- Controlador: Analiza las entradas de teclado y mouse del usuario, anunciando al modelo y/o vista de posibles cambios.

ASP.NET

Modelo de desarrollo web con servicios necesarios para la creación de aplicaciones empresariales. Forma parte de .NET framework; este lenguaje es compatible con el Common Language Runtime (CLR), entre los cuales se encuentran: Microsoft Visual Basic, C#, JScript.Net y J#.

AJAX (Asynchronous JavaScript And XML)

JavaScript asíncrono y XML, no es sí un lenguaje de programación, es una técnica de desarrollo web para la realización de aplicaciones interactivas y de manejo en tiempo real; es una tecnología asíncrona, en la que básicamente se solicitan los datos al servidor y estos se cargan en segundo plano sin intervenir con la visualización y el comportamiento de la aplicación.

Razor

Suministra una sintaxis de programación simple donde se incrusta en los formatos HTML de las páginas web; Crea contenido dinámico y puede generar código HTML. Cuando se crean archivos que incluyan contenido Razor tendrán una extensión de archivos especiales (cshtml o .vbhtml).

Visual Studio 2015

Es un entorno de desarrollo integrado (IDE), multiplataforma el cual sostiene varios lenguajes de programación como: C++, C#, entre otros. Esta edición de Visual Studio permite trabajar en diferentes versiones de frameworks .NET como: NET Framework 4.5.1, .NET Framework 4.5, .NET Framework 4.6, entre otros; los cuales facilitan el desarrollo de aplicaciones ASP tipo web.

SQL Server

Es un sistema-gestor de base de datos, desarrollado por Microsoft y basado principalmente en un modelo relacional o de organización (RM). Su principal función es almacenar y recuperar datos conforme a la solicitud que se realice.

Team Explorer

Es una aplicación que se vincula con otra (en este caso Visual Studio), para realizar diferentes funciones como: administrar código fuente, elementos de trabajo y la ejecución del proyecto.

Git

Es un software que facilita el control y manejo de las distintas versiones que se realizan de aplicativos, al momento de hacer cambios en el código fuente.

SCRUM

Es un proceso ligero y manejable para gestionar el desarrollo de aplicaciones, su objetivo fundamental, es optimizar los tiempos de desarrollo de las actividades de un proyecto en equipo, además realizar la descripción de los componentes, roles y organización adecuada en las actividades que se llevan a cabo a diario, se basa en tres pilares: transparencia, inspección y adaptación.

Htmlhelper

Es un método que permite presentar en una vista contenido HTML, generalmente devuelve una cadena de tipo texto. Estas cadenas pueden representar cualquier tipo de etiquetas generales HTML como por ejemplo `<img>` o `<button>`.

Gestión de investigación en la web

Enfocada principalmente en el apoyo a la investigación, que realizan los diferentes grupos y/o semilleros de manera eficaz, mejorando en las distintas actividades que deben realizar, facilitando el manejo de información entre los diferentes usuarios gracias al uso de las plataformas web.

Estrategias para mejorar la difusión de resultados de investigación con la Web 2.0

“La comunicación científica se está viendo enriquecida por la utilización de nuevos modos de almacenamiento, publicación y difusión de los resultados” (Página 1), por ejemplo, tendríamos los servicios de la web 2.0 que nos brindan aplicaciones donde las publicaciones científicas puedan ser públicas para toda persona que esté interesada en el tema, pero la idea de este documento no es solo hablar de dichas aplicaciones, sino de explicar que papel desempeñaría la aplicación respecto a la divulgación de los resultados de la, investigación, que influencia tendrá en el público y si es una ventaja o una desventaja utilizar esta la aplicación para anunciar investigaciones (Torres & Delgado, 2009).

Metodología

Se realizaron entrevistas con la líder Sennova del centro de gestión administrativa y fortalecimiento empresarial en la ciudad de Tunja. Teniendo en cuenta que la plataforma desarrollada es una solución a medida no fue necesario incluir más personas para la recolección de información, esto a razón que la líder Sennova del centro gestiona la totalidad de la información de los proyectos de investigación propuestos, en curso y finalizados.

