

## Desarrollo del CMS Monoil, una aplicación destinada al diseño, presentación y administración de contenidos relacionados al proyecto Monoil de la Universidad de Guayaquil

[ Development of CMS Monoil, an application destined to design, render and manage contents related to Monoil project - University of Guayaquil ]

*Luis Cabrera Arias, Christian Antón Cedeño, and Diana Minda Gilces*

Faculty of Mathematics and Physics Science, University of Guayaquil, Guayaquil, Ecuador

Copyright © 2017 ISSR Journals. This is an open access article distributed under the *Creative Commons Attribution License*, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**ABSTRACT:** CMS stands for Content Management System. There are different types of CMS, usually, refers to Web CMS (WCMS) which is a software that allows common users to manage content of web pages without necessary knowledge from web developers or databases administrators. Project of the research program "Monitoring of Oil activities in Ecuador : a cross-disciplinary approach between Environment, Health and People" better known as Monoil, needs to publish information to internet, for example, data from agreements, associations, investigators résumé, investigations, news, etc. In the way we are thinking about web sites, all these data is treated as a "content", then, Monoil needed a WCMS. Before solutions that were implemented, demanded certain performance and managers (of contents) were not agreed with them. Thus, the project decided to create CMS Monoil application which is a made-to-order Manager Content software. This solution includes CMS Monoil Application, destined to investigators and general public, and a Module for maintaining tables of CMS, for administrators who change frequently the content of the pages; these projects are included in the main solution Monoil as a built-in software. The CMS is developed with open source technologies such as Java platform with JSF Framework, Apache Tomcat web application and PostgreSQL database. In order to carry out this project with Software Engineering standards, SCRUM was selected as the agile methodology of developing, which through Sprints, created the CMS Monoil under these agile practices.

**KEYWORDS:** CMS, Content Management System, Information, Web, WCMS, Web Content Management System.

**RESUMEN:** Un Sistema de Gestión de Contenidos CMS es aquel que permite administrar correctamente la información de un proyecto o entidad. Comúnmente un CMS se refiere a un Web CMS (WCMS), el cuál es un software que permite a los usuarios comunes administrar el contenido de las páginas web sin el conocimiento necesario de desarrollo web o administración de bases de datos. El proyecto binacional (Francia-Ecuador) denominado Monitoreo ambiental, salud, sociedad y petróleo en Ecuador "Monoil", necesita publicar información en Internet, por ejemplo, datos de acuerdos o convenios, instituciones participantes, currículum de investigadores, diferentes investigaciones, noticias, etc. Cada uno de estos datos es tratados como un "contenido", Monoil necesitaba un WCMS. Las soluciones que se implementaron anteriormente exigían cierto rendimiento y capacidades de almacenamiento, por lo que, los administradores (de contenidos) no estaban de acuerdo con ellas. Es así como se decidió crear la aplicación CMS Monoil, un gestor de contenidos hecho a la medida del proyecto. Esta solución incluye la aplicación CMS Monoil, destinada a investigadores y público en general, y un módulo de administración de las tablas del sistema para los administradores; estos sistemas se incluyen en la solución Monoil como un software incorporado. El CMS trabaja con tecnologías de código abierto como la plataforma Java con el Framework JSF, Apache Tomcat y PostgreSQL. Para llevar a cabo este proyecto con los estándares de Ingeniería de Software, SCRUM fue seleccionado como la metodología ágil de desarrollo, la cual a través de Sprints, creó el CMS Monoil.

**PALABRAS CLAVES:** CMS, Sistema gestor de contenidos, información, Web, WCMS, Sistema gestor de contenidos web.

## **1 INTRODUCCIÓN**

El presente documento se refiere a la importancia de los Content Management System, más conocido por sus siglas en inglés CMS (Sistemas de Gestión de Contenidos), para la información pública y privada de una entidad, en este caso orientado exclusivamente al Proyecto Binacional (Francia – Ecuador) denominado Programa de Monitoreo Ambiental, Salud, Sociedad y Petróleo del Ecuador “Monoil”; por lo que éste se ve en la necesidad de poder gestionar cada uno de sus contenidos (información obtenida de investigaciones, resultados, acuerdos, noticias, entre otros) que van a servir para el público en general, la comunidad de investigadores y los administradores de contenidos web del proyecto.

La información ha sido tratada desde los tiempos más remotos de la humanidad; era necesario comunicar sucesos, relacionarse, dejar constancia eventos y así impartir este conocimiento a sus próximas generaciones. Por lo que, se puede argumentar que la información nace de los datos tomados de hechos, los cuales deben ser debidamente tratados y procesados para ser impartidos correctamente, deben tener un orden y una secuencia lógica para su correcto entendimiento. [1]

Dejando las costumbres de antaño en donde se realizaban pinturas señalizadas en rocas, cuevas dejando su conocimiento, damos un gran salto a la era informática y nos encontramos con el nacimiento de la Internet. Aquí la información toma un rumbo diferente, todo con la intención de automatizarse, evitar los papeles y tenerla almacenada en medios digitales; este es un paradigma que se ha ido introduciendo cada vez con mayor intensidad con el pasar de los años. “A partir de la década del 80, la información comienza a ser asumida, de manera consciente, como un recurso estratégico y vital, capaz de acelerar el progreso en todos los espacios: sociales, económicos, tecnológicos, políticos, entre otros”. [2]

La información que se halla en Internet, debe ser correctamente administrada por las empresas o compañías que brindan el conocimiento a un usuario final; el tipo de gestión de contenidos varía dependiendo de la importancia, políticas internas, entre otros aspectos. Hoy en día, la mayoría de las páginas web a las que accedemos a través de nuestros navegadores son creadas por algún tipo de gestor de contenidos.

Un CMS no es sólo un sistema encargado de dibujar o diseñar sitios en internet, sino que va más allá de eso, entre los tipos de CMS más conocidos tenemos los Enterprise Content Management, por sus siglas en inglés ECM (Administradores de Contenidos Empresariales), Document Management System, por sus siglas en inglés DMS (Administrador de Documentos y Contenidos Multimedia), entre otros. Pero, no todos los estos sistemas son adecuados para una misma situación debido a la complejidad o sencillez que presentan estos diferentes sistemas. [3]

El proyecto Monoil se ve en la necesidad de contar con un sistema capaz de administrar sus contenidos con la finalidad de actualizar de forma inmediata, sencilla y práctica la información que se debe mostrar a la sociedad de investigadores de Monoil y cualquier tipo de persona que desee acceder a las publicaciones del proyecto. Inicialmente se deseó introducir un CMS completo con todas las funcionalidades que existen en el mercado, sin embargo, esto implicaba subutilizar los recursos del CMS; adicionalmente no cubría las necesidades de los requerimientos presentados de un administrador de contenidos exclusivo para el proyecto Monoil. Además se deseaba que el administrador de contenidos funcione de la mano con la base de datos implementada, para que así el CMS sirva de enlace o entrada a la solución tecnológica Monoil desarrollada por la Universidad de Guayaquil.

Además, existen situaciones críticas dentro del actual sistema de gestión de contenidos, el cual, al no trabajar conjuntamente con la base de datos de Monoil, no puede actualizarse debidamente. En la actualidad existen imágenes estáticas y contenidos que no son relacionados, los cuales, en ocasiones se pierden dentro del diseño actual y no brindan una buena experiencia para el usuario que visite el sitio.

Actualmente, los profesionales de la información sirven de mediador entre las tecnologías de información y el usuario. Ellos se encargan de gestionar el contenido existente en las bases de datos, actualizando el contenido de la página que llega a los usuarios. [4] Definir un CMS, es un trabajo difícil, se debe investigar a fondo cada una de las características que poseen, lo que conlleva en ocasiones al uso excesivo de recursos, para cada caso será diferente debido a sus requerimientos. Monoil está en la necesidad de tener un Sistema Gestor de Contenidos por lo que, se dedujo que su idóneo sería uno personalizado, el cual cuente con características y funcionalidades desarrolladas en base a los requerimientos exclusivos del proyecto binacional.

