

UNIVERSIDAD POLITÉCNICA DE PUEBLA
Ingeniería en Informática



Proyecto de Estadía Profesional

“Generador De Minisitios”

Área temática del CONACYT: VII
Ingenierías y tecnologías

Presenta:

Uriel Castillo Kuri

Asesor técnico

Ing. Rigoberto Martínez López

Asesor académico

M.C Rebeca Rodríguez Huesca

Juan C. Bonilla, Puebla, México.

30/abril/2021

Resumen

El proyecto tiene como propósito poner en práctica los conocimientos que se obtuvieron durante la carrera de Informática las cuales nos ayudan a mejorar y agilizar el desarrollo de un proyecto.

La organización consultoría contracorriente S.A. de C.V. requiere un sistema que pueda brindar ayuda a periodistas y administradores a la gestión de contenidos para los diferentes minisitios que se lleguen a manejar. además de agrupar información de acuerdo a tópicos y así como de presentar la información en un portal con identidad propia y estructurada, es por esto que se tomó la decisión de desarrollar un sistema que permita la ejecución de estas actividades, para esto se utilizó la metodología de prototipado, la cual permite que el usuario pueda visualizar la versión del sistema y pueda dar una retroalimentación para la mejora del sistema, además de ayudar al desarrollador a cumplir las necesidades del usuario, el uso de herramientas es fundamental para el desarrollo del sistema por lo cual se optó por el uso del modelo vista-controlador para utilizar codeigniter como framework, mysql para la base de datos, php y html 5 como lenguajes de desarrollo. Finalmente se realizarán pruebas de usabilidad al sistema para asegurar al usuario que funciona correctamente y cumple con las actividades solicitadas.

Índice

1. Introducción	6
1.1 Descripción del problema o necesidad	6
1.2 Justificación	6
1.3 Objetivo General y Específicos	6
2. Metodología y herramientas	7
2.1 Metodología “Prototipado”	7
2.2 Modelo Vista Controlador (MVC)	10
2.3 CodeIgniter 4	11
2.4 PHP	12
2.5 MYSQL	13
2.6 HTML 5	14
3. Resultados	15
3.1 Definición de los requisitos	15
3.2 Análisis y diseño	20
3.3 Codificación	23
3.4 Pruebas	34
3.5 Mantenimiento	39
4. Conclusiones y recomendaciones	41
5. Referencias bibliográficas	42

Índice de figuras

FIGURA 1 ETAPAS DE LA METODOLOGÍA	9
FIGURA 2. FUNCIONAMIENTO DEL MVC	10
FIGURA 3. CASO DE USO GENERAL DEL SISTEMA.....	15
FIGURA 4. CASO DE USO PARA USUARIOS.....	15
FIGURA 5. CASOS DE USO AUTORES	16
FIGURA 6. CASO DE USO CATEGORÍAS.....	17
FIGURA 7. CASO DE USO SITIOS.....	18
FIGURA 8. CASOS DE USO CONTENIDO.....	19
FIGURA 9. BASE DE DATOS.....	21
FIGURA 10. REPRESENTACIÓN INTERFAZ PARA VISUALIZAR CONTENIDOS.....	22
FIGURA 11. REPRESENTACIÓN DE INTERFAZ PARA INGRESAR Y EDITAR CONTENIDOS.....	22
FIGURA 12. REPRESENTACIÓN DE INTERFAZ PARA INGRESAR NUEVO CONTENIDO.....	23
FIGURA 13. INICIO DE SESIÓN VERSIÓN 1.....	23
FIGURA 14. CODIFICACIÓN DE LOGIN.....	24
FIGURA 15. VISTA DE TODOS LOS USUARIOS REGISTRADOS.....	24
FIGURA 16. CODIFICACIÓN VISTA DE TODOS LOS USUARIOS REGISTRADOS.....	25
FIGURA 17. VISTA NUEVO USUARIO.....	25
FIGURA 18. CODIFICACIÓN VISTA NUEVO USUARIO.....	26
FIGURA 19. VISTA EDICIÓN DE USUARIOS.....	26
FIGURA 20. CODIFICACIÓN VISTA EDICIÓN DE USUARIO.....	27
FIGURA 21. VISTA AL INTENTAR ELIMINAR UN REGISTRO.....	27
FIGURA 22. CODIFICACIÓN VISTA ELIMINACIÓN DE USUARIO.....	28
FIGURA 23. VISTA AUTORES REGISTRADOS.....	28
FIGURA 24. VISTA NUEVO AUTOR.....	29
FIGURA 25. VISTA EDITAR AUTOR.....	29
FIGURA 26. VISTA ELIMINAR AUTOR.....	30
FIGURA 27. VISTA DE CATEGORÍAS REGISTRADAS.....	30
FIGURA 28. VISTA AGREGAR NUEVA CATEGORÍA.....	31
FIGURA 29. EDITAR CATEGORÍA.....	31
FIGURA 30. VISTA ELIMINAR CATEGORÍA.....	32
FIGURA 31. VISTA SITIOS REGISTRADOS.....	32
FIGURA 32. VISTA NUEVO SITIO.....	33
FIGURA 33. VISTA EDITAR SITIO.....	33
FIGURA 34. VISTA ELIMINAR SITIO.....	34
FIGURA 35. PRUEBA USABILIDAD 1.....	35
FIGURA 36. PRUEBA USABILIDAD 2.....	35
FIGURA 37. PRUEBA USABILIDAD 3.....	36
FIGURA 38. RESULTADOS GENERALES DE LAS PRUEBAS.....	38
FIGURA 39. PRIMERA VERSIÓN PARA INGRESAR CONTENIDO NUEVO.....	38
FIGURA 40. VISTA DE CONTENIDOS INGRESADOS.....	39
FIGURA 41. SEGUNDA VERSIÓN EN EL APARTADO DE CONTENIDO.....	39
FIGURA 42. SEGUNDA VERSIÓN EN EL INICIO DE SESIÓN.....	40

Índice de Tablas

TABLA 1. CASO DE USO USUARIOS	16
TABLA 2. CASO DE USO AUTORES.....	17
TABLA 3. CASO DE USO CATEGORÍAS	18
TABLA 4. CASO DE USO SITIO.....	19
TABLA 5. CASO DE USO CONTENIDOS.	20
TABLA 6. ACTIVIDADES A REALIZAR	20
TABLA 7. EVALUACIÓN DE USABILIDAD CP-01	36
TABLA 8. EVALUACIÓN DE USABILIDAD CP-02.	37
TABLA 9. EVALUACIÓN DE USABILIDAD CP-03.	37

1. Introducción

En este capítulo se hablará de cuál es la necesidad que tiene la Consultoría contracorriente S.A. de C.V. y cómo se puede solucionar.

1.1. Descripción del problema o necesidad

Debido a la reciente situación de la pandemia y la gran cantidad de información que se genera alrededor de este tema, en el medio periódico digital surge la necesidad de agrupar información de acuerdo a tópicos, así como de presentar la información en un portal con identidad propia y estructura. De ahí la necesidad de crear un generador de minisitios, que agrupe información para los tópicos que surjan y tengan una administración centralizada.

1.2 Justificación

El generador de minisitios es la solución adecuada para la empresa, ya que con una programación ajustada a las necesidades y que la administración esté en una misma plataforma, da pie a que un sólo webmaster pueda manejar la información, así como una vez implantada la solución se pueda dar mantenimiento y escalar el proyecto de una manera más controlada y sencilla.

1.3 Objetivo General y Específicos

Objetivo General:

Desarrollar una plataforma que permita generar minisitios así como toda su administración y configuración.

Objetivos específicos:

- Realizar análisis sobre la propuesta del sistema.
- Realizar el diseño de base de datos y prototipos del sistema.
- Implementar el sistema con base al diseño previamente realizado.
- Realizar pruebas de usabilidad.

2. Metodología y herramientas

En este segundo capítulo se presentará una aproximación al proceso metodológico y con los conceptos básicos que se han aplicado a lo largo de este proyecto, así también como a las herramientas que se usaran.

2.1 Metodología: Prototipado.

[1] Pertenece a la rama de los modelos evolutivos tales como el modelo incremental, en este modelo se hace el uso de prototipos los cuales represe tan partes esenciales del software, sin embargo, a diferencia del modelo incremental los prototipos no son funcionales y en su mayoría únicamente corresponden a representaciones del diseño visible para el usuario y no del diseño interno tales como algoritmos y manejo de la Información, a al desarrollador una mejor comprensión de lo que hay que hacer.

Su característica principal es la retroalimentación, este modelo permite crear una muestra de la solución, para comprobar si será funcional, esta se consulta con el cliente y en base a la opinión del cliente, se hacen las modificaciones necesarias hasta llegar a un modelo final que satisfaga las necesidades del cliente en cuestión. Sus ventajas principales son el menor fallo de errores ya que se encuentra con una constante comunicación con el cliente. Además de que es una excelente herramienta para la recolección de requisitos del cliente. Que hacer, a lo largo de todo el ciclo de vida del software, para construir un producto bueno, de calidad, dentro del presupuesto y a tiempo. La utilización de técnicas para un desarrollo de software de calidad y que cumpla con todos los requerimientos que el cliente solicitó.

El Modelo de prototipos, en Ingeniería de software, pertenece a los modelos de desarrollo evolutivo. El prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar muchos recursos.

El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final. Este diseño conduce a la construcción de un prototipo, el cual es evaluado por el cliente para una retroalimentación. [2]

El paradigma de creación de prototipos puede ser cerrado o abierto. El enfoque cerrado se denomina a menudo prototipo desechable. Este prototipo sirve únicamente como una basta demostración de los requisitos. Después se desecha y se hace una ingeniería del software con un paradigma diferente. Un enfoque abierto, denominado prototipo evolutivo, emplea el prototipo como primera parte de una actividad de análisis a la que seguirá el diseño y la construcción. El prototipo del software es la primera evolución del sistema terminado.

Además de los prototipos cerrados y abiertos existen modelos de prototipo que podrían adaptarse mejor a cada proyecto, por ejemplo.