El análisis de la información se realizó clasificando la información proporcionada en subprocesos los cuales a su vez se trabajaron con el detalle suficiente para determinar la información necesaria en cada caso, posteriormente se elaboró el modelo entidad relación de cada subproceso los cuales en conjunto conformaron el modelo conceptual de la base de datos de la plataforma.

En la codificación de la plataforma se empleó el IDE Visual Studio 2015 Community y ASP.NET MVC Framework, se trabajó con el lenguaje de programación C#, y para dotar de dinamismo a la interfaz se empleó, java script, jquery y ajax.

En cuanto a la renderización de vistas se trabajó con Razor, y para el manejo de la lógica de las mismas se utilizaron Display Templates y Editor Templates. Las validaciones en el servidor a nivel de modelo incorporaron las funcionalidades de data-annotations, las cuales a su vez se complementaron con la validación del lado del cliente, para lo cual se incluyeron librerías javascript.

A nivel transaccional se trabajó con SQL Server 2016 combinando el enfoque DataBaseFirst y CodeFirst del ORM Entity Framework. Codefirst incluyó la utilización del modelo de Migraciones presentes en Entity Framework. En cuanto a la gestión de versiones se utilizó Team Explorer de Visual Studio apoyados en las cuentas del dominio @sena.

Con base al proceso de desarrollo Scrum se definieron varios sprint, dentro de los cuales se realizaron reuniones de manera periódica las cuales se pueden clasificar en cuatro fases. En la primera participaban el equipo de desarrollo y la líder Sennova (fase de recolección de información) del centro, a fin de establecer los requisitos de la plataforma. Posteriormente en una segunda fase (fase de diseño) se llevaron a cabo reuniones solo con el equipo de desarrollo, esto a fin de puntualizar sobre el proceso de construcción donde se incluían temas como: modelo

entidad relación, padrón de diseño, validaciones, modelo de clases y gestión de las versiones de código fuente, en una tercera fase (fase de construcción) se realizó la codificación de los módulos funcionales y se estableció con la líder Sennova la manera como se realizarían las pruebas y el cargue de la información inicial a la base de datos. Para la última fase (fase de despliegue) se realizó el despliegue de la plataforma usando los servicios de Internet Information Server (IIS).

Los sprints, junto con su objetivo principal, se detallan en la siguiente tabla.

Tabla 1. Sprints realizados durante la construcción de la plataforma.

Fase	Nombre del Sprint	Objetivo
Fase de recolección de información	Definición de instrumentos de recolección de información.	Establecer los instrumentos idóneos para el levantamiento de información.
	Elaboración y validación de instrumentos de recolección de información.	Establecer en detalle las características de los instrumentos de recolección de información.
	Reuniones semanales con la líder Sennova y el equipo de desarrollo.	Recolectar de manera puntual la información que deberá gestionar la plataforma web.
Fase de diseño	Elaboración del modelo conceptual de la base de datos.	Construir el modelo entidad relación de la base de datos.
	Patrón de diseño y estándares de codificación.	Definir el patrón de diseño y validaciones a nivel gráfico y de modelo.
	Gestión de versiones de código.	Definir la herramienta para gestionar las versiones de código fuente.
Fase de construcción	Estándar para codificación sobre el patrón de diseño seleccionado.	Definir el estándar para generación de código fuente con base al modelo de clases y el patrón de diseño.
	Generación de código y pruebas unitarias.	Construir los módulos funcionales de la plataforma web.
	Pruebas funcionales y cargue de información.	Diseñar el plan de pruebas funcionales y cargue de información inicial a la base de datos.
Fase de despliegue	Preparación de infraestructura.	Realizar la configuración y preparación de la infraestructura necesaria para publicar la plataforma web.
	Puesta en funcionamiento en servidor.	Desplegar la plataforma web en el servidor para su puesta en marcha.
	Plan de capacitación.	Capacitar a los usuarios finales de la plataforma web.