Dentro de los productos tecnológicos que aporta la presente solución al proyecto general Monoil tenemos:

- El CMS Monoil desarrollado bajo las tecnologías JavaServer Faces, con la integración de Primefaces, el cual será la aplicación que interactuará constantemente con los usuarios públicos o de la comunidad científica de Monoil, en donde encontrar información a fin sus intereses y podrá acceder a la plataforma Monoil.

- El M3dulo de administraci3n del CMS Monoil (MACMS Monoil) desarrollado bajo las tecnolog3as JavaServer Pages, con la integraci3n del Framework de desarrollo web Bootstrap. Este m3dulo formar3 parte del Sistema general de Monoil, en el cual se administraran dem3s contenidos, donde el CMS Monoil es parte de la soluci3n total.

## 2 SISTEMA ADMINISTRADOR DE CONTENIDOS

Un Content Management System (CMS) que traducido al espa1ol significa Sistema Administrador de Contenidos, es un sistema capaz de administrar (ingresar, modificar, eliminar, mostrar) estructuralmente la informaci3n perteneciente a una entidad, empresa o proyecto; sean estos textos, im3genes, documentos, v3deos de soporte o cualquier tipo de informaci3n relacionada a un tema espec3fico para cumplir prop3sitos espec3ficos de la entidad.

Los sistemas evolucionan constantemente y los CMS no se han quedado atr3s, por lo que hoy en d3a existen varios tipos administradores con conceptos y amplitudes diferentes, entre ellos, el m3s com3n es el Web Content Management System (WCMS) que en espa1ol significa Sistema Administrador de Contenidos Web. Por lo general, cuando se escucha acerca de un CMS, lo primero que se viene a la mente son los WCMS; la mayor3a de estos sistemas funcionan con similares tecnolog3as de tipo *open source* o denominado *software libre* como son Apache, PHP y MySQL. [5]

En Internet, existen otros tipos de CMS m3s espec3ficos, con diferentes funcionalidades acorde a diferentes requerimientos, es decir el tipo de contenido que gestionan; algunos van m3s all3 de administrar 3nicamente sitios web como por ejemplo el Sistema Administrador de Contenidos Empresariales, Sistema Gestor de Documentos, entre otros. A continuaci3n en la Tabla 1, se muestra una breve clasificaci3n de los tipos de administradores m3s utilizados en la actualidad. [3]

**Tabla 1. Clasificaci3n de los sistemas de gesti3n de contenidos**

Siglas	Ingl3s	Espa1ol
<b>DMS</b>	Document Management System	Sistema Gestor de Documentos
<b>WCMS</b>	Web Content Management System	Sistema Administrador de Contenidos Web.
<b>KMS</b>	Knowledge Management System	Sistema Gestor de Conocimiento
<b>LMS</b>	Learning Management System	Sistema Administrador de Aprendizaje
<b>SCMS</b>	Software Configuration Management System	Sistema Gestor de Configuraci3n de Software
<b>ECMS</b>	Enterprise Content Management System	Sistema Administrador de Contenidos Empresariales
<b>BPMS</b>	Business Process Management System	Gestores de Flujos de Trabajo o Workflow

Por lo tanto, se puede comprender que un CMS abarca diferentes conceptos, magnitudes y funcionalidades, por lo que, a partir de ahora en adelante cuando se mencionen las siglas <<CMS>> o las palabras <<Administrador de Contenidos>>, se estar3 haciendo referencia a los Administradores de Contenidos Web (WCMS), debido a que 3ste es el tema principal del presente art3culo.

Un CMS es un tipo de soluci3n que facilita el trabajo de almacenamiento en una o m3s bases de datos la detallada informaci3n de un sitio web, este sistema se basa en la creaci3n de una estructura de soporte para gestionar o administrar cada una de los contenidos, com3nmente estos sistemas se alojan en servidores web. [6]

### 2.1 CARACTER3STICAS DE UN CMS

Entre las caracter3sticas del CMS tenemos:

- Actualizaci3n constante del contenido del sitio.
- Base de datos robusta.
- Hist3ricos de p3ginas y sus contenidos.
- M3dulos intuitivos de mantenimiento de las tablas del sitio.
- B3squeda de contenidos espec3ficos
- Autenticaci3n y autorizaci3n de usuarios (roles).
- Divide la parte visual de la l3gica.
- Establece procesos de aceptaci3n de contenidos.
- Control de Internacionalizaci3n del Sistema

A continuación se explicarán cada una de las características antes mencionadas enfocadas en el entorno de ejecución del CMS Monoil las cuales van de la mano con el alcance del proyecto.

#### Actualización constante del contenido del sitio.-

El CMS es capaz de crear o alterar sus contenidos periódicamente, por lo que en el momento en que el administrador cambie la información actual a través del Módulo de Administración del CMS Monoil (MACMS Monoil), este cambio se verá reflejado tanto en la pantalla de administración, como en el producto para los usuarios finales, el CMS Monoil, que es la interfaz que interactúa con el usuario, presentando así siempre contenidos actualizados.

#### Base de datos robusta.-

El secreto detrás del CMS Monoil, es su base de datos debido a que ésta contiene la información de manera ordenada y relacionada entre sí para poder obtener el resultado, el contenido. El CMS cumple con la normativa ACID que significa Atomicidad, Consistencia, Aislamiento y Durabilidad; mantiene la información de manera íntegra, consistente y disponible cuando ésta se requiera.

#### Históricos de páginas y sus contenidos.-

Un gran reto para un sistema es almacenar datos históricos. Esto se ha venido introduciendo desde la década de los 80. Gracias al registro de estos datos, se puede estudiar o supervisar el comportamiento o crecimiento del sistema a través del tiempo [7]. En este caso el CMS es capaz de almacenar cada una de las modificaciones de sus contenidos y determinar en qué fecha fue alterado, qué usuario realizó la modificación y de esta forma poder recuperar información y tomar medidas correctivas si el caso lo amerita.

#### Módulos intuitivos de mantenimiento de las tablas del sitio.-

Debido a que el CMS utiliza un sistema bien estructurado de base de datos, debe de existir una forma de administrar los datos existentes en sus tablas, por lo que estos sistemas cuentan con el Módulo de Administración de CMS Monoil que interactúa con las diferentes tablas que ayudan a gestionar correctamente el sistema administrador de contenidos.

#### Autenticación y autorización de usuarios (roles).-

Al establecer usuarios dentro la solución total Monoil, significa que éstos van a tener sus respectivos roles, por lo que sólo los usuarios autorizados pueden gestionar los diferentes contenidos, evitando que un administrador lleve toda la carga de gestionar la información de un sitio web. El Sistema Monoil incluye las respectivas normas de autenticación y autorización para determinar qué usuarios están en la capacidad de administrar los contenidos del sitio.

#### Divide la parte visual de la lógica.-

Al utilizar el patrón Modelo Vista Controlador MVC, se puede separar la lógica de negocio que realiza el proceso para presentar las imágenes, la lógica de modelos que se encarga de obtener la información de la base de datos Monoil y finalmente la lógica que se utiliza en las vistas o páginas web dinámicas, es utilizada exclusivamente para su contenido con sus respectivos diseños para diferentes dispositivos, sean estos teléfonos inteligentes, tabletas, laptops o computadores de escritorio, teniendo así un diseño escalable para cada escenario.

#### Control de Internacionalización del Sistema.-

En la actualidad es común ver sitios con la capacidad de cambiar de idiomas, gracias a esto diferentes personas en cualquier parte del mundo pueden acceder a un sitio público informativo de la red. Dentro de los alcances del proyecto presentado por la Universidad de Guayaquil para la solución Monoil, no se contempla esta característica, pero, sí se diseñó el modelo de base de datos para que en una posterior versión del sistema, la internacionalización pueda ser implementada con facilidad (para los idiomas español, inglés y francés).

#### Búsqueda de contenidos específicos.-

Es de conocimiento general que la mayoría de sitios en Internet tienen esta característica incluida, que nos permite hacer búsquedas de contenidos específicos por la fecha, autor, lugar de publicación, entre otros campos. Con esto, el usuario se ve en la facilidad de encontrar información específica acorde a sus necesidades.