- A. Modelo de Prototipos rápido: Metodología de diseño que desarrolla rápidamente nuevos diseños, los evalúa y prescinde del prototipo cuando el próximo diseño es desarrollado mediante un nuevo prototipo.
- B. Modelo de Prototipos reutilizable: También conocido como "Evolutionary Prototyping"; no se pierde el esfuerzo efectuado en la construcción del prototipo pues sus partes o el conjunto pueden ser utilizados para construir el producto real. Mayormente es utilizado en el desarrollo de software, si bien determinados productos de hardware pueden hacer uso del prototipo como la base del diseño de moldes en la fabricación con plásticos o en el diseño de carrocerías de automóviles.
- C. Modelo de Prototipos Modular: También conocido como Prototipado Incremental (Incremental prototyping); se añaden nuevos elementos sobre el prototipo a medida que el ciclo de diseño progresa.
- D. Modelo de Prototipos Horizontal: El prototipo cubre un amplio número de aspectos y funciones, pero la mayoría no son operativas. Resulta muy útil para evaluar el alcance del producto, pero no su uso real.
- E. Modelo de Prototipos Vertical: El prototipo cubre sólo un pequeño número de funciones operativas. Resulta muy útil para evaluar el uso real sobre una pequeña parte del producto.
- F. Modelo de Prototipos de Baja-fidelidad: El prototipo se implementa con papel y lápiz, emulando la función del producto real sin mostrar el aspecto real del mismo. Resulta muy útil para realizar tests baratos.
- G. Modelo de Prototipos de Alta-fidelidad: El prototipo se implementa de la forma más cercana posible al diseño real en términos de aspecto, impresiones, interacción y tiempo. [3]

Antes de poder elegir un enfoque es necesario determinar si se puede crear un prototipo del sistema a construir. Se pueden definir varios factores candidatos a la creación de prototipos área de aplicación, complejidad, características del cliente y del proyecto.

En el proyecto se utilizará la metodología de modelo de prototipo abierto o prototipo evolutivo ya que se basa en la idea de desarrollar basándose en los comentarios del usuario y al final refinando los detalles a través de diferentes versiones hasta que se desarrolla el sistema adecuado para el usuario.

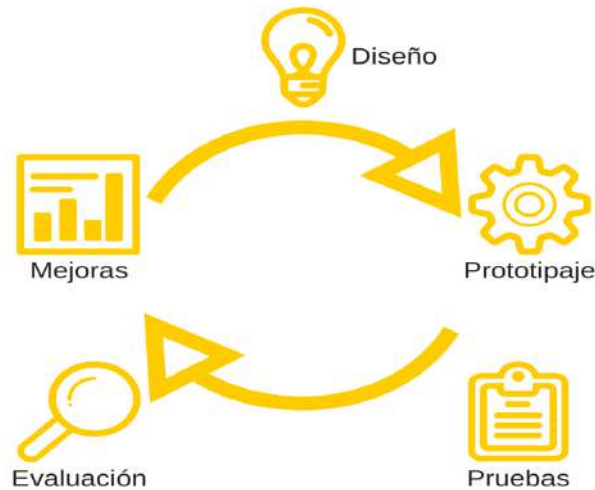


Figura 1 Etapas de la metodología

Como se muestra en la figura 1 la metodología de prototipo consta de 5 etapas como se muestran en el esquema las cuales son:

1. Definición de los requisitos (Diseño). En este proceso se identifican las necesidades y requerimientos del cliente con respecto al software.
2. Análisis y Diseño (prototipaje). En el análisis se establece la viabilidad del software desde el punto de vista técnico y económico, se planifican las actividades y el presupuesto. En el diseño del software se centra en cuatro atributos de un programa: estructura de datos, arquitectura del software, representaciones de interfaz y detalle procedimental (algoritmo).
3. Codificación (pruebas): Se traduce en forma legible por la máquina el modelo previamente diseñado. El paso de generación de código lleva a cabo esta tarea. Si lleva a cabo el diseño de una forma detallada, la generación de código se realiza mecánicamente.
4. Pruebas(evaluación). El proceso de pruebas se centra en los procesos lógico internos del software, y en los procesos externos funcionales. Se deben realizar las pruebas para detección de errores y asegurarse que las entradas definidas produzcan resultados reales que concuerden con los resultados requeridos.
5. Mantenimiento(mejoras). El software indudablemente sufrirá cambios después de ser entregado al cliente. El mantenimiento vuelve a aplicar cada una de las fases precedentes a un programa ya existente y no a uno nuevo. [4]

Modelo Vista Controlador (MVC) [5] Model-View-Controller ("MVC") es el modelo de arquitectura recomendado para aplicaciones interactivas. MVC, organiza una aplicación interactiva en tres módulos separados: uno para el modelo de la aplicación con su representación de datos y lógica de negocios, el segundo para vistas que proporcionan representación de datos y entrada de datos del usuario, y el tercero para un controlador para procesar las peticiones y controlar el flujo como se muestra en la figura 2. La mayoría de las aplicaciones web usan alguna variación del patrón de diseño MVC.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

El modelo es el responsable de:

Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento. Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor". Lleva un registro de las vistas y controladores del sistema. Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes que actualiza los datos, un temporizador que desencadena una inserción, etc.).

El controlador es responsable de: Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.), Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas.

Las vistas son responsables de: Recibir datos del modelo y los muestra al usuario como lo muestra la figura 2.

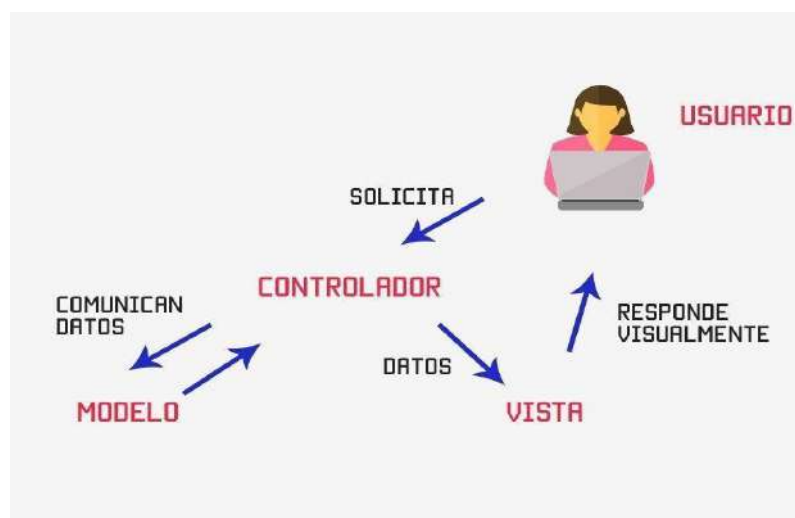


Figura 2. Funcionamiento del MVC

Herramientas

En este capítulo se mostrarán las herramientas utilizadas para el desarrollo del sistema.

2.2. Framework CodeIgniter 4. [6] Un framework es un programa para desarrollar otros programas, CodeIgniter, por tanto, es un programa o aplicación web desarrollada en PHP para la creación de cualquier tipo de aplicación web bajo PHP. Es un producto de código libre, libre de uso para cualquier aplicación.

CodeIgniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que debemos seguir para obtener provecho de la aplicación. marca una manera específica de codificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado y sea más fácil de crear y mantener. CodeIgniter implementa el proceso de desarrollo llamado Model View Controller (MVC), que es utilizado tanto para hacer sitios web como programas tradicionales.

Ventajas de CodeIgniter.

- **Versatilidad:** Quizás la característica principal de CodeIgniter, en comparación con otros frameworks PHP. CodeIgniter es capaz de trabajar la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo tenemos un acceso por FTP para enviar los archivos al servidor y donde no tenemos acceso a su configuración.
- **Compatibilidad:** CodeIgniter es compatible con la versión PHP 4, lo que hace que se pueda utilizar en cualquier servidor, incluso en algunos antiguos además funciona correctamente en PHP 5.
- **Facilidad de instalación:** No es necesario más que una cuenta de FTP para subir CodeIgniter al servidor y su configuración se realiza con apenas la edición de un archivo, donde debemos escribir cosas como el acceso a la base de datos. Durante la configuración no es necesario el acceso a herramientas como la línea de comandos, que no suelen estar disponibles en todos los alojamientos.
- **Flexibilidad:** CodeIgniter es bastante menos rígido que otros frameworks. Define una manera de trabajar específica, pero en muchos de los casos podemos seguirla o no y sus reglas de codificación muchas veces nos las podemos saltar para trabajar como más a gusto encontremos.

Algunos módulos como el uso de plantillas son totalmente opcionales. Esto ayuda muchas veces también a que la curva de aprendizaje sea más sencilla al principio.

- **Ligereza:** El núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.
- **Documentación tutorializada:** La documentación de CodeIgniter es fácil de seguir y de asimilar, porque está escrita en modo de tutorial. Esto facilita mucho la referencia rápida, cuando ya sabemos acerca del framework y se quiere consultar sobre una función o un método en concreto.[6]

Desventajas de CodeIgniter.

- Al no definir una manera estricta de trabajar puede ser difícil trabajar en equipo.
- Al poder utilizar cualquier versión de PHP pueden darse fallos de seguridad en las versiones más antiguas

2.3 PHP. [7] Es un lenguaje para programar scripts del lado del servidor, que se incrustan dentro del código HTML. Este lenguaje es gratuito y multiplataforma.

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación rápido, con una gran librería de funciones y mucha documentación.

PHP se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

Algunas ventajas son:

- Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado.
- Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes, como MySQL, mSQL, Oracle, Informix, y ODBC
- Cuenta con una amplia gama de documentación.

Desventajas:

- El inconveniente es que el código fuente no pueda ser ocultado de una manera eficiente. La ofuscación es una técnica que puede dificultar la lectura del código y, en ciertos aspectos representa tiempos de ejecución.
- El código estará seguro para ejecutar si es nuestro propio servidor. Por lo tanto, si un cliente requiere su código en su pc, tendríamos que dejar el código fuente, sin manera de ocultarlo.
- Si no se configura y/o protege correctamente quedan abiertas muchas brechas de seguridad que a la larga presentaran problemas.
- Sólo se ejecuta en un servidor y se necesita un servidor web para que funcione.

2.4 MYSQL. [8] MySQL es un sistema de gestión de bases de datos de código abierto utilizando Arquitectura Cliente y Servidor: MySQL basa su funcionamiento en un modelo cliente y servidor. Es decir, clientes y servidores se comunican entre sí de manera diferenciada para un mejor rendimiento. Cada cliente puede hacer consultas a través del sistema de registro para obtener datos, modificarlos, guardar estos cambios o establecer nuevas tablas de registros.

Ventajas:

- Compatibilidad con SQL: SQL es un lenguaje generalizado dentro de la industria. Al ser un estándar MySQL ofrece plena compatibilidad por lo que si se ha trabajado en otro motor de bases de datos no habrá problemas en migrar a MySQL.
- Procedimientos almacenados. MySQL posee la característica de no procesar las tablas directamente, sino que a través de procedimientos almacenados es posible incrementar la eficacia de implementación.
- Desencadenantes. MySQL permite además poder automatizar ciertas tareas dentro de una base de datos. En el momento que se produce un evento otro es lanzado para actualizar registros u optimizar su funcionalidad.
- Transacciones: Una transacción representa la actuación de diversas operaciones en la base de datos como un dispositivo. El sistema de base de registros avala que todos los procedimientos se establezcan correctamente o ninguna de ellas. En caso por ejemplo de una falla de energía, cuando el monitor falla u ocurre algún otro inconveniente, el sistema opta por preservar la integridad de la base de datos resguardando la información.