Resultados

En la primera fase luego de realizar reuniones periódicas a fin de recolectar la información necesaria mediante entrevistas, y luego de realizar el análisis correspondiente, se obtuvieron los subprocesos a sistematizar dentro de la labor de investigación en el centro de gestión administrativa y fortalecimiento empresarial. En la siguiente figura se ilustra los subprocesos identificados, así como la información necesaria en cada caso.

Plataforma web para el centro de gestión administrativa y fortalecimiento empresarial

El patrón de desarrollo de la “Plataforma web para la gestión de conocimiento y divulgación de resultados de Investigación en CEGAFE¹”, es el Modelo Vista Controlador, el cual se encuentra presente en el framework² ASP.NET MVC³. Esta arquitectura permite aislar los componentes y las responsabilidades de las distintas capas; Modelo, Vista y Controlador, con un nivel de acoplamiento mínimo entre las mismas, lo que genera una aplicación fácil de mantener, de controlar y con un alto grado de adaptabilidad.

A continuación, se describen las funcionalidades esenciales que se incorporaron en el desarrollo de la plataforma web y sus componentes.

Modelo

Representa el dominio de la aplicación y gestiona la persistencia de datos a través de la incorporación de sus entidades⁴ utilizando el ORM⁵ (Object Relational Mapper) Entity Framework⁶ bajo el enfoque Database-first⁷, empleando Fluent API⁸ para representar sus relaciones y la técnica de Eager Loading⁹ para indicar explícitamente la forma en que se desean cargar las entidades relacionadas.

¹Centro de Gestión Administrativa y Fortalecimiento Empresarial

²Marco de trabajo

³Plataforma de desarrollo web de Microsoft basada en el patrón Modelo Vista Controlador

⁴Clase que representa datos en una aplicación

⁵Mapeo de objeto-relacional, Modelo de programación que consiste en transformar las tablas de bases de datos relacionales en entidades de datos utilizando programación orientada a objetos

⁶Tecnología que permite trabajar con modelos conceptuales que se asignan a esquemas lógico de orígenes de datos

⁷Generación de entidades a través de tablas de bases de datos

⁸Permite configurar de forma fluida y centralizada las propiedades de las entidades

⁹Técnica para cargar entidades relacionadas

Para el versionamiento de base de datos se trabajó con migraciones las cuales permiten mantener actualizada la base de datos a través de la utilización de comandos en el Package Manager Console¹⁰, ofreciendo de esta forma la ventaja de mantener un seguimiento sobre cada uno de los cambios generados a nivel transaccional, los cuales son registrados en una tabla de la base de datos.

A nivel de presentación para indicar la manera en que el controlador entrega la información a la vista, se utilizó el esquema de data annotations¹¹, con la implementación de atributos que ofrecen al usuario una mejor experiencia en la visualización de información, al personalizar los controles que hacen parte de la vista.

Otra de las utilidades de data annotations a nivel de modelo es la definición de restricciones para la validación de datos, evitando la inconsistencia de información entre vista y modelo cubriendo de esta forma la validación en el servidor, es así como se incluyó una personalización de los mensajes de error sobre el tipo de datos de cada una de las propiedades que hacen parte de las entidades del modelo y que son gestionados desde el controlador y presentadas en la vista, las validaciones que no fueron cubiertas a través de los atributos de data annotations y que indicaban un comportamiento específico se definieron utilizando los siguientes esquemas:

- Implementación del método Validate de la interfaz IValidatableObject, el cual devuelve un listado de los errores al encontrar violación en las restricciones de los tipos de datos de las propiedades del modelo.
- Creación de un componente especial dentro del modelo para alojar las validaciones que son comunes a varias entidades, lo cual permite la facilidad de mantenimiento al centralizar las tareas de actualización reflejadas en una propagación funcional del código a lo largo de la aplicación, cumpliendo de esta forma con uno de los principios claves en la programación DRY¹² (Don't Repeat Yourself)