#### Establece procesos de aceptación de contenidos.-

Ciertos CMS tienen como lógica de negocio pasar por un proceso de aceptación del contenido, en donde el usuario elige un contenido y lo sube al sistema. Posteriormente este contenido es analizado por el o los administradores encargados y una vez que haya cumplido con los estándares de la compañía, éste es publicado en el sitio. El CMS Monoil no contempla esta

característica porque dentro de Monoil no existe un proceso de aceptación de contenidos, sólo pueden publicar contenidos los usuarios con el rol de “Administrador de Contenidos”.

### 3 TECNOLOGÍAS DEL CMS MONOIL

La solución informática del presente funciona bajo diferentes sistemas informáticos entre los cuales un grupo de ellos son utilizados en el Módulo de Mantenimiento del CMS (MACMS Monoil) y otros son exclusivos de la aplicación CMS Monoil. Se hace hincapié en esta sección debido a que la mayoría de soluciones del mercado tienen como común denominador las plataformas Apache, PHP y MySQL [5]; mientras el CMS Monoil es desarrollado con tecnologías relacionadas a la plataforma Java. A continuación se especifican las tecnologías haciendo énfasis en qué sección del proyecto son utilizadas y cuáles se presentan en ambas secciones.

Como norma de estandarización para el desarrollo de los módulos de mantenimiento de la solución Monoil (que incluye al Módulo de Administración del CMS Monoil), se fijaron tecnologías (JSP y Servlets) y elementos de diseño (Bootstrap) para desarrollar las diferentes soluciones que integran el proyecto. Entonces, a través de JSP y Servlets, se pueden presentar páginas web dinámicas, en este caso formularios de mantenimiento para la base de datos Monoil. Cabe recalcar que estos módulos serán accedidos dependiendo del usuario y los permisos que éste tenga.

Por otro lado, Bootstrap nos brinda la facilidad de añadirle estilos y ciertas funcionalidades al lado del cliente por lo tanto estas herramientas son importantes para solucionar la problemática. El módulo de administración está alojado en Apache Tomcat y trabaja conjuntamente con la base de datos PostgreSQL.

*Tabla 2. Comparación entre CMS Monoil y su módulo de administración*

CMS Monoil y MACMS Monoil	MACMS Monoil	CMS Monoil
<ul style="list-style-type: none"> <li>• JSE</li> <li>• JEE</li> <li>• NetBeans IDE</li> <li>• Apache Tomcat</li> <li>• PostgreSQL</li> </ul>	<ul style="list-style-type: none"> <li>• JavaServer Pages</li> <li>• JavaServer Pages Standard Tag Library</li> <li>• Java Servlet Technology</li> <li>• Bootstrap</li> </ul>	<ul style="list-style-type: none"> <li>• JavaServer Faces</li> <li>• Primefaces</li> </ul>

#### 3.1 JAVA

Java es un lenguaje de programación de alto nivel creado por Microsystems, se basa en el paradigma de la Orientación a Objetos (POO). Se estima que en la actualidad un gran número de sitios o aplicaciones web están desarrollados bajo este lenguaje. [8] Los arquitectos de software y desarrolladores requieren cada vez más de un ambiente distribuido, transaccional y portable que aproveche la velocidad, seguridad y fiabilidad de la tecnología del servidor [9].

La solución Monoil es desarrollada bajo el lenguaje de programación Java porque posee diferentes características. Por ejemplo, es un sistema multiplataforma, es decir puede ser ejecutado en cualquier sistema operativo ya sea este Linux, Windows, Mac OS, entre otros; esta característica ha posicionado a Java como uno de los lenguajes de programación más importantes, debido a que sólo se necesita tener instalada la Java Virtual Machine (JVM) que en español se denomina Máquina Virtual de Java, que es la encargada de gestionar cada uno de los procesos, sentencias, hilos, espacios de memoria, entre otros que hacen a este lenguaje de programación robusto. [10]

Otra ventaja es que se incluyen nuevas Application Programming Interface (APIs) para reducir la complejidad de implementación de código y así facilitar el trabajo al desarrollador para reducir tiempo; de entre las diferentes APIs, el proyecto Monoil implementa JavaMail, que brinda la funcionalidad de enviar correos electrónicos y los ManagedBeans quienes trabajan fuertemente con JavaServer Faces, los cuáles son sugeridos a partir de de Java Platform Enterprise Edition versión 6 (JEE). [11]

Una de las facilidades que adopta el CMS Monoil de JEE 6, es el uso de anotaciones en mayor cantidad con la finalidad de reducir la creación y modificación de archivos de configuraciones XML; debido a que las anotaciones dan esta funcionalidad al elemento afectado y el servidor de Java se encarga de las acciones de crear los archivos de configuración. [9]

#### 3.2 NETBEANS IDE

La solución CMS Monoil es desarrollada bajo el lenguaje de programación Java, y al ser una aplicación web dinámica necesita trabajar conjuntamente con ciertos lenguajes y se ve en la necesidad de crear y editar archivos bajo las extensiones

html, xml, xhtml, css, js, jsp, java; por lo que el presente implementa el Entorno de Desarrollo Integrado (IDE) NetBeans que brinda soluciones para los requerimientos antes mencionados.

Además, NetBeans soporta las tecnologías de Java con sus respectivos Frameworks, siendo así el IDE oficial para Java 8. Finalmente, facilita el uso de diferentes servidores con sus respectivos servicios, como servidores web y servidores de bases de datos. [12]

### **3.3 APACHE TOMCAT (TOMCAT)**

Una vez que se tengan todas las líneas de código del CMS Monoil y el MACMS Monoil, al ser aplicaciones web las antes mencionadas, deben estar alojadas y desplegadas en un servidor web o, sea el caso, un contenedor de servlets. Con Apache Tomcat se ponen a disposición las aplicaciones dinámicas a través de la red. Para la aplicación CMS Monoil, Apache Tomcat es capaz de alojar aplicaciones desarrolladas en JSP y servlets, y JSF.

Apache Tomcat<sup>®</sup> es una implementación de código abierto de tecnologías Java Servlet, JavaServer Pages, Java Expression Language y Java WebSocket. [13] Naturalmente Apache Tomcat trabaja como un contenedor web, en donde serán alojados los servlets para aplicaciones web dinámicas.

### **3.4 POSTGRESQL**

Entre las características más importantes del CMS se mencionó que la base de datos es un aspecto indispensable; en ella se almacenan los contenidos que van a ser mostrados al público en general y a los investigadores. La solución Monoil integra diferentes módulos y uno de ellos es el MACMS Monoil; por lo que se designa un esquema exclusivo para el CMS.

PostgreSQL es un Sistema Gestor de Base de Datos Relacional y Orientado a Objetos (ORDBMS, object-relational database management system) que entre sus características principales cuenta con bases de datos a nivel empresarial, es un sistema multiplataforma (se encuentra disponible para sistemas operativos como Linux, Mac OS X, Solaris y Windows), además se dice que cumple con las características Atomicity, Consistency, Isolation, Durability (ACID). [14] A continuación en la Tabla 3, se pueden observar alguno de los alcances principales de la herramienta PostgreSQL.