Desventajas:

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo.
- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- Se debe controlar/monitorizar el rendimiento de las aplicaciones en busca de fallos.

2.5 HTML5 [9] HTML, Hyper Text Markup Language (Lenguaje de marcado de Hipertexto) es el lenguaje que se utiliza para crear las páginas web. Se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán la página web e indicar a los navegadores cómo deben mostrarse estos elementos. HTML5 es la versión más nueva de HTML, y permite la publicación de archivos de audio y video.

Ventajas:

- Posee una gran cantidad de funcionalidades que los desarrolladores sólo lograban con el uso de algún plug-in de terceros como Applets de Java o Flash embebidos en el código.
- Mejoras en el manejo de multimedia son ampliamente superiores; imágenes, video, audio y fuentes de texto son mucho más manipulables.

Desventajas:

- HTML5 es tan nuevo que algunas personas que todavía usan computadoras viejas podrían usar navegadores que no pueden visualizarlo aún. [10]

3. Resultados

En este capítulo se presentan los resultados que se obtuvieron con base a la metodología usada.

3.1 Definición de los requisitos.

En las siguientes figuras se mostrarán los casos de uso que se obtuvieron en la junta con el asesor, para poder iniciar a diseñar un prototipo para el sistema.

En la figura 3 se muestra el caso de uso general del sistema.

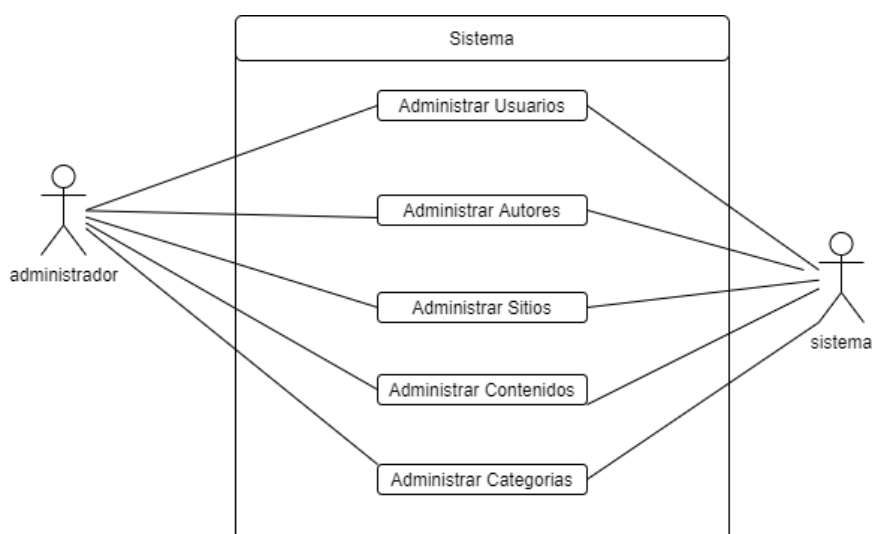


Figura 3. Caso de uso general del Sistema

En la figura 4 se muestra el caso de uso de los usuarios.

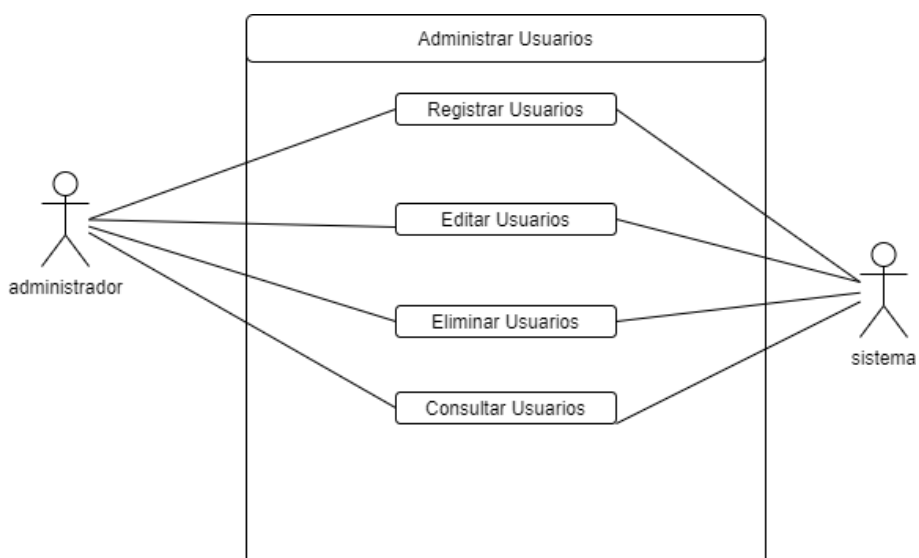


Figura 4. Caso de uso para Usuarios

En la tabla 1 se muestra la descripción paso a paso de el caso de uso para administrar usuarios

Casos de uso : Administrar Usuarios
Actores: sistema, administrador
Descripción paso a paso: 1. Registrar usuario: 1.1. El Administrador teclea los datos del nuevo usuario 1.2. El sistema guarda la información 2. Modificar usuario: 2.1. El administrador selecciona los datos que desea cambiar 2.2. El administrador teclea los nuevos datos del usuario 2.3. El sistema guarda los cambios 3. Eliminar usuario: 3.1. El administrador selecciona los datos del usuario que desea eliminar. 3.2. El administrador elimina los datos del usuario. 3.3. El sistema elimina los datos del usuario. 4. Consultar usuarios 4.1. El administrador entra en la pantalla principal de usuarios 4.2. El administrador puede navegar en la tabla que muestran a los usuarios registrados

Tabla 1. Caso de uso Usuarios

En la figura 5 se muestra el caso de uso para el apartado de autor.

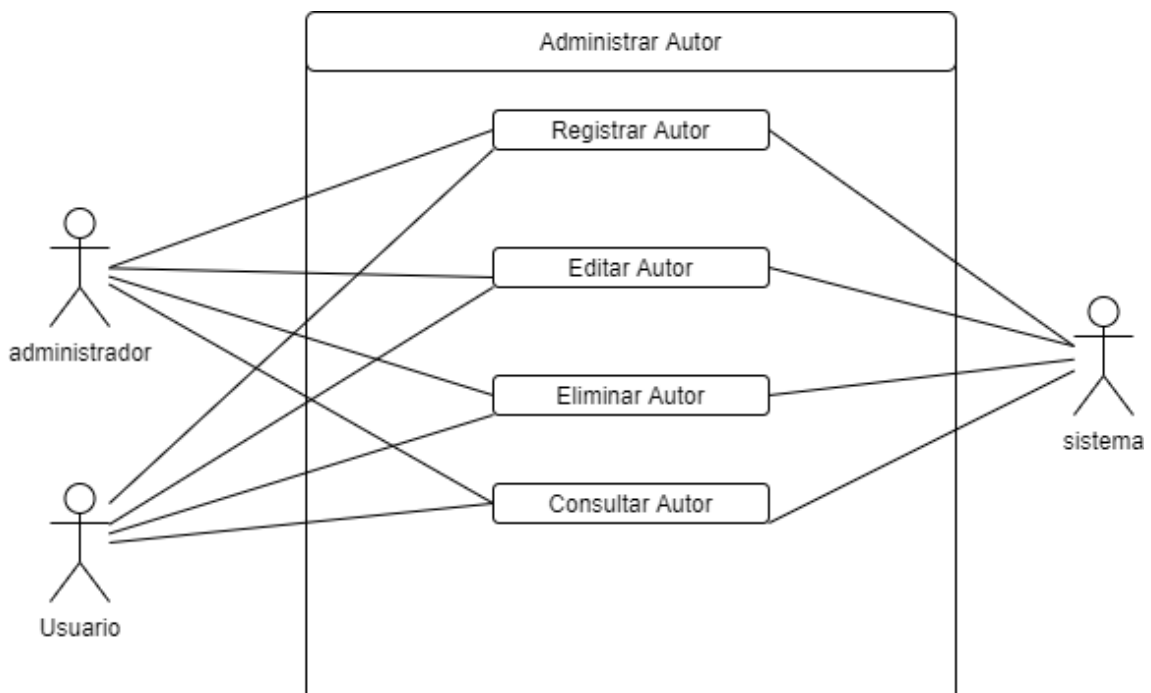


Figura 5. Casos de uso Autores

En la tabla 2 se muestra la descripción paso a paso de el caso de uso para administrar los autores.

Casos de uso : Administrar Autores
Actores: sistema, administrador, usuario.
<p>Descripción paso a paso:</p> <ol style="list-style-type: none"> 1. Registrar autor: <ol style="list-style-type: none"> 1.1. El Administrador teclea los datos del nuevo autor. 1.2. El sistema guarda la información. 2. Modificar autor: <ol style="list-style-type: none"> 2.1. El administrador selecciona los datos que desea cambiar. 2.2. El administrador teclea los nuevos datos del autor. 2.3. El sistema guarda los cambios 3. Eliminar autor: <ol style="list-style-type: none"> 3.1. El administrador selecciona los datos del autor que desea eliminar. 3.2. El administrador elimina los datos del autor. 3.3. El sistema elimina los datos del autor. 4. Consultar autores. <ol style="list-style-type: none"> 4.1. El administrador entra en la pantalla principal de autores 4.2. El administrador puede navegar en la tabla que muestran a los autores registrados.

Tabla 2. Caso de uso autores.

En la figura 6 se muestra el caso de uso para categoría.