¹⁰Consola de administración de paquetes

¹¹Atributos destinados a la validación de y presentación de datos

¹²Principio de programación, no te repitas busca evitar la repetición de código innecesario

Controlador

En su papel de capa intermedia entre la vista y el modelo se divide su funcionalidad en dos esquemas: Sistema de routing¹³ para lo cual se trabajó con ruteo tradicional, y la definición de controladores en donde sus acciones en muchas ocasiones utilizó como retorno el formato JSON¹⁴ (JavaScript Object Notation) debido a su naturaleza para realizar transacciones de información de forma más rápida y compacta, este proceso consistió en tomar objetos de C#, convertirlos en JSON y enviarlos en forma de string a la vista, lo cual generó un mayor rendimiento dentro de la aplicación

Con el propósito de encapsular la lógica y el acceso a datos se definen una serie de clases gestoras que permiten mantener aisladas las responsabilidades de las entidades del modelo de una forma eficiente.

Para realizar la validación en servidor de los datos enviados por el usuario, se utilizó model binder, el cual consiste en la comprobación del cumplimiento de las restricciones del Modelo a través de la utilización de data annotations; de esta forma al instanciar una vista, el controlador envía la información necesaria para la maquetación y la información sobre el estado del modelo.

Vista

El alcance de la capa Vista se puede resumir en tres aspectos fundamentales: generación de resultados enviados al usuario, maquetación de interfaz e implementación de la lógica de presentación. En cuanto a la maquetación se utilizó HTML, CSS, Bootstrap¹⁵ y en ciertos casos Javascript con el fin de dotar a las vistas de autonomía y dinamismo.

Para la generación de controles de edición y visualización se empleó una técnica fundamentada en la tecnología de datos dinámicos, la cual se basa en el tipo de datos de las propiedades de las entidades del modelo, como lo son los tipos bool o DateTime. Se utilizaron también los DisplayTemplates y EditorTemplates, las cuales se ubicaron en un lugar específico dentro de la estructura de la aplicación de acuerdo con las necesidades de imple-

¹³ Controlador frontal que recibe las interacciones de la capa del cliente

¹⁴ JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) formato ligero de intercambio de datos

¹⁵ Framework de diseño web que permite la construcción de sitios adaptativos

mentación; cuando se evidenció que la presentación era común para varias entidades se colocaron en la carpeta compartida de las vistas del proyecto, sin embargo también se incluyeron dentro de cada vista cuando la funcionalidad de presentación era más específica.

La presentación de controles se realizó a través de la utilización de Helpers¹⁶ que ofrece el framework ASP.NET MVC, diferente a la forma tradicional como se trabajaría en un documento HTML. La inclusión de Helpers como BeginForm permitió una mayor flexibilidad al momento de referenciar una url, ya que su ruta está basada en acciones y controladores que se mapean en la tabla de ruteo; por otra parte, se evidenció un mapeo excelente de las propiedades del modelo y los controles en la vista, ya que a cada elemento HTML se le asigna por omisión como atributo el nombre de la propiedad de la entidad a la que está haciendo referencia, dichos Helpers permitieron a su vez generar mensajes de validación por campos, que demuestran algún tipo de inconveniente encontrado con la propiedad del modelo a la cual se encuentra asociado.

Se utilizaron algunas librerías y plugin de JavaScript para realizar la validación en el lado del cliente, las cuales fueron referenciadas a través de vistas maestras o vistas particulares dependiendo de su implementación.

En cuanto a validaciones personalizadas se incorpora la funcionalidad de Javascript¹⁷ utilizando el atributo Remote¹⁸ de data annotations, lo cual permitió realizar una comprobación vía Ajax, para lo cual se creó una acción en el controlador que retorna un JSON (JavaScript Object Notation) con un valor booleano que indica el resultado de la validación.