**Tabla 3. Alcances de PostgreSQL**

<b>Límite</b>	<b>Valor</b>
<b>Tamaño máximo de una base de datos</b>	Ilimitado
<b>Tamaño máximo de una tabla</b>	32 TB
<b>Tamaño máximo de un registro</b>	1.6 TB
<b>Tamaño máximo de un campo</b>	1 GB
<b>Máximo de registros por tablas</b>	Ilimitado
<b>Máximo de columnas por tablas</b>	250 - 1600 dependiendo el tipo de dato de las columnas
<b>Máximo de índices por tablas</b>	Ilimitado

### **3.5 JAVASERVER PAGES (JSP)**

JavaServer Pages o más conocido como JSP, como una tecnología de Java orientada al desarrollo de aplicaciones web. El MACMS Monoil incluye la biblioteca de JavaServer Pages Standard Tag Library (JSTL), además de las siguientes características:

- Textos estáticos como HTML y XML.- que me permiten dibujar la interfaz de usuario en las vistas JSP tales como botones, tablas, entre otros.
- Elementos JSP que ayudan a hacer una página web dinámica.- Aquí interviene el uso de secciones de código JSP denominada <<scriptlet>> o etiquetas JSTL que es lo más recomendado. [9]

### **3.6 JAVA SERVLET TECHNOLOGY**

Java Servlet Technology o simplemente conocido como Servlets, son clases que adquieren las capacidades de los servidores web bajo el protocolo Hypertext Transfer Protocol (HTTP); donde existen solicitudes o peticiones al servidor (request) y respuestas hacia el cliente que realizó la petición (response). Aquí se pueden dibujar nuevos elementos HTML que servirán para interactuar con el usuario encargado del MACMS Monoil. [10]

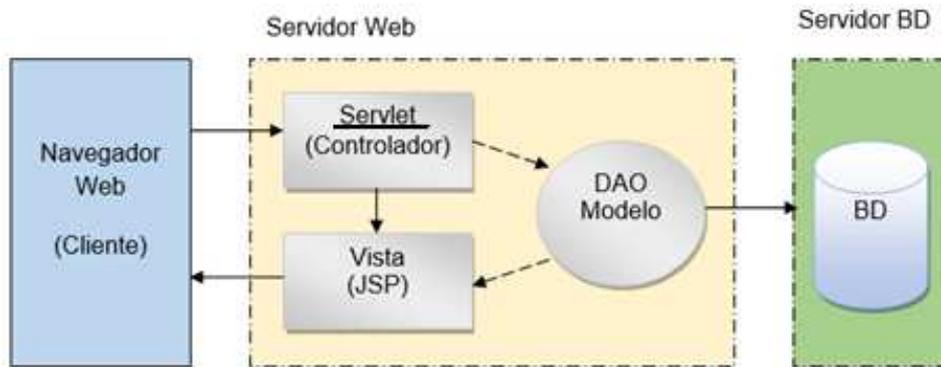


Fig. 1. Petición/respuesta de un Servlet

A continuación se detalla lo que se muestra en la Fig. 1, la arquitectura de la aplicación utilizando las tecnologías JSP y Servlets.

- El cliente.- A través de su navegador web realiza una petición a través del protocolo HTTP al servidor el cual queda a la espera de una posterior respuesta.
- El Servlet.- Recibe la petición HTTP y la procesa, determinando que vista JSP es la que se va a devolver. El Servlet hace las veces de controlador.
- La vista (JSP) obtiene la información dinámica del modelo (si así se dispone) mediante scriptlets o JSTL y está lista para ser visualizada por el cliente.
- La vista es devuelta al cliente mediante una respuesta HTTP <<response>>

### 3.7 BOOTSTRAP

Bootstrap es uno de los Frameworks más usados para el desarrollo de aplicaciones web, gracias a su sistema de rejillas o grillas que permite hacer que una página web sea *responsive*, además se utiliza para sitios en internet destinado a dispositivos móviles. Este Framework se basa en HTML, CSS y JS Framework, facilitando y estandarizando colores y diseños de los elementos de desarrollo web en la capa de front-end o interfaz de la solución Monoil. [15]

### 3.8 JAVASERVER FACES (JSF)

JavaServer Faces o también conocido por sus siglas JSF, es un Framework de desarrollo web bajo los estándares de Java, orientado hacia la interfaz de usuario. Presenta nuevas características como: interacción Asynchronous JavaScript And XML, AJAX con el usuario, reemplazo de JSP por XHTML facelets, navegación implícita, usar anotaciones en vez de descriptores en archivos de configuración XML, manejo de eventos, validaciones de componentes de entradas, entre otros. [16]

Para poder interactuar directamente con el usuario en la tecnología JSP se mencionaba que se utilizan los Servlets; pero, para este Framework, existen clases definidas como ManagedBeans o CDI Beans, que sirven de controladores. Además este Framework está orientado para trabajar bajo el patrón Modelo Vista Controlador, MVC.

El CMS Monoil implementará el uso de JSF debido a que al ser una aplicación destinada al público en general, requiere de un mayor alcance en la capa visual, aquí intervienen las ventajas de este Framework el cuál al trabajar conjuntamente con Primefaces, hacen de la aplicación un producto agradable para la vista de los clientes.

### 3.9 PRIMEFACES

Primefaces es un Framework de JSF de tipo *open source*, con diferentes elementos web, con sus características o propiedades y funcionalidades mejoradas del *core* o núcleo básico de JSF. Este componente facilita enormemente el trabajo de diseño de una página XHTML por lo que agiliza el desarrollo al momento de definirse por un tema o brindar estilos y funcionalidades AJAX a los componentes de las páginas web que interactúan directamente con el usuario. Además incluye herramientas para desarrollar aplicaciones web para dispositivos móviles. Su documentación es bien detallada por lo que es sencillo utilizar un elemento de Primefaces y darle la funcionalidad que se requiera. [17]

## 4 METODOLOGÍAS

### 4.1 GESTIÓN DE CALIDAD

Dentro de esta sección se realizó un estudio para obtener la situación o contexto en el que se encuentra el presente y el entorno luego de aplicar el CMS Monoil mostrando qué cambios se han realizados. Entre el análisis anterior al desarrollo de la solución tecnológica se encuentra el Diagrama de Ishikawa (Causa-Efecto); posteriormente, se explican los Casos de Uso que corresponden a los procesos implementados por la solución CMS Monoil.

### 4.2 DIAGRAMA DE ISHIKAWA

El Diagrama de Ishikawa o también denominado Causa-Efecto, permite encontrar las razones principales (causas) que conllevan a un gran problema (efecto). Estas causas son clasificadas y ubicadas dentro de las diferentes categorías y así se van obteniendo las razones para determinar cuál es el origen del problema. [18]

El principal problema o efecto se halla en que el Sitio Web Monoil actual presenta contenidos estáticos y a su vez desactualizados. Los elementos de la Fig. 2 que se encuentran dentro de recuadros, son considerados como categorías, las flechas horizontales son las causas en primer nivel y las líneas inclinadas u oblicuas indican las causas en segundo nivel.

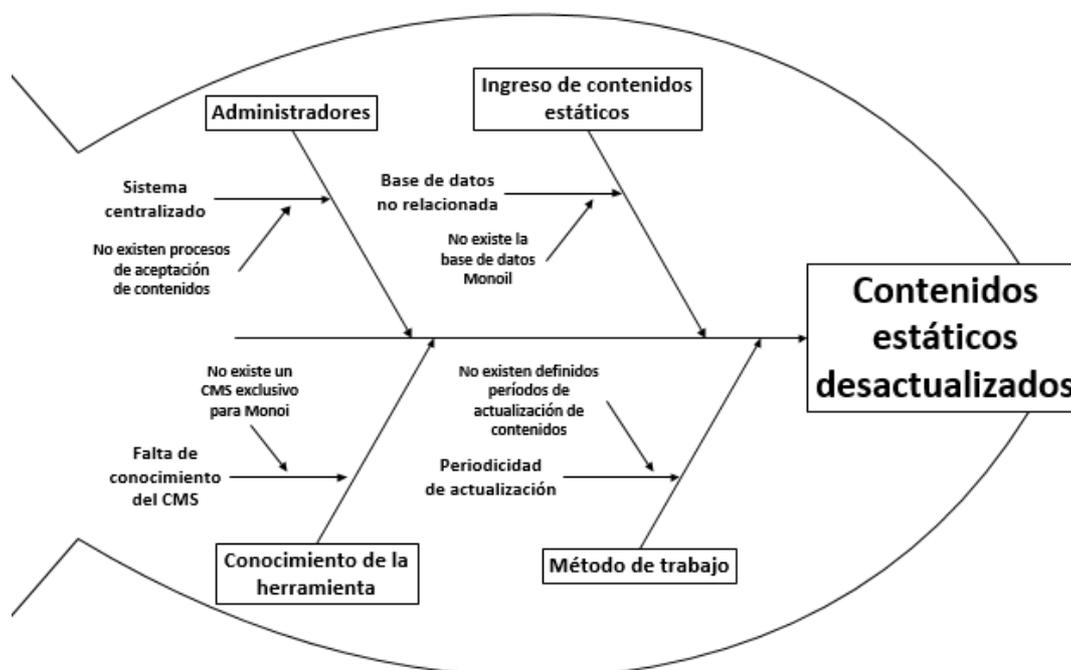


Fig. 2. Diagrama Causa-efecto

A continuación se presenta el análisis realizado luego de haber encontrado y clasificado las diferentes causas de Monoil.