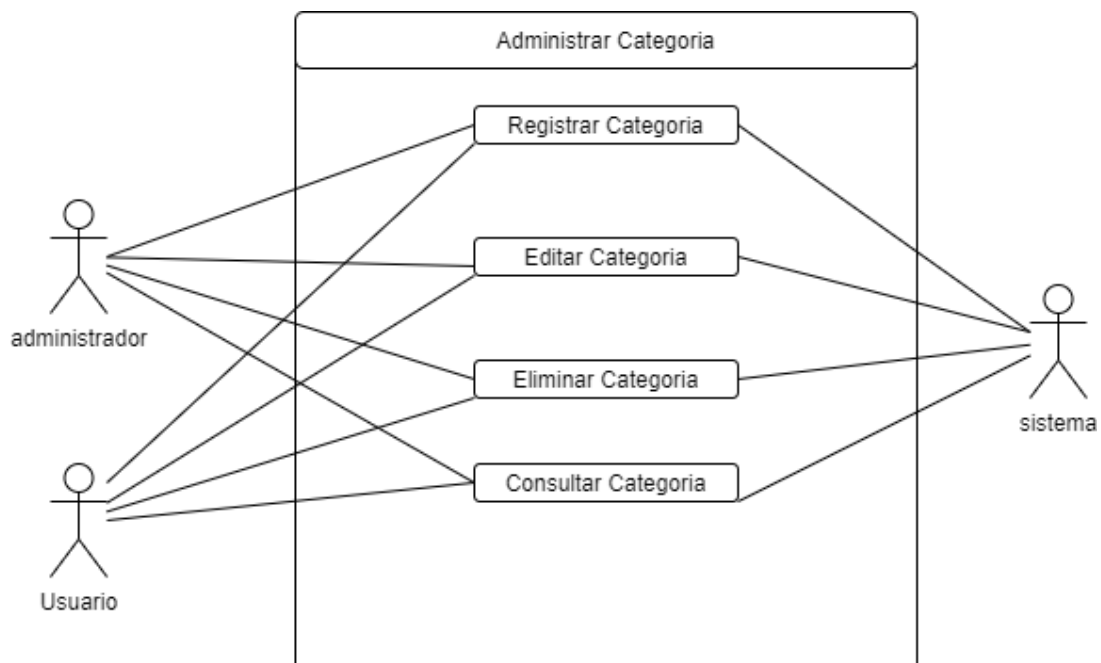


Figura 6. Caso de uso Categorías.

En la tabla 3 se muestra la descripción paso a paso de el caso de uso para administrar las categorías.

Casos de uso : Administrar Categorías
Actores: sistema, administrador, usuario.
<p>Descripción paso a paso:</p> <ol style="list-style-type: none"> 1. Registrar categoría: <ol style="list-style-type: none"> 1.1. El Administrador teclea los datos de la nueva categoría. 1.2. El sistema guarda la información. 2. Modificar categoría: <ol style="list-style-type: none"> 2.1. El administrador selecciona los datos que desea cambiar. 2.2. El administrador teclea los nuevos datos de la categoría. 2.3. El sistema guarda los cambios 3. Eliminar categoría: <ol style="list-style-type: none"> 3.1. El administrador selecciona los datos de la categoría que desea eliminar. 3.2. El administrador elimina los datos de la categoría. 3.3. El sistema elimina los datos de la categoría. 4. Consultar categorías. <ol style="list-style-type: none"> 4.1. El administrador entra en la pantalla principal de categorías 4.2. El administrador puede navegar en la tabla que muestran a las categorías registrados

Tabla 3. caso de uso Categorías

En la figura 7 se muestra el caso de uso para sitio.

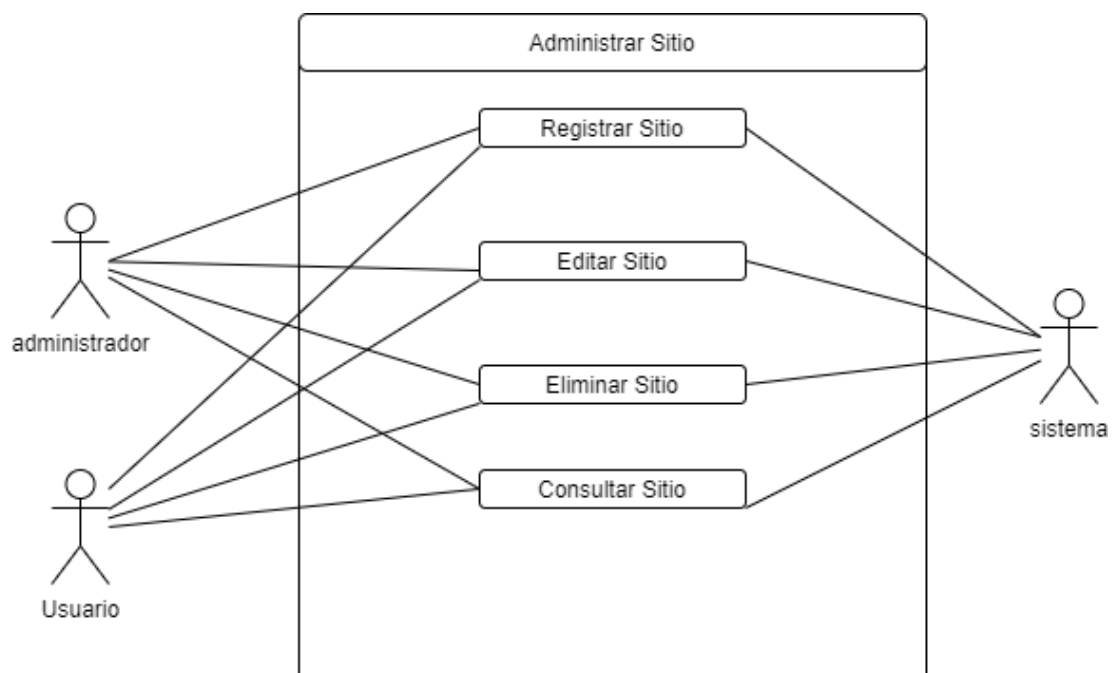


Figura 7. Caso de uso Sitios.

En la tabla 4 se muestra la descripción paso a paso de el caso de uso para administrar los sitios.

Casos de uso : Administrar Sitio
Actores: sistema, administrador, usuario.
<p>Descripción paso a paso:</p> <ol style="list-style-type: none"> 1. Registrar sitio: <ol style="list-style-type: none"> 1.1. El Administrador teclea los datos del nuevo sitio. 1.2. El sistema guarda la información. 2. Modificar sitio: <ol style="list-style-type: none"> 2.1. El administrador selecciona los datos que desea cambiar. 2.2. El administrador teclea los nuevos datos del sitio. 2.3. El sistema guarda los cambios 3. Eliminar sitio: <ol style="list-style-type: none"> 3.1. El administrador selecciona los datos del sitio que desea eliminar. 3.2. El administrador elimina los datos del sitio. 3.3. El sistema elimina los datos del sitio. 4. Consultar sitios <ol style="list-style-type: none"> 4.1. El administrador entra en la pantalla principal de sitios 4.2. El administrador puede navegar en la tabla que muestran a los sitios registrados

Tabla 4. Caso de uso Sitio.

En la figura 8 se muestra el caso de uso para el contenido

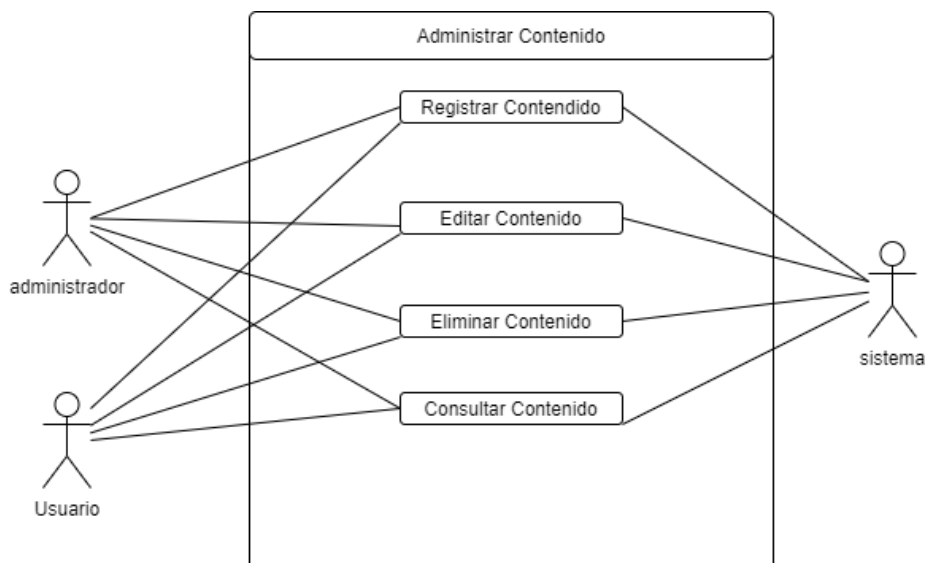


Figura 8. Casos de uso Contenido.

En la tabla 5 se muestra la descripción paso a paso de el caso de uso para administrar los contenidos.

Casos de uso : Administrar Contenidos
Actores: sistema, administrador, usuario.
<p>Descripción paso a paso:</p> <ol style="list-style-type: none"> 1. Registrar contenido: <ol style="list-style-type: none"> 1.1. El Administrador teclea los datos del nuevo contenido. 1.2. El sistema guarda la información. 2. Modificar contenido: <ol style="list-style-type: none"> 2.1. El administrador selecciona los datos que desea cambiar. 2.2. El administrador teclea los nuevos datos del contenido. 2.3. El sistema guarda los cambios 3. Eliminar contenido: <ol style="list-style-type: none"> 3.1. El administrador selecciona los datos del contenido que desea eliminar. 3.2. El administrador elimina los datos del contenido. 3.3. El sistema elimina los datos del contenido. 4. Consultar usuarios <ol style="list-style-type: none"> 4.1. El administrador entra en la pantalla principal de contenidos 4.2. El administrador puede navegar en la tabla que muestran a los contenidos registrados

Tabla 5. Caso de uso Contenidos.

3.2 Análisis y Diseño

En la tabla 6 se muestra la planificación de las actividades a realizar y las fechas estimadas.

Actividades: Desarrollo de portal	Fechas. 1 Marzo a 25 Marzo
-Desarrollo de base de datos.	1 marzo a 3 marzo
Creación de formularios para subir información de, minisitios, contenidos, autores, secciones, menús y logo.	4 marzo a 10 marzo
Rutinas de actualización y borrado de contenidos.	11 marzo a 18 marzo
Formularios y programación de usuarios y sus permisos.	19 marzo a 25 marzo
Actividades:Puesta en producción	26 Marzo a 5 Abril
Testeo de página web.	26 marzo a 29 marzo
Ajustes dependiendo de la retroalimentación del testeo.	29 marzo a 2 abril
Aprobación de página web	2 abril a 4 abril
Puesta en producción	5 abril

Tabla 6. Actividades a realizar

Con los requisitos ya planteados por el cliente se dio inicio al diseño de la base de datos, que será implementada en el sistema como se muestra en la figura 9.