Muchas veces se evidenciaron vistas similares en el proyecto, las cuales podían ser reutilizadas por otros componentes del proyecto. Para esto se recurrió a las vistas parciales, las cuales, sencillamente, abstraen fragmentos de código que se encapsulan en las vistas, permitiendo que cuando se realice un cambio en dicha vista parcial, la propagación se realice en todas las vistas que realizaron su encapsulamiento.

Seguridad

¹⁶ Método que devuelve una cadena de texto

¹⁷ Lenguaje de programación utilizado para crear webs dinámicas

¹⁸ Atributo de validación Javascript

Para asegurar la protección de información relevante como es el caso de las credenciales de acceso a base de datos se procedió a realizar la encriptación de la conexión a base de datos a través de los comandos propios del Framework de .NET.

Con el objetivo de combatir las vulnerabilidades a las cuales se ve expuesta una aplicación web se implementaron las siguientes acciones:

- Para la inyección Sql la cual es conocida por ejecutar queries¹⁹ arbitrarios a la base de datos por usuarios no autorizados se utilizó: Entity Framework LINQ/Lambda, queries parametrizados y procedimientos almacenados.
- Cross-Site Scripting (XSS) esta vulnerabilidad permite al usuario correr código arbitrario de javascript, para esto se evitó la utilización del método raw y se validó la información enviada a las vistas a través de anotaciones.
- Referencia Directa Insegura: se trabajó con la restricción de las anotaciones en este caso se utilizó Authorize junto con el esquema de autenticación del MVC en su clase Identity.
- Cross-Site Request Forgery (CSRF) o falsificación de petición en sitios cruzados. Una de las técnicas utilizadas para la validación entre Vista y Controlador al momento de realizar un POST fue AntiForgeryToken, la cual garantiza que el usuario que realizó la solicitud inicial es el mismo que envía la información al servidor. Cuando se hace la solicitud a la página se genera un token para el usuario, dicho token se verifica cuando el usuario envía la información al servidor evitando que sea un tercero el que se encuentra realizando la suplantación del usuario inicial. Para completar con el esquema en el controlador cada acción Post se decoró bajo el esquema de anotaciones utilizando ValidateAntiForgeryToken.
- Vulnerabilidad de Asignaciones Masivas (OverPosting). En muchas oportunidades no se desea que la información desde la vista al controlador coincida en su totalidad con la cantidad de propiedades de la entidad que estamos afectando o realizando el correspondiente POST, para ello se utilizó el model binding el cual permite excluir las propiedades que no se quisieron incluir en la vista.

Autenticación

¹⁹ Consultas a bases de datos

Para la autenticación y membresía se trabajó bajo el esquema de Individual User Account²⁰ que maneja el framework MVC por defecto para lo cual se definieron roles, usuarios y permisos, definiendo restricciones a controladores y acciones utilizando anotaciones las cuales permiten identificar el nivel de acceso a cada componente del controlador.

Por otra parte, se configuró el bloqueo al acceso de la aplicación tras 5 intentos fallidos de ingreso por un período de cinco minutos. Se trabajó con programación asíncrona para todo el esquema de autenticación de usuarios, asignación de roles y permisos.

Rendimiento

Para la presentación de datos en la vista se utilizó lo que se puede denominar un postback parcial, ya que no se escribe todo el documento HTML sino una parte de este. Este postback parcial invoca código C# desde javascript utilizando Asynchronous Javascript and XML (AJAX).

Con el fin de lograr un mayor rendimiento en el acceso a la información se utilizó OutputCache, en la cual se definió el tiempo de duración del cache, el lugar almacenamiento o consulta y la forma en la que se debe actualizar la información presente en el mismo. Esto resultó muy útil cuando se comprobó la existencia de acciones que no generan gran cantidad de cambios a nivel transaccional.

Conclusiones

Teniendo en cuenta que gran parte de los productos originados en el proceso de investigación son digitales o tienen un respaldo en un archivo digital, la construcción de una plataforma web que ofrezca acceso a los usuarios con acceso a internet se considera la opción ideal para dar solución a la problemática planteada. Todo esto, al tener en cuenta que la información estaría centralizada en un servidor en las instalaciones del centro y así su gestión no dependería de un funcionario en particular.