#### 1. Categoría “Administración”

En esta categoría se encasilla al personal encargado de administrar los contenidos de Monoil. En la actualidad no todos los usuarios tienen permisos para subir contenidos al sitio sino solamente los que tienen el rol de “administrador de contenidos”, lo que conlleva a tener un sistema centralizado. En ocasiones sólo una persona se encarga de este trabajo, lo que produce tener contenidos desactualizados.

#### 2. Categoría “Ingreso de contenidos estáticos”.-

En esta categoría se hace referencia a que en ocasiones la información que se ingresa al sitio informativo son contenidos estáticos que no cambian o no se actualizan. El sistema actual no cuenta con la base de datos Monoil, por lo que optó por ingresar los contenidos directamente en la página que provee el actual CMS, al no tener la base de datos correctamente relacionada muchas veces los contenidos que se ingresan llegan a ser textos, imágenes de carácter estáticos.

### 3. Categoría “Conocimiento de la herramienta”.-

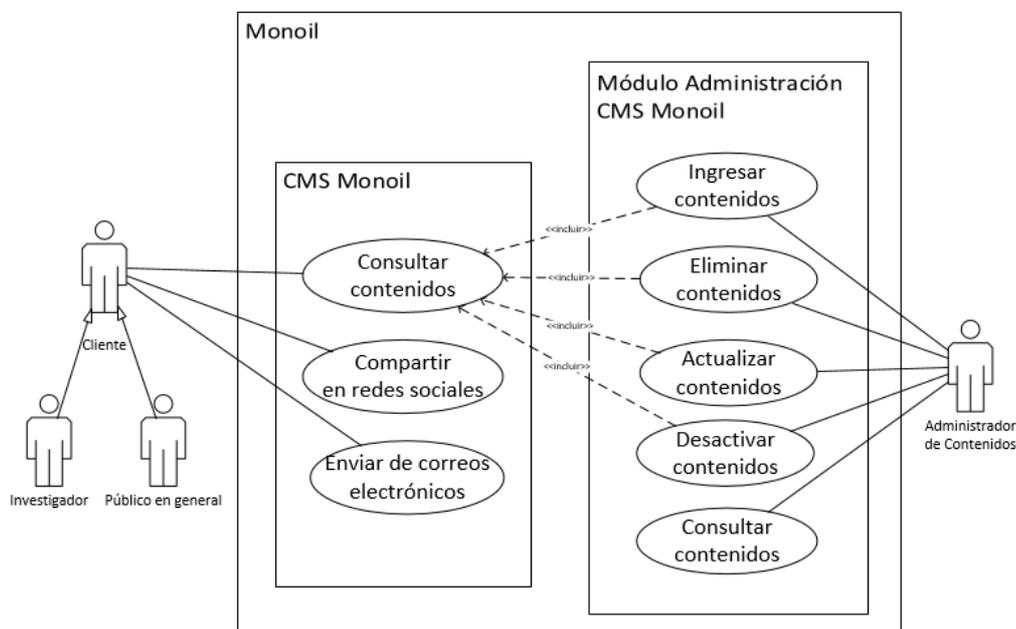
En esta categoría se concentra el uso del CMS actual que gestiona los contenidos del Sitio Monoil. Al no existir un CMS exclusivo para Monoil, se optó por utilizar una solución del mercado, pero al desconocer su correcto uso, no se actualizan los contenidos y muchas veces se ubican en un diseño poco agradable.

### 4. Categoría “Método de trabajo”.-

En esta categoría se encuentra la carencia de procesos dentro de Monoil para gestionar contenidos. Monoil, para administrar su sitio web informativo, no cuenta con metodologías o procesos de actualización de contenidos, lo que implica no tener períodos de administración de información. Al trabajar bajo esta libertad de trabajo en ocasiones se presentan contenidos desactualizados y de poco interés para los usuarios.

#### 4.2.1 DIAGRAMAS DE CASOS DE USO

El Diagrama de Casos de Uso nos ayuda a explicar gráficamente los requerimientos y el alcance que tiene la solución CMS Monoil para abordar o solucionar los diferentes problemas a los que se enfrenta el proyecto binacional. [19] Se recuerda que la solución consta de dos partes: 1. Módulo de administración del CMS Monoil; 2. CMS Monoil.



**Fig. 3. Diagrama de Casos de Uso - Monoil**

Entre los actores de sistema tenemos:

- Cliente, el cuál puede ser un investigador de Monoil o cualquier persona que desea obtener información del sitio.
- Administrador de contenidos, quien gestiona la información del sitio a través del MACMS Monoil.

Casos de Uso del CMS Monoil relacionados directamente al cliente:

- Consultar contenidos.- Es el proceso para consultar los contenidos de la base de datos.
- Compartir en redes sociales.- Es el proceso para compartir la información del sitio en las redes sociales.
- Enviar correos electrónicos.- Es el proceso para contactarse con el administrador de contenidos.
- Acceder a la plataforma Monoil.- Es el proceso para acceder a la plataforma principal de Monoil.

Casos de Uso del MACMS Monoil relacionados directamente al administrador de contenidos:

- Ingresar contenidos.- Es el proceso para ingresar contenidos a la base de datos.
- Eliminar contenidos.- Es el proceso para eliminar lógicamente los contenidos de la base de datos.
- Actualizar contenidos.- Es el proceso para actualizar los contenidos de la base de datos.
- Desactivar contenidos.- Es el proceso para desactivar los contenidos de la base de datos.
- Consultar contenidos.- Es el proceso para consultar los contenidos de la base de datos.

### 4.3 METODOLOGÍA DE DESARROLLO SCRUM

SCRUM es una metodología ágil enfocada a productos o servicios, en este caso orientada al desarrollo de software, basada en un modelo donde el principal objetivo es obtener cada uno de los requerimientos o características principales del sistema y demás funcionalidades. Se deben ir creando partes del sistema de mayor significado para el cliente con la finalidad de añadirle valor al producto final; en caso de no cumplir con los tiempos estimados, se tendrán características o funcionalidades esenciales del producto final y solicita una prórroga para las tareas menos complejas de desarrollar. [20]

Dentro de la implementación de esta metodología en el proyecto Binacional Monoil, se realizan cada uno de los procesos y prácticas involucrando a los diferentes integrantes con sus respectivos roles. Donde el Scrum Máster es el Ing. Christian Antón, el Product Owner lo conforman los directivos del proyecto Monoil, y el equipo de desarrollo lo conforman cada uno de los integrantes del Grupo Monoil de la Universidad de Guayaquil.

Cumpliendo con las diferentes etapas de la metodología, quienes estaban a cargo del proyecto, realizaron reuniones con cada uno de los aspirantes a conformar el equipo de desarrollo esta etapa se denomina “Exposición de prioridades, Resolución de dudas”. En cada una de estas reuniones fueron despejadas dudas como sitios de trabajo, disponibilidad de recursos, horarios de trabajo, entre otros temas importantes para cumplir con los objetivos del proyecto. En esta sección el Product Owner ya cuenta con la “Pila del Producto”, o la lista de las historias de usuario.

En el proceso o actividad denominada “Objeto del Sprint”, se asignaron cada una de las tareas a los diferentes recursos. Estas actividades conllevan esfuerzo, uso de ciertos recursos de hardware y software, por lo que el presente artículo recae en la aplicación CMS Monoil, que se encuentra en el **Sprint número 9 del proyecto binacional** el cual tenía la tarea asignada de cumplir con el desarrollo de la “Aplicación CMS Monoil destinada al diseño, presentación y administración de contenidos relacionados al proyecto Monoil”.