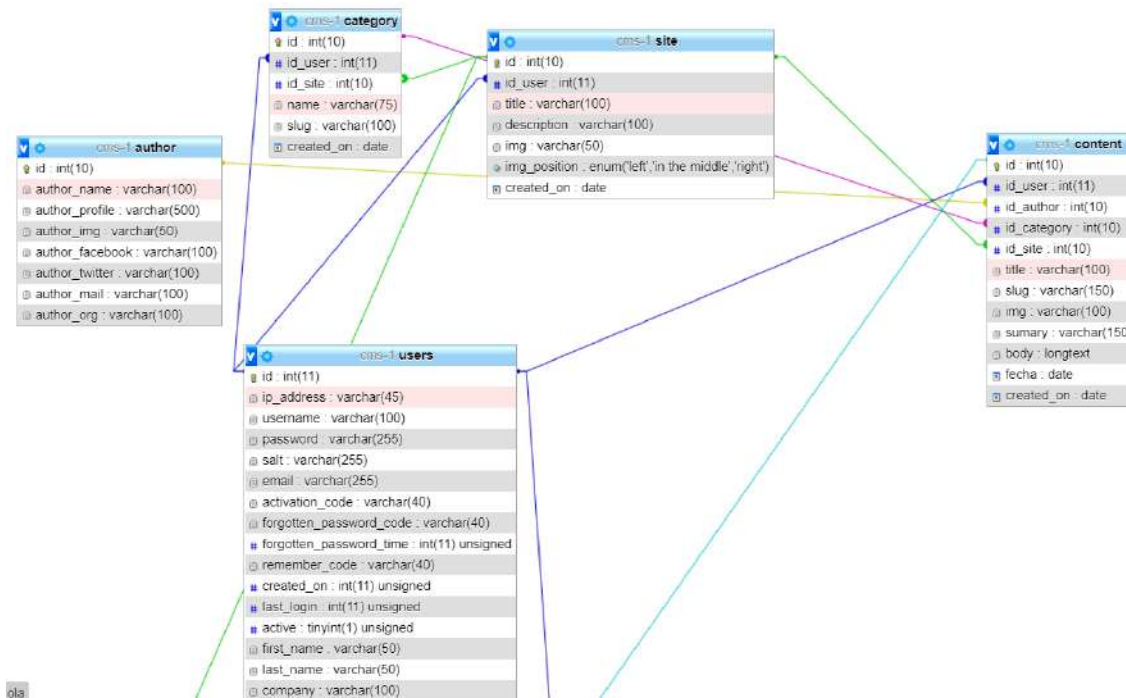


Figura 9. Base de datos.

En la figura 10 se muestra el planteamiento inicial de la interfaz para visualizar los contenidos registrados.

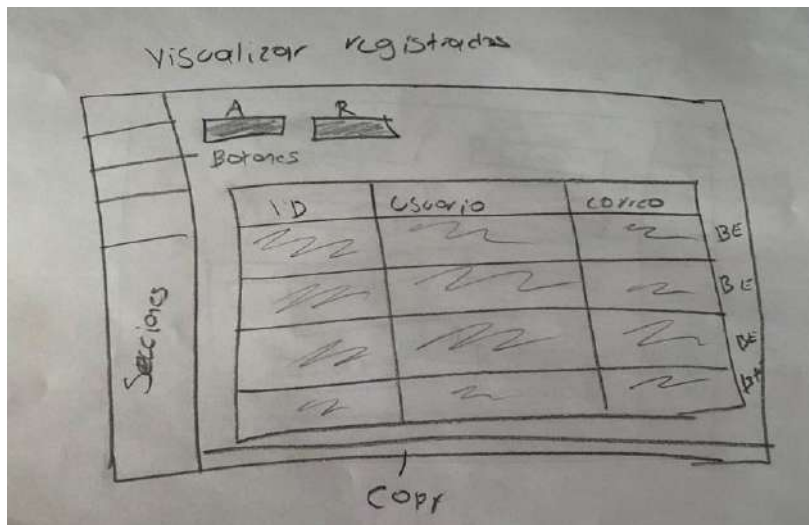


Figura 10. Representación interfaz para visualizar contenidos

En la figura 11 se muestra el planteamiento inicial de la interfaz para ingresar un nuevo contenido y de la misma forma para la edición del contenido.

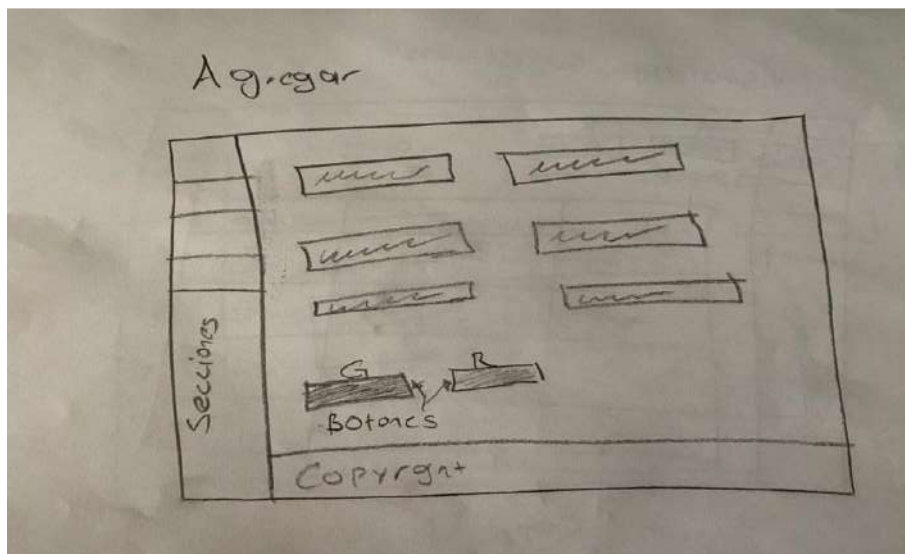


Figura 11. Representación de interfaz para ingresar y editar contenidos.

En la figura 12 se muestra el planteamiento inicial de la interfaz de agregar un contenido nuevo.

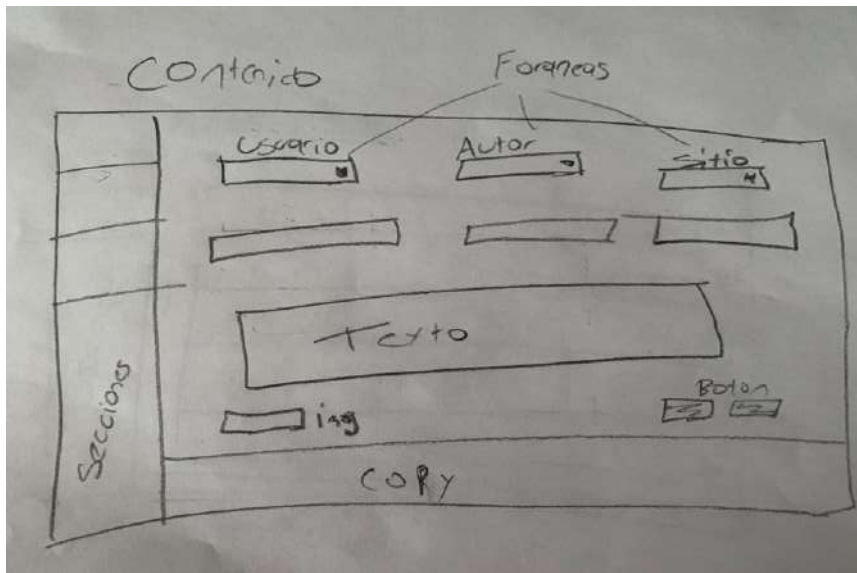


Figura 12. Representación de interfaz para ingresar nuevo contenido.

3.3 Codificación.

En la Figura 13 se muestra la primera versión del inicio de sesión la cual consiste en ingresar un correo electrónico junto con la contraseña.

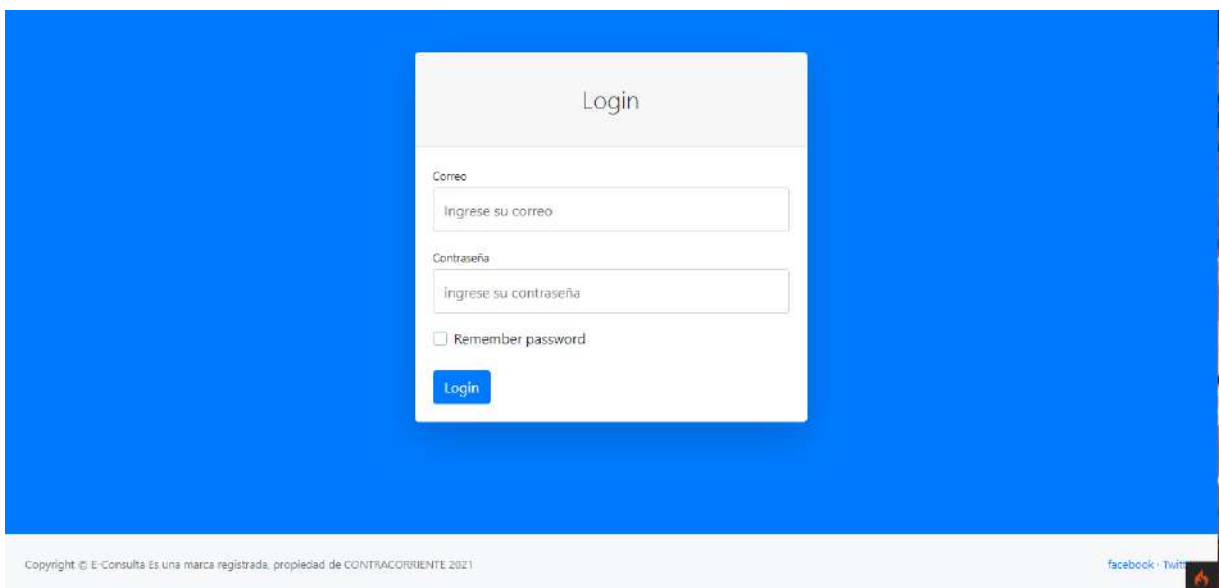


Figura 13. Inicio de sesión versión 1.

En la figura 14 se muestra la codificación del inicio de sesión.

```

<div class="col-lg-5">
  <div class="card shadow-lg border-0 rounded-lg mt-5">
    <div class="card-header"><h3 class="text-center font-weight-light my-4">Login</h3></div>
    <div class="card-body">
      <form method="POST" action="{?php echo base_url();}/users/validacion">
        <div class="form-group">
          <label class="small mb-1" for="username">Usuario</label>
          <input class="form-control py-4" id="username" name="username" type="text" placeholder="Ingrese su usuario" />
        </div>
        <div class="form-group">
          <label class="small mb-1" for="Password">Contraseña</label>
          <input class="form-control py-4" id="Password" name="password" type="password" placeholder="ingrese su contraseña" />
        </div>
        <div class="form-group">
          <div class="custom-control custom-checkbox">
            <input class="custom-control-input" id="rememberPasswordCheck" type="checkbox" />
            <label class="custom-control-label" for="rememberPasswordCheck">Recordar la contraseña </label>
          </div>
        </div>
        <div class="form-group d-flex align-items-center justify-content-between mt-4 mb-0">
          <button class="btn btn-primary" type="submit">Ingresar</button>
        </div>

        <?php if (isset($validation)) { ?>
          <div class="alert alert-danger">
            <?php echo $validation->listErrors();>
          </div>
        <?php } ?>

        <?php if (isset($error)) { ?>
          <div class="alert alert-danger">

```

Figura 14. Codificación de login.