El diseño de una herramienta tecnológica para gestionar los procesos y re-

²⁰

Esquema de autenticación de usuario del framework ASP.NET MVC

sultados del proceso de investigación en CEGAFE permite la estandarización de procesos al interior del centro. Esto, teniendo en cuenta que la dinámica de la investigación en las diferentes disciplinas permite cierta independencia en el desarrollo de proyectos que origina inconvenientes para consolidar la información del proceso y la gestión de resultados.

ASP.NET MVC brindó la oportunidad de desarrollar una plataforma con responsabilidades bien definidas para cada uno de sus componentes y un nivel de acoplamiento mínimo entre sus capas. Si bien es cierto que MVC define la existencia de tres capas para el desarrollo del software, a medida que se avanzó en la construcción de la plataforma se vio la necesidad de utilizar componentes que aportaron funcionalidad a las capas principales de MVC, las cuales incluyeron lógica entre el controlador y el modelo y en ocasiones lógica para la vista.

Aprovechando las funcionalidades del framework se desarrolló una lógica para cubrir de forma funcional los dos sistemas de validación cliente y servidor, en materia de seguridad se utilizaron técnicas como encriptación de componentes, se tomaron patrones para evitar la vulnerabilidad de la inyección sql, a través de la utilización de consultas parametrizables y procedimientos almacenados, finalmente model binding permitió tener un mayor control sobre la información enviada de la vista al controlador. Todo esto se traduce en una aplicación con responsabilidades claramente definidas lo cual facilitara de mantenimiento, mayor flexibilidad y rendimiento en la plataforma.

Referencias

Microsoft – Developer Network. (2017). *Information general sobre ASP.NET MVC*. Recuperado de [https://msdn.microsoft.com/es-es/library/dd381412\(v=vs.108\).aspx](https://msdn.microsoft.com/es-es/library/dd381412(v=vs.108).aspx)

Microsoft – Developer Network. (2017). *Model-View-Controller*. Recuperado de <https://msdn.microsoft.com/en-us/library/ff649643.aspx>

Ecured.(2017). CSS3. Recuperado de <https://www.ecured.cu/CSS3>

Microsoft – Developer Network. (2017). *Information general sobre ASP.NET*. Recuperado de [https://msdn.microsoft.com/es-es/library/4w3ex9c2\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/4w3ex9c2(v=vs.100).aspx)

Aprende Web Crea y diseña tus sitios en la Web.(2017). Manual de Ajax. Recuperado de http://aprende-web.net/progra/ajax/ajax_1.php

Torres, D., Delgado, E. (2009). *Estrategias para mejorar la difusión de resultados de investigación con la Web 2.0*. Granada, España: El Profesional de la información.

Microsoft – Developer Network. (2017). Tutorial: *Crear un sitio web con sintaxis de Razor en Visual Studio*. Recuperado de [https://msdn.microsoft.com/es-es/library/gg606533\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/gg606533(v=vs.100).aspx)

Programar facil. (2016). *Novedades del Visual Studio 2015, desarrollo multi-plataforma*. Recuperado de <https://programarfacil.com/blog/programacion/novedades-del-visual-studio-2015-desarrollo-multi-plataforma/>

Microsoft.(2017). *Work in Team Explorer*. Recuperado de <https://www.visualstudio.com/en-us/docs/connect/work-team-explorer>

git- distributed-even-if-your-workflow-isnt. (2017). *Empezando - Acerca del control de versiones*. Recuperado de <https://git-scm.com/book/es/v1/Empezando-Acerca-del-control-de-versiones>

softeng. (2017). *Metodología SCRUM para el desarrollo de software - aplicaciones complejas*. Recuperado de <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>

Procesos de Software.(2017). *Metodología SCRUM*. Recuperado de <https://procesosdesoftware.wikispaces.com/METODOLOGIA+SCRUM>