Para desarrollar la presente propuesta tecnológica, se ha escogido la metodología SCRUM para hacer una subdivisión de las actividades, debido a que se basa en procesos de desarrollo ágiles, éste es susceptible a cambios. El presente artículo se refiere al desarrollo del CMS Monoil, por lo que, se detallan únicamente las actividades necesarias para alcanzar el objetivo de desarrollar de la aplicación requerida. A continuación en la Tabla 4, se listan cada uno de los Sprint, los cuales se cumplen en los tiempos estimados.

**Tabla 4. Sprints y entregables de SCRUM**

No.	Tiempo (días)	Actividades
1	7	Definición de pantallas para cada uno de los módulos asignados.
2	9	Análisis y Diseño de la base de datos en base a los requerimientos
3	20	Desarrollo del módulo mantenimiento de tablas de alto nivel de jerarquía del sistema.
4	18	Desarrollo del módulo de mantenimiento de tablas de medio y bajo nivel de jerarquía del sistema.
5	20	Desarrollo del CMS Monoil.
6	5	Pruebas SonarQube
7	7	Mejoras post pruebas
8	8	Manual técnico y de usuario para módulo de Mantenimiento y CMS Monoil.

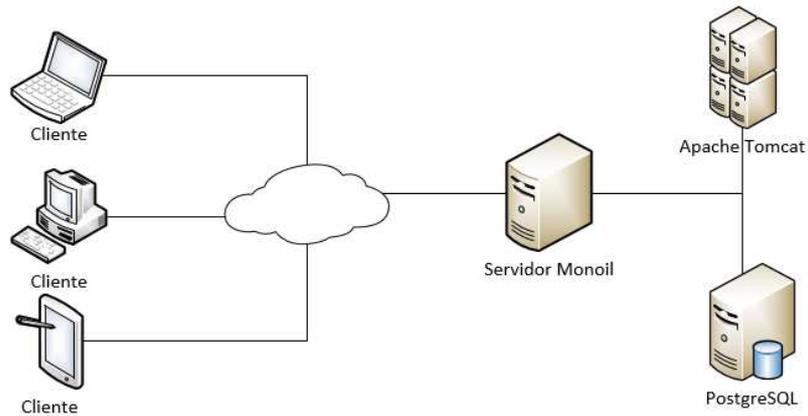
Para determinar el nivel de cumplimiento de cada Sprint, se validan las tareas que lo componen, es decir, cada tarea tiene un tiempo estimado, el cual permite obtener el *cálculo del porcentaje del nivel de cumplimiento del Sprint*. Si al finalizar el período establecido para cada Sprint no se cumplen o no pueden ser medidas todas las tareas, el Scrum Master hace el *cálculo del porcentaje de cumplimiento de las tareas del Sprint*, efectúa el análisis para detectar posibles causas de los impedimentos y una vez obtenida la información antes mencionada, se llega a un acuerdo con el Product Owner asignando una prórroga para cumplir al 100% con las actividades y el Sprint.

Para obtener el *Cálculo del porcentaje de cumplimiento del Sprint*, es necesario conocer el tiempo estimado del Sprint y el de cada una de sus tareas. Una vez que se obtengan estos datos, se realiza una suma del tiempo de las tareas que se han cumplido en su totalidad, luego se realiza la siguiente operación.

$$\% \text{ de cumplimiento} = \frac{(\text{tiempo de tareas cumplidas}) * 100}{(\text{tiempo estimado del Sprint})}$$

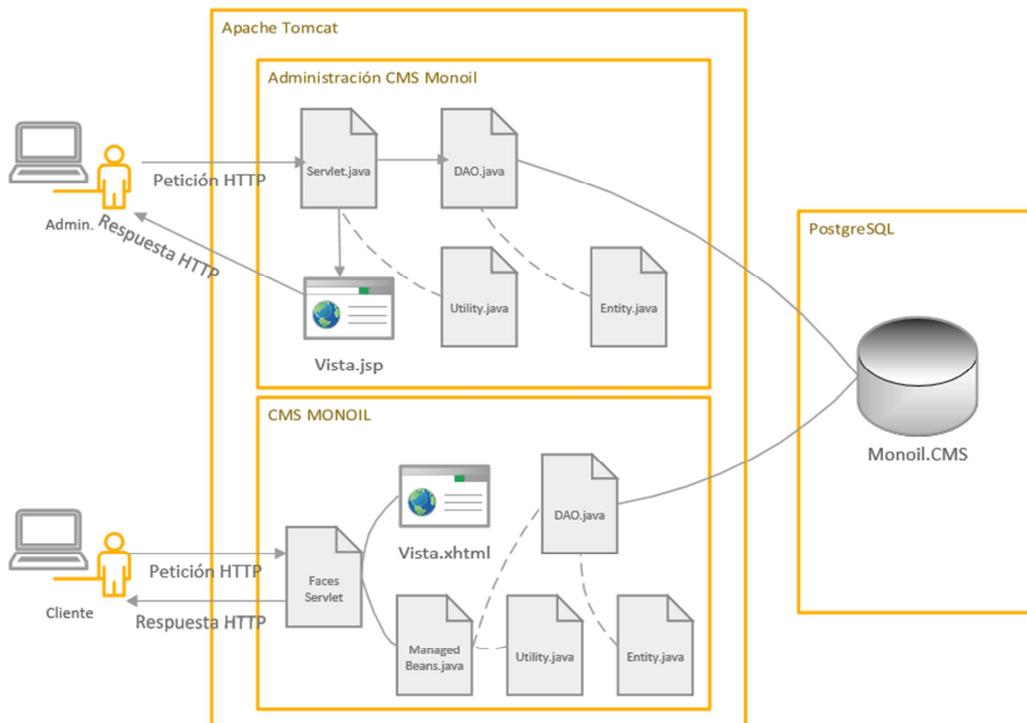
#### 4.3.1 ARQUITECTURA DEL SISTEMA

La arquitectura general diseñada para el proyecto Monoil que se muestra en la **Fig. 4**. Esta arquitectura es implementada tanto para aplicación CMS Monoil, como para el Módulo de Administración del CMS Monoil. Dentro de esta arquitectura encontramos tecnologías antes mencionadas como Apache Tomcat y PostgreSQL, los cuales brindan información a través de Internet gracias a la aplicación Monoil que se pone a disposición en el servidor de aplicaciones.



**Fig. 4. Arquitectura General de Monoil**

En la **Fig. 5** se detalla la arquitectura tanto del Módulo de mantenimiento del CMS (tecnología JSP), como la aplicación CMS Monoil (Framework JSF) a manera más detallada dentro del servidor de aplicaciones. Las arquitecturas varían debido a las tecnologías utilizadas, pero, ambos sistemas se encuentran desplegados en el servidor Apache Tomcat y trabajan conjuntamente haciendo referencia a la base de datos nombrada Monoil, en el esquema CMS. Existen 2 tipos de usuarios: el administrador de contenidos en el MACMS Monoil y cualquier persona que acceda directamente al sitio informativo del CMS.



**Fig. 5. Arquitectura CMS Monoil - Monoil**

A continuación se explica la funcionalidad de cada una de las clases presentadas dentro de la arquitectura de las aplicaciones Web desplegadas en el Servidor Monoil. El usuario es quien constantemente envía peticiones HTTP al servidor y estos son interpretados por los elementos controladores; una vez realizado todo el proceso necesario, existe una respuesta de tipo HTTP por parte del servidor el cual permite visualizar las páginas web al cliente.