En la figura 15 se muestra la interfaz principal donde se muestran todos los usuarios registrados por el administrador.

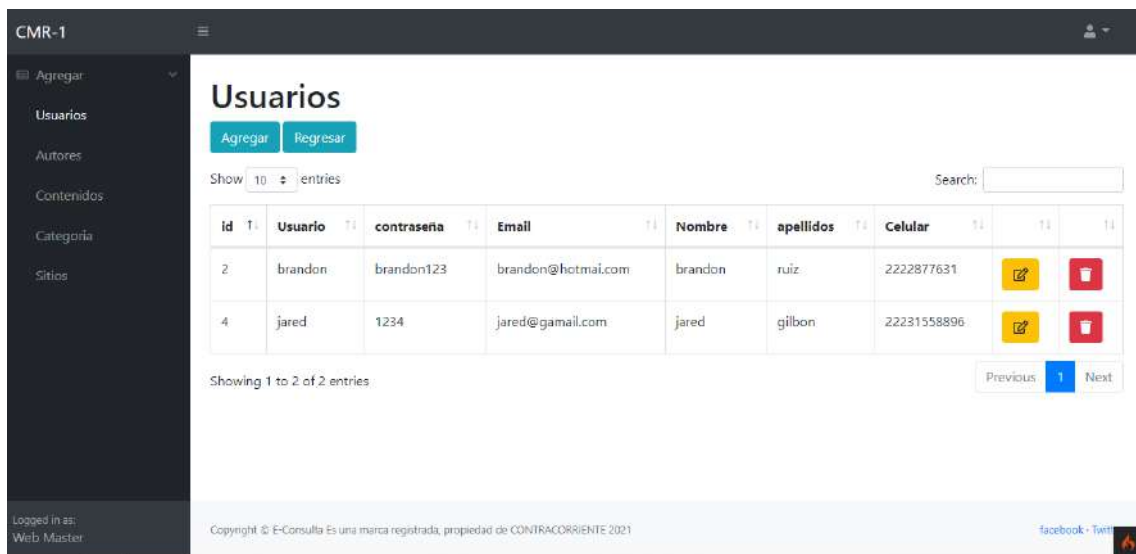


Figura 15. Vista de todos los usuarios registrados.

En la figura 16 se muestra la codificación de la vista donde se muestran los usuarios registrados.


```

<thead>
  <tr>
    <th>id</th>
    <th>Usuario</th>
    <th>contraseña</th>
    <th>Email</th>
    <th>Nombre</th>
    <th>apellidos</th>
    <th>Celular</th>
    <th></th>
    <th></th>
  </tr>
</thead>
<tbody>
  <?php foreach ($datos as $users) { ?>
    <tr>
      <td><?php echo $users ['id']></td>
      <td><?php echo $users ['username']></td>
      <td><?php echo $users ['password']></td>
      <td><?php echo $users ['email']></td>
      <td><?php echo $users ['first_name']></td>
      <td><?php echo $users ['last_name']></td>
      <td><?php echo $users ['phone']></td>
      <td><a href="<?php echo base_url(). '/users/editar/'. $users ['id']; ?>"
        class="btn btn-warning"><i class="far fa-edit"></i></a></td>
      <td><a href="#" data-href="<?php echo base_url() . '/users/eliminar/'. $users ['id']; ?>"
        data-toggle="modal" data-target="#modal_confirmar" data-placement="top" title="eliminar registro"
        class="btn btn-danger"><i class="fas fa-trash"></i></a></td>
    </tr>
  </tbody>

```

Figura 16. Codificación vista de todos los usuarios registrados.

En la figura 17 se muestra la interfaz donde se agrega un nuevo usuario.

Figura 17. Vista nuevo usuario.

En la figura 18 se muestra la codificación de la vista donde se muestra el formulario para ingresar un usuario nuevo.

```
<div class="col-12 col-sm-6">
  <label>Nombre De Usuario</label>
  <input class ="form-control" id="username" name="username" type="text" autofocus required />
</div>
<div class="col-12 col-sm-6">
  <label>Contraseña</label>
  <input class ="form-control" id="password" name="password" type="password" required />
</div>
<div class="col-12 col-sm-6">
  <label>Email</label>
  <input class ="form-control" id="email" name="email" type="text" />
</div>
<div class="col-12 col-sm-6">
  <label>Nombre</label>
  <input class ="form-control" id="first_name" name="first_name" type="text" required/>
</div>
<div class="col-12 col-sm-6">
  <label>Apellidos</label>
  <input class ="form-control" id="last_name" name="last_name" type="text" required />
</div>
<div class="col-12 col-sm-6">
  <label>Celular</label>
  <input class ="form-control" id="phone" name="phone" type="text" required />
</div>
</div>
</div>
<p>
  <button type="submit" class="btn btn-success">Guardar</button>
  <a href="{php echo base_url (); }/users" class="btn btn-primary">regresar</a>
</p>
</div>
```

Figura 18. Codificación vista nuevo usuario.

En la figura 19 se muestra la interfaz para editar un usuario registrado anteriormente.

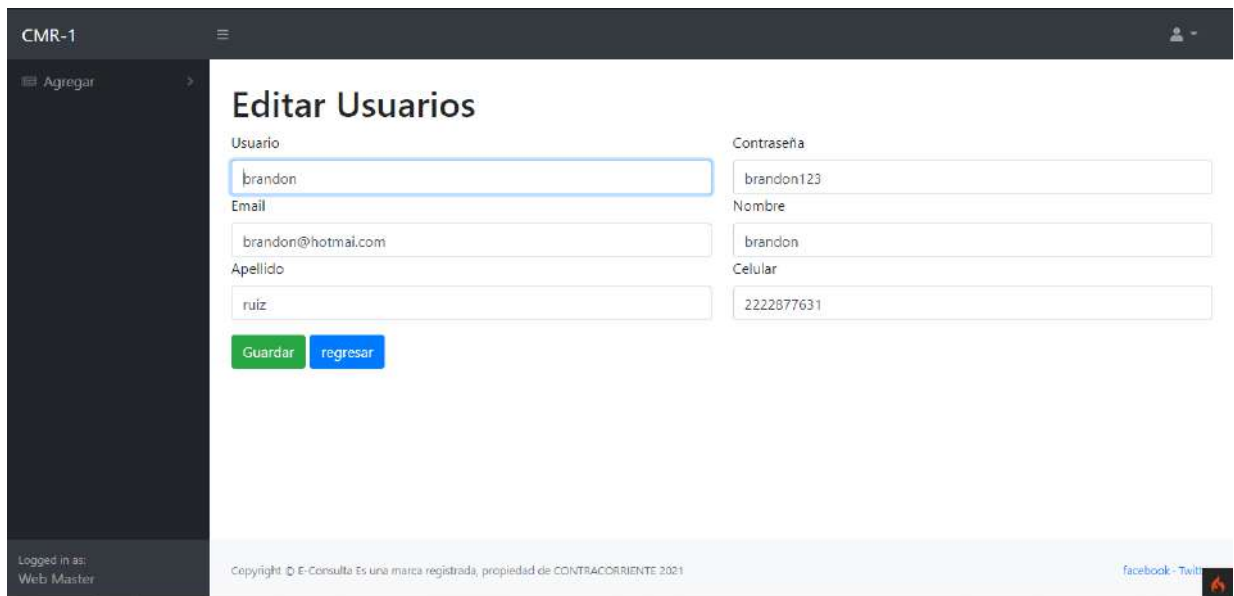


Figura 19. Vista edición de usuarios.

En la figura 20 se muestra la codificación de la vista donde se modifica un usuario ya registrado.

```
<div class="col-12 col-sm-6">
  <label>Usuario</label>
  <input class ="form-control" id="username" name="username" type="text"
  value="<?php echo $datos ['username']; ?>" autofocus require />
</div>
<div class="col-12 col-sm-6">
  <label>Contraseña</label>
  <input class ="form-control" id="password" name="password" type="text"
  value="<?php echo $datos ['password']; ?>" require />
</div>
<div class="col-12 col-sm-6">
  <label>Email</label>
  <input class ="form-control" id="email" name="email" type="text"
  value="<?php echo $datos ['email']; ?>" />
</div>
<div class="col-12 col-sm-6">
  <label>Nombre</label>
  <input class ="form-control" id="first_name" name="first_name" type="text"
  value="<?php echo $datos ['first_name']; ?>" require />
</div>
<div class="col-12 col-sm-6">
  <label>Apellido</label>
  <input class ="form-control" id="last_name" name="last_name" type="text"
  value="<?php echo $datos ['last_name']; ?>" require />
</div>
<div class="col-12 col-sm-6">
  <label>Celular</label>
  <input class ="form-control" id="phone" name="phone" type="text"
  value="<?php echo $datos ['phone']; ?>" require />
</div>
```

Figura 20. Codificación vista edición de usuario.

En la figura 21 se muestra una alerta al querer eliminar un usuario.

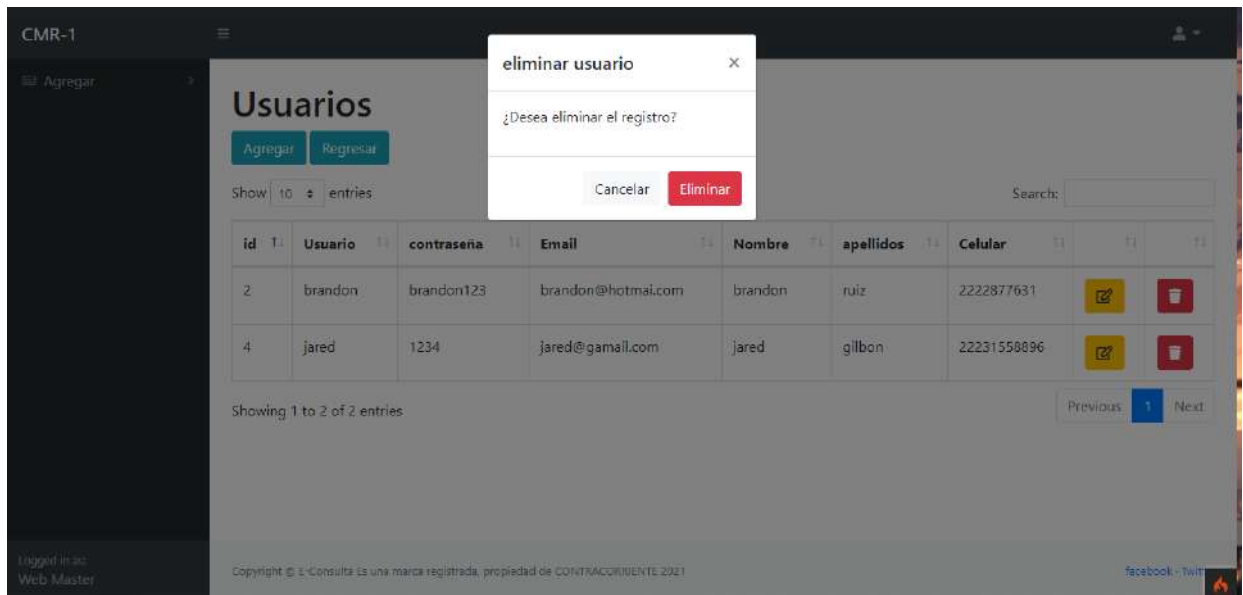


Figura 21. Vista al intentar eliminar un registro.

En la figura 22 se muestra la codificación de la vista donde se elimina un usuario.

```

<!-- Modal -->
<div class="modal fade" id="modal_confirmar" tabindex="-1" role="dialog" aria-labelledby="exampleModallabel" aria-hidden="true">
  <div class="modal-dialog modal-sm" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModallabel">eliminar usuario</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <p>¿Desea eliminar el registro?</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-light" data-dismiss="modal">Cancelar</button>
        <a class="btn btn-danger btn-ok">Eliminar</a>
      </div>
    </div>
  </div>
</div>

```

Figura 22. Codificación vista eliminación de usuario.

En la figura 23 se muestra la vista de los autores registrados.

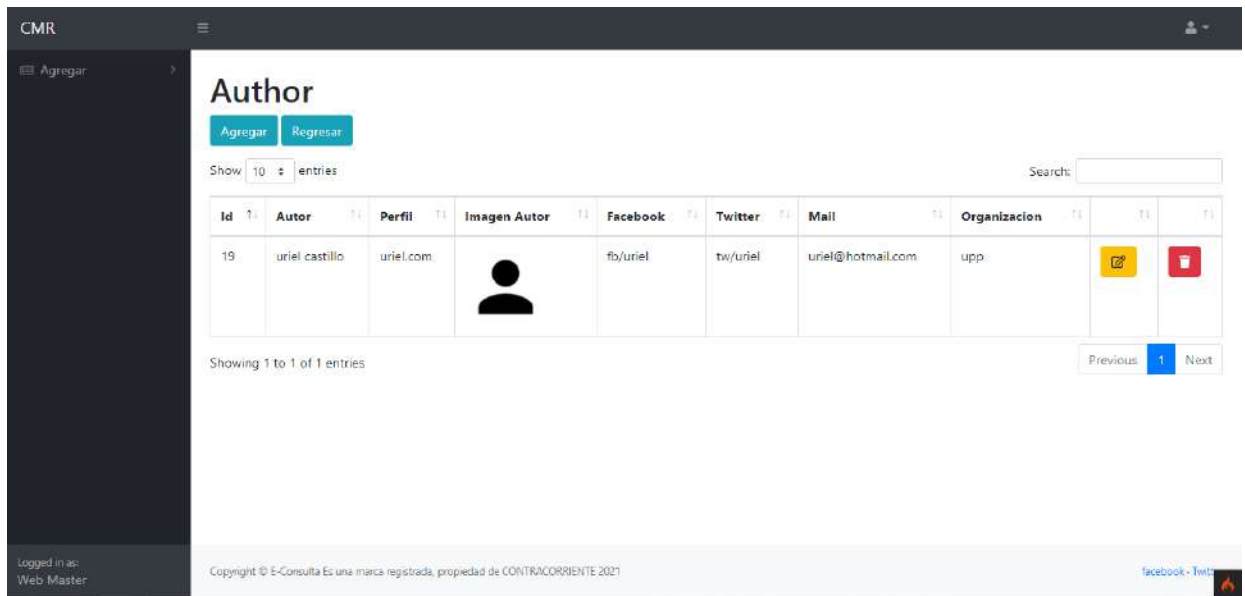


Figura 23. Vista autores registrados.

En la figura 24 se muestra la vista para agregar un nuevo autor.

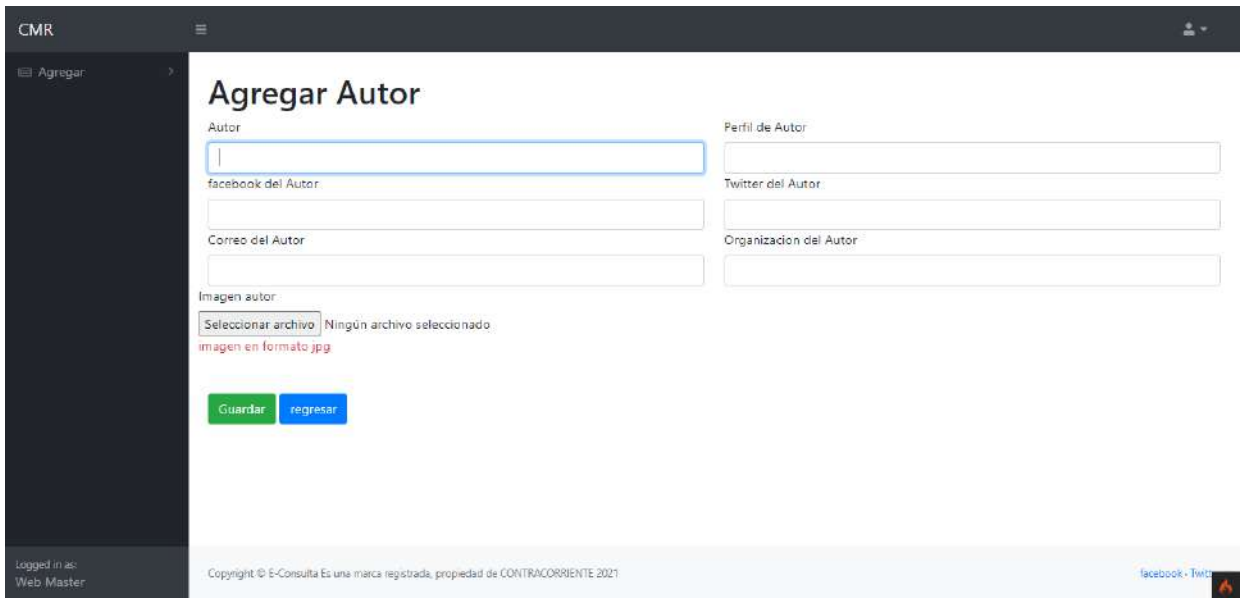


Figura 24. vista nuevo autor.

En la figura 25 se muestra la vista al editar un autor registrado.

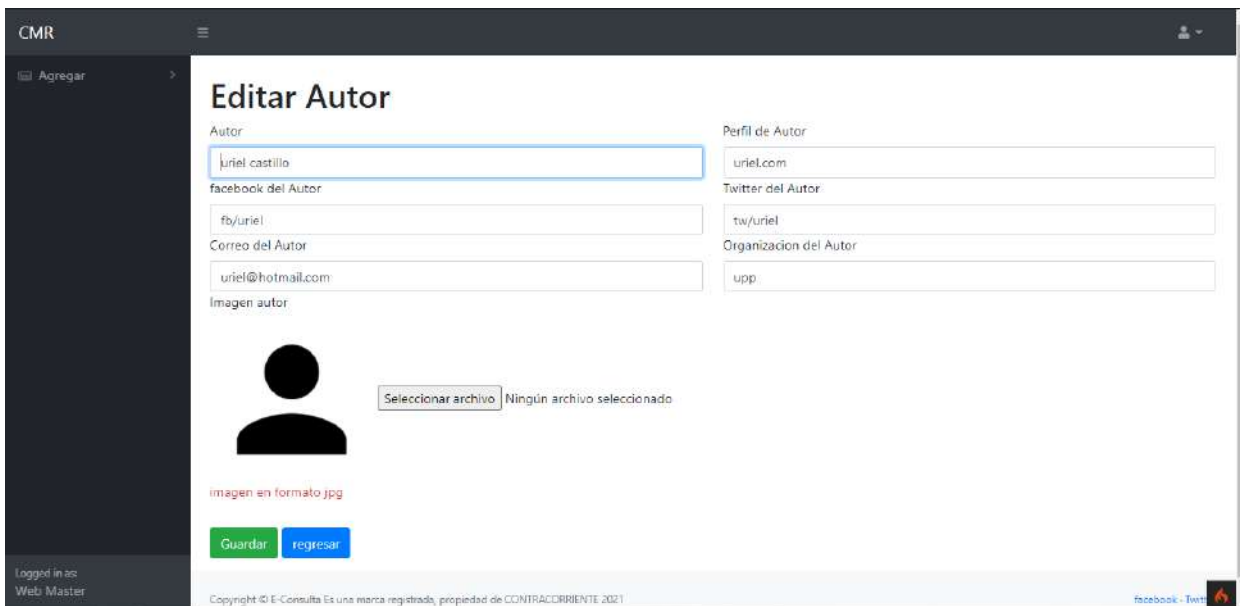


Figura 25. vista editar autor

En la figura 26 se muestra la vista al querer eliminar un autor registrado.