**Tabla 5. Elementos dentro de la arquitectura del Proyecto**

<b>Elemento</b>	<b>Función</b>
<b>Servlet</b>	Las clases Servlet, son archivos java que sirven de controladores para determinar cuál es la petición del cliente y así guiar hacia el flujo de procesos correspondiente.
<b>DAO</b>	Las clases DAO por sus siglas Data Access Object que en español significa Objeto de Acceso de Datos, son clases java que sirven para interactuar directamente con la base de datos. En ocasiones este tipo de clases son también conocidas como modelos. Esta capa se encargará sólo de las transacciones a niveles de base de datos, por lo que la lógica no será detallada en esos archivos.
<b>Utility</b>	Las clases Utility o también llamadas utilidades, son archivos java que se crean con el objetivo de agrupar funciones genéricas o que siempre se utilizan; de esta forma son siempre utilizadas por otros segmentos de código, para así no repetir bloques de programación de manera innecesaria.
<b>Entity</b>	Las clases Entity o también llamadas entidades, son archivos java que tienen como objetivo mapear o representar en objetos las tablas del modelo relacional de la base de datos Monoil.
<b>JSP</b>	Las vistas JSP son archivos que permiten el diseño y programación de páginas web dinámicas, para la implementación del MACMS Monoil, se utilizó la biblioteca JSTL.
<b>Faces Servlet</b>	El archivo Faces Servlet se encarga de captar las peticiones y enviar respuestas hacia el cliente; este archivo es creado y manejado por el Framework JSF en la versión 2.1 por lo que no existe mayor desarrollo de este elemento dentro de la metodología de desarrollo.
<b>Managed Beans</b>	Las clases ManagedBean son archivos de java que trabajan con el Framework JSF, estos cumplen la funcionalidad de un controlador. Además las instancias de estas clases permiten obtener los valores del contexto de manera más fácil y rápida, agilizando el trabajo de presentación de contenidos en el CMS Monoil.
<b>XHTML</b>	Las vistas XHTML son archivos para desarrollar páginas web dinámicas bajo el núcleo de JSF; permiten relacionarse directamente con los ManagedBeans que se encuentran en el contexto de la aplicación CMS Monoil.

#### **4.3.2 MODELO ENTIDAD RELACIÓN**

A continuación en la **Fig. 6**, se presenta el diseño de la base de datos del esquema CMS con cada una de sus relaciones. El primer campo de cada tabla hace referencia a la clave primaria y los campos que demuestran relación son las claves foráneas. Dentro de las tablas más importantes encontramos: "Menu", "Pagina", "PaginaContenido", "PaginaContenidoImagen" que permiten generar los contenidos estructuradamente para que el usuario tenga una buena experiencia visual e informativa.

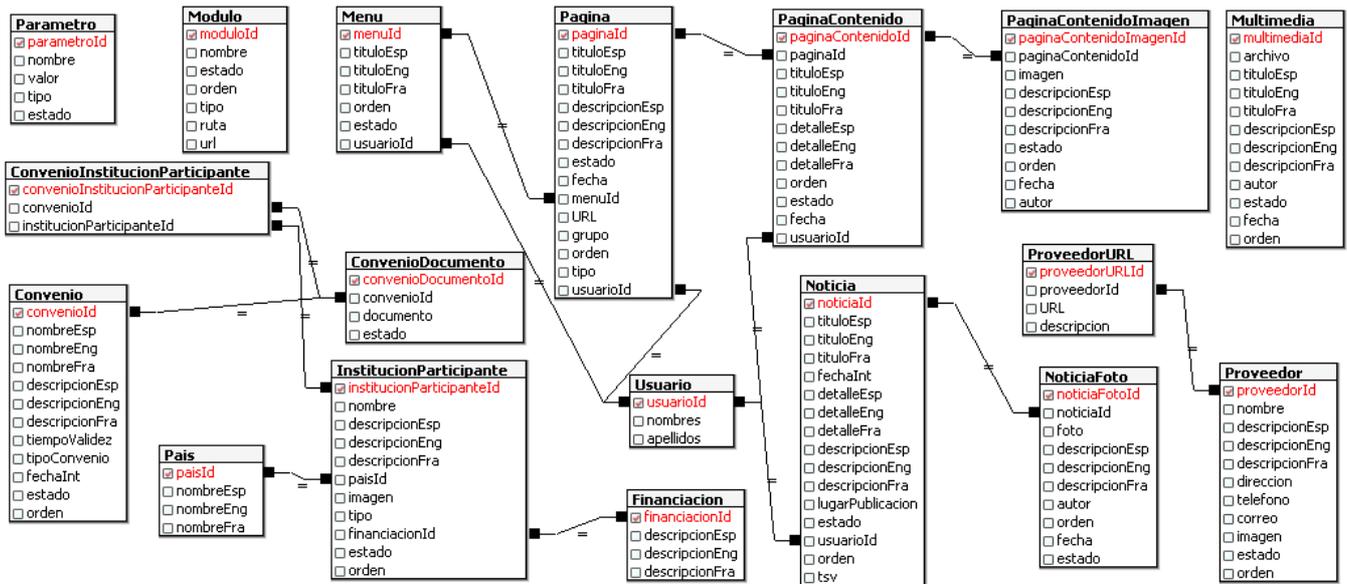


Fig. 6. Modelo Entidad Relaci3n Monoil.CMS

## 5 RESULTADOS

Una vez que finalizaron cada una de las etapas de la metodolog3a, el Scrum M3ster es el encargado de revisar detenidamente cada uno de los entregables y posteriormente aprobar o sugerir cambios del producto tecnol3gico. El presente fue aprobado debido a que se cumpli3 con lo esperado en los tiempos estimados. A continuaci3n se muestra una pantalla de una de las funcionalidades m3s importantes del MACMS Monoil que es la pantalla que permite el ingreso de los contenidos.

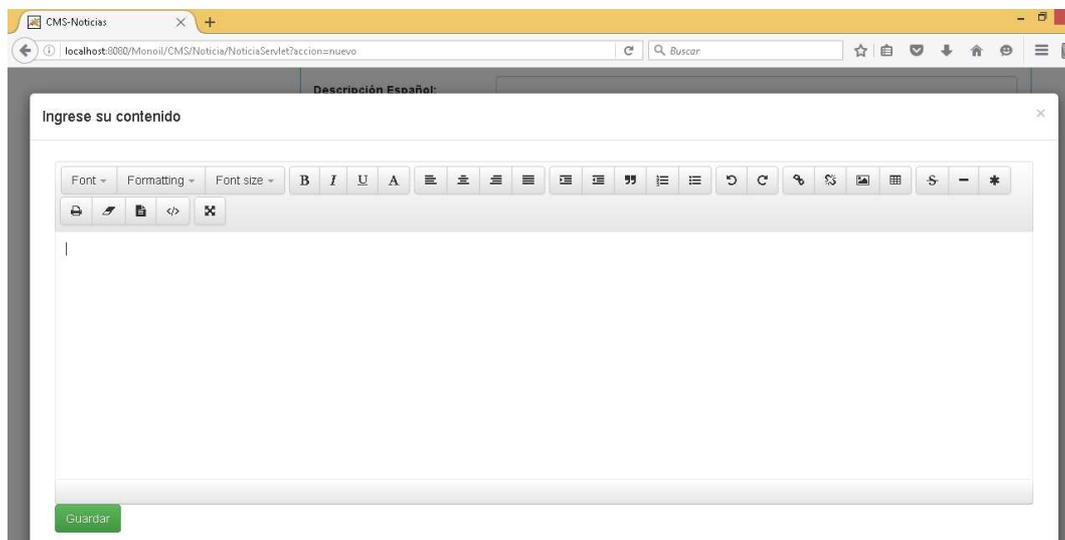


Fig. 7. Editor de contenidos

Una vez ingresada la informaci3n correspondiente a un contenido de una p3gina web creada por el CMS Monoil, se procede a revisar las actualizaciones que realiza el sistema de manera inmediata. En la Fig. 8 se muestra el resultado de una p3gina web que extrae los contenidos de la base de datos Monoil. La cual consta de una secci3n destinada para los contenidos y una adicional para adjuntar im3genes relacionadas con la informaci3n. Adem3s el CMS Monoil, puede ser accedido a trav3s de dispositivos m3viles sin alterar o malograr el dise1o del sitio. Dentro del m3dulo de actualidades, los usuarios pueden compartir la informaci3n en sus redes sociales. Finalmente, existen secciones personalizadas como por ejemplo el m3dulo de convenios que extrae informaci3n de su tabla especifica y adem3s presenta los documentos que convalidan los acuerdos de Monoil.

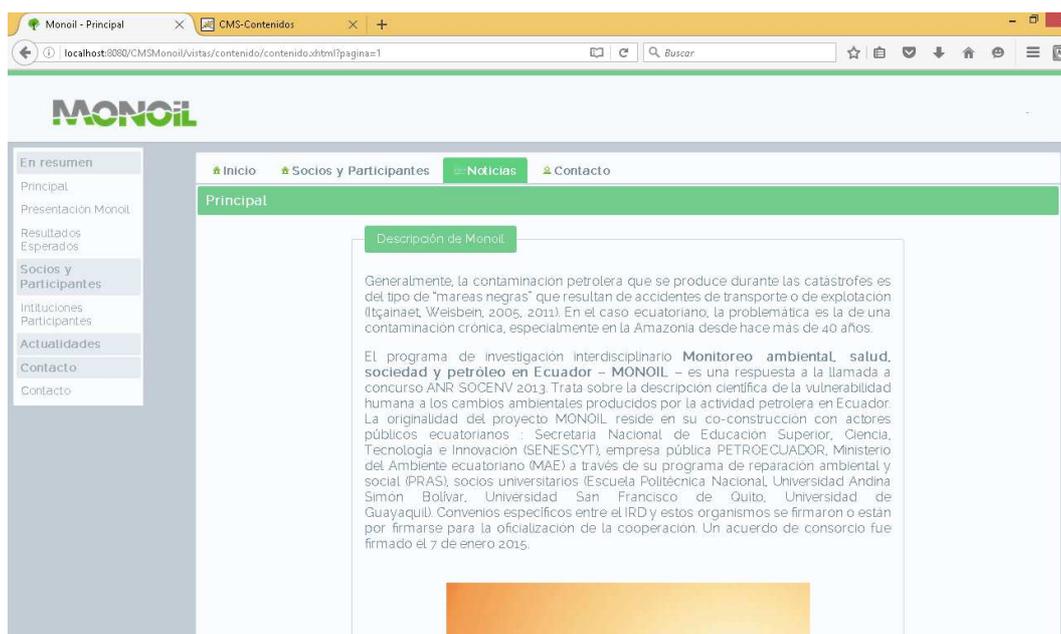


Fig. 8. Resultado del CMS Monoil

## 6 CONCLUSIONES

Los Administradores de Contenido pueden clasificarse en varios tipos, dependiendo de los requerimientos que éstos abarquen. Comúnmente cuando se refiere a un CMS, se trata de un Administrador de Contenidos Web WCMS. Monoil requiere de un CMS personalizado a sus necesidades, por lo que se creó el CMS Monoil, con la finalidad de crear páginas web sin tener mayor conocimiento en el desarrollo de aplicaciones web y mantener su sitio informativo actualizado constantemente en beneficio de los investigadores y el público en general.

Las características del CMS Monoil se encuentran en su base de datos relacional; aquí es donde se almacenan las páginas web con sus respectivos contenidos e imágenes. La base de datos Monoil fue creada en base al conjunto de proyectos de la solución final Monoil y el esquema CMS, responde a la problemática del presente artículo.

Para evitar la obligación de contar con conocimientos web, se desarrolló un módulo de mantenimiento orientado a las tablas de Monoil, para poder gestionar la información existente en la base de datos y mejorar los procesos evitando la curva de aprendizaje para un CMS complejo.

La aplicación CMS Monoil responde a la problemática de presentar contenidos web dinámicos y actualizados gracias a la integración de su base de datos y su módulo de mantenimiento. Este sistema con su módulo, forman parte de la solución final Monoil el cual se basa en la metodología de desarrollo ágil SCRUM, la cual permite desarrollar grandes proyectos de forma ágil.

## REFERENCIAS

- [1] M. J. Vidal Ledo and A. B. Araña Pérez, "Gestión de la información y el conocimiento," *Educación Médica Superior*, vol. 26 no. 3, pp. 474-484, 2012.
- [2] Y. Rosell León, (2011). "Estudio de los Sistemas Gestores de Contenido: Una mirada a la productividad en el campo desde el Web de la Ciencia," *Ciencias de la Información*, vol. 42 no. 2, pp. 25-34, 2011.
- [3] Centro de Apoyo Tecnológico a Emprendedores, Fundación Parque Científico y Tecnológico de Albacete. *Estudio de los sistemas de gestión de contenido web*. Creative Commons By - Sa., 2012
- [4] M. R. Osasuna Alarcón and E. De la Cruz Gómez, "Los sistemas de gestión de contenidos en Información y Documentación," *Revista General de Información y Documentación*, vol. 20, pp. 67-100, 2010.
- [5] Y. Carballo Pérez, "La gestión de contenidos en portales Web," *Acimed*, vol. 15, no.3, 2007.
- [6] M. Iglesias-García, C. González-Díaz and A. Navalón-García, "Cambio de plataforma de edición del ciberperiódico Comunic@ndo", *Universidad de Alicante*, 2014.

- [7] D. Rodríguez Hernández, A. Rodríguez Pérez and E. Díaz Martínez, "Selección de Base de Datos No SQL para almacenamiento de Históricos en Sistemas de Supervisión," *Revista Cubana de Ciencias Informáticas*, vol. 10, no. 3 pp. 85-96, 2016.
- [8] Java, *What is Java technology and why do I need it?*, 2016.  
[Online] Available: [https://java.com/en/download/faq/whatis\\_java.xml](https://java.com/en/download/faq/whatis_java.xml) (July 21, 2016)
- [9] Oracle, *The Java EE 6 Tutorial: Oracle*, 2013
- [10] J. Ramos, "Java Platform Enterprise Edition". En L. Pino (Dirección Ejecutiva de la Unidad de Educación Continua), *Java Persistence API*. Capacitación dirigida por la Universidad de Especialidades Espiritu Santo. (May, 2016)
- [11] L. Peñaherrera, "Java Platform Enterprise Edition". En L. Pino (Dirección Ejecutiva de la Unidad de Educación Continua), *Enterprise JavaBeans*. Capacitación dirigida por la Universidad de Especialidades Espiritu Santo. (June, 2016).
- [12] NetBeans, *NetBeans IDE – Overview*, 2016. [Online] Available: <https://netbeans.org/features/> (July 27, 2016)
- [13] The Apache Software Foundation, *Apache Tomcat*, 2016 [Online] Available: [tomcat.apache.org](http://tomcat.apache.org) (July 23, 2016)
- [14] The PostgreSQL Global Development Group, *Tutorial de PostgreSQL* (Vol. 8.4), 2014.  
[Online] Available: <http://www.postgresql.org/docs/8.4/interactive/index.html> (July 29, 2016)
- [15] Bootstrap, *Bootstrap - The world's most popular mobile-first and responsive front-end framework*, 2016. [Online] Available: [getbootstrap.com](http://getbootstrap.com) (July 28, 2016)
- [16] Oracle, *JavaServer Faces Specification*, 2014. [Online] Available: <https://jvaserverfaces-spec-public.java.net/> (September 1, 2016)
- [17] Çivici, *Primefaces User Guide 6.0*, 1st Ed. Prime, 2016
- [18] E. Romero Bermúdez and J. Díaz Camacho, (2010). "El uso del diagrama causa-efecto en el análisis de casos," *Revista Latinoamericana de Estudios Educativos (México)*, vol. XL, no 3-4, pp. 127-142, 2010.
- [19] Universidad Católica de Murcia. [portalucam], *Ingeniería del Software II - Modelado de requisitos con UML (Casos de Uso) - Fernando Pereñiguez*, 25 de noviembre de 2013.  
[Video]. Available: <https://www.youtube.com/watch?v=5ezWOj0k02k> (September 10, 2016)
- [20] Rubin, *Essential Scrum*. Pearson Education, Inc., 2013.