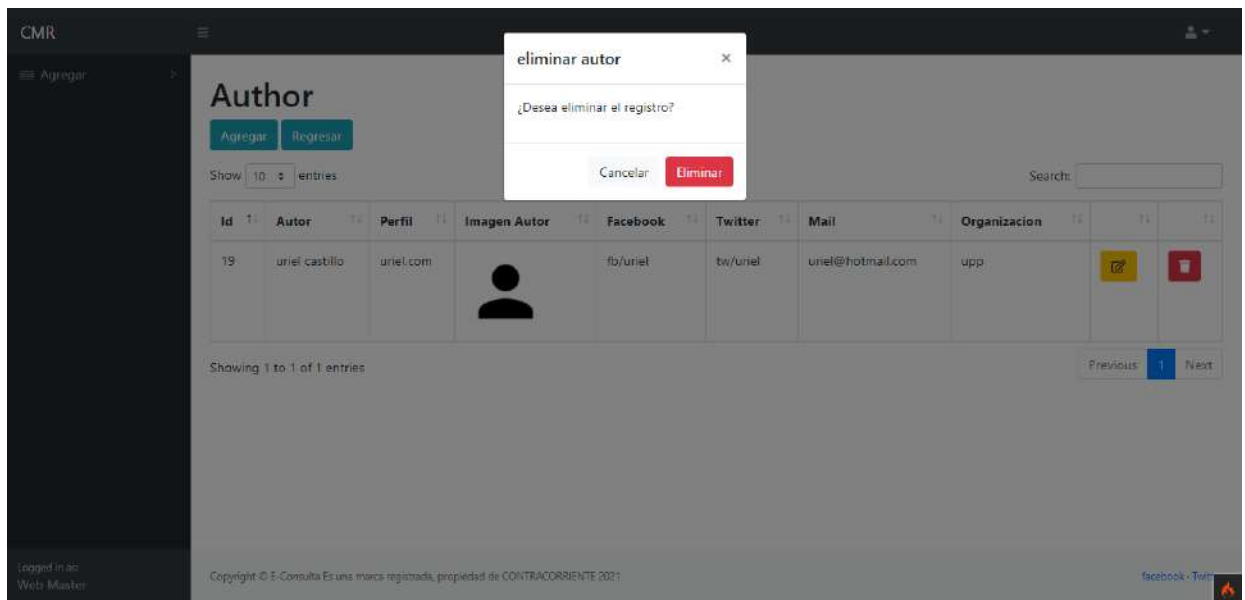


Figura 26. vista eliminar autor

En la figura 27 se muestra la vista de las categorías registradas

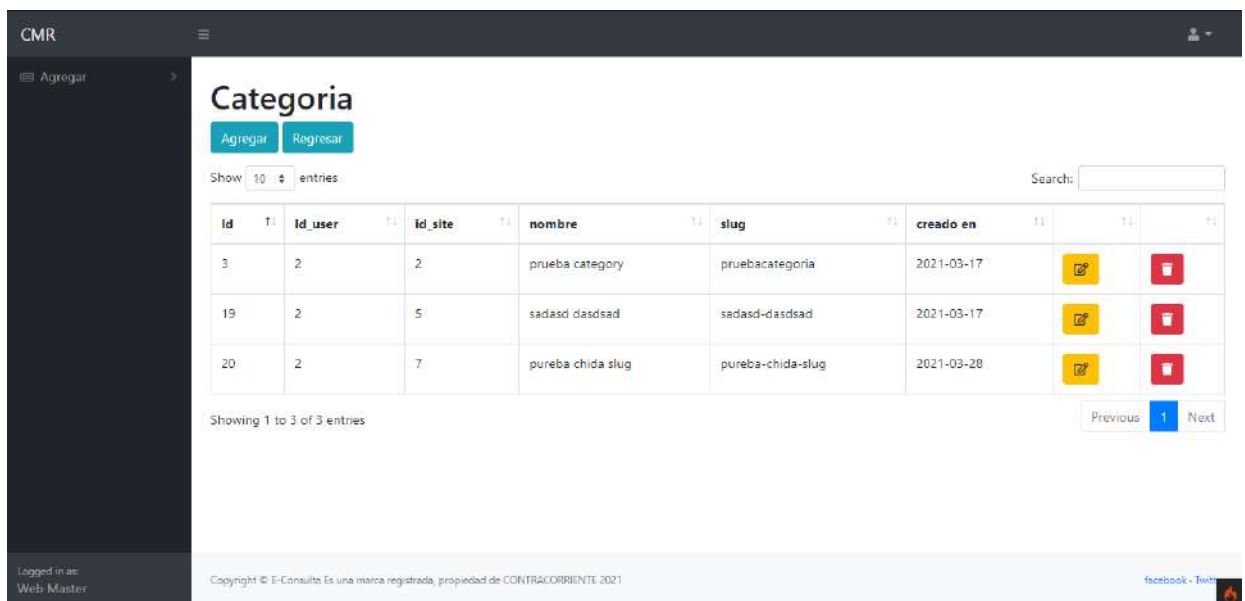


Figura 27. vista de categorías registradas

En la figura 28 se muestra la vista para agregar una nueva categoría donde se puede seleccionar el id de un usuario y un sitio ya registrado.

The screenshot shows a web interface for adding a new category. The page title is 'Categoría'. It features two dropdown menus for selecting a user ID ('id de usuario') and a site ID ('id del sitio'). Below these are input fields for 'Nombre' (Name) and 'Slug'. A note indicates that the slug is generated automatically upon saving. A date picker is labeled 'Creado el' (Created on). At the bottom, there are 'Guardar' (Save) and 'regresar' (Back) buttons. The footer includes the user 'Web Master', copyright information for 'E-Consulta' (2021), and social media links for Facebook and Twitter.

Figura 28. vista agregar nueva categoría

En la figura 29 se muestra la vista para editar una categoría registrada.

The screenshot shows a web interface for editing an existing category. The page title is 'Editar categoría'. It features two dropdown menus for selecting a user ID ('id del usuario') and a site ID ('id del sitio'). Below these are input fields for 'Nombre' (Name) and 'Slug'. A date picker is labeled 'Creado el' (Created on). At the bottom, there are 'Guardar' (Save) and 'regresar' (Back) buttons. The footer includes the user 'Web Master', copyright information for 'E-Consulta' (2021), and social media links for Facebook and Twitter.

Figura 29. editar categoría

En la figura 30 se muestra la vista al querer eliminar una categoría.

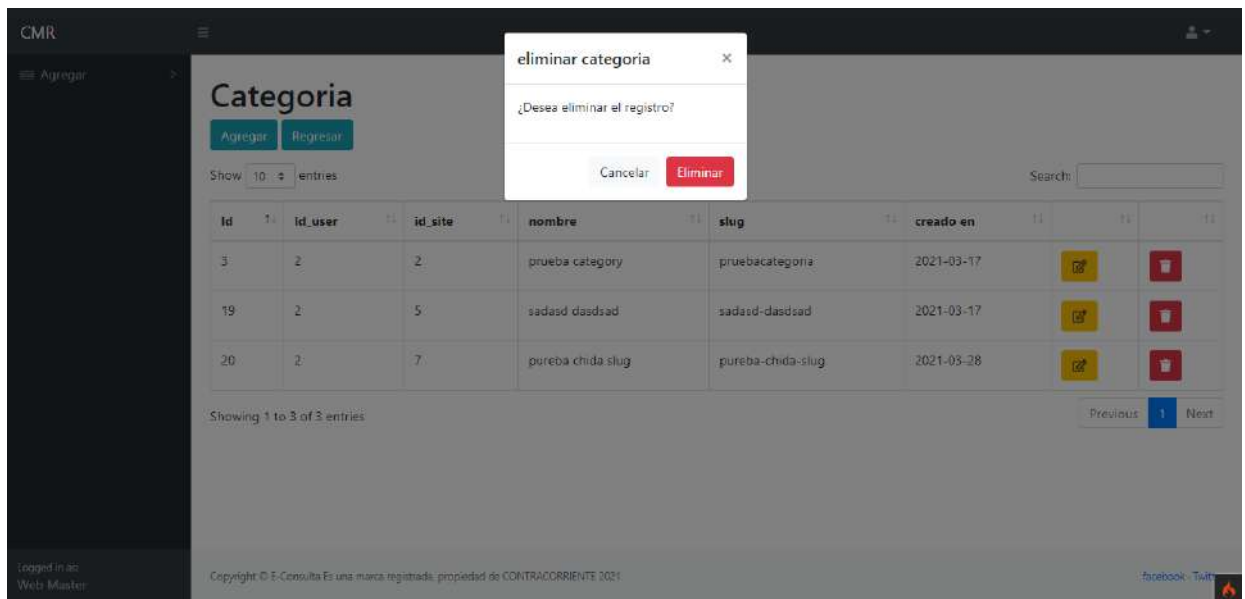


Figura 30. vista eliminar categoría.

En la figura 31 se muestra la vista de los sitios registrados.

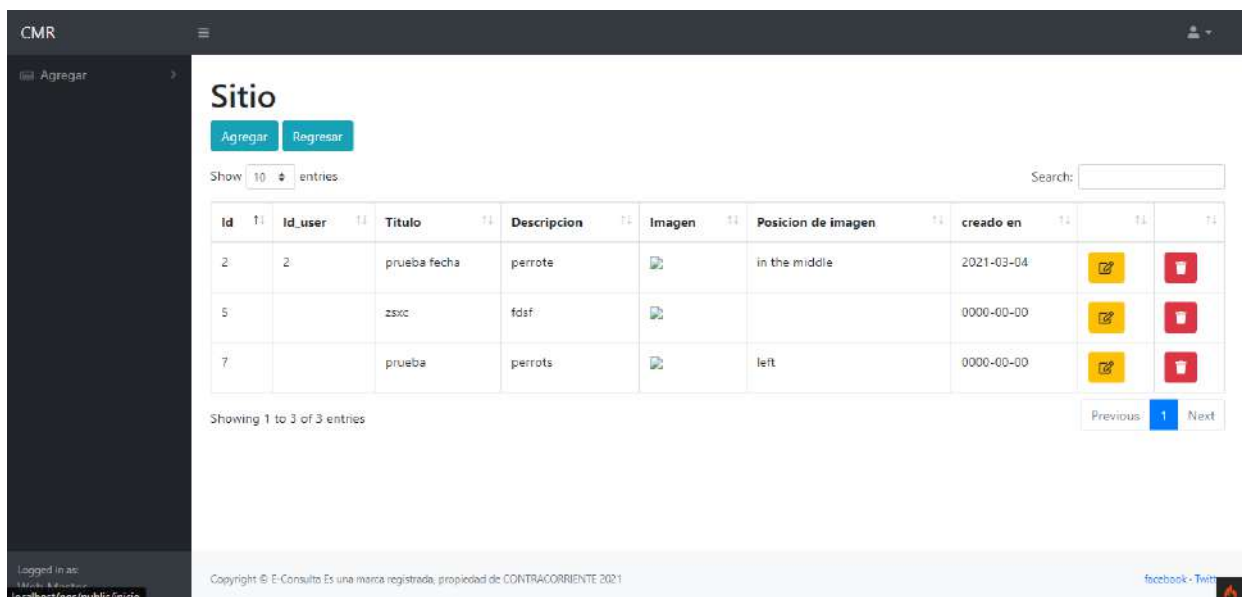


Figura 31. vista sitios registrados.

En la figura 32 se muestra la vista para agregar un nuevo sitio, donde se podrá seleccionar el id del usuario para realizar el registro.

The screenshot shows a web application interface for adding a new site. The header includes 'CMR' and a user profile icon. A dark sidebar on the left contains a menu with 'Agregar' and a right-pointing arrow. The main content area is titled 'Sitios'. It features several form fields: a dropdown for 'Seleccione el id del usuario' with 'id de usuario' selected; a dropdown for 'Seleccione la posición de la imagen' with 'Posición de imagen' selected; a text input for 'Titulo'; a date input for 'Creado el' with 'dd/mm/aaaa' and a calendar icon; a text input for 'Descripcion'; and an image upload section with a 'Seleccionar archivo' button, the text 'Ningún archivo seleccionado', and a red error message 'imagen en formato jpg'. At the bottom of the form are two buttons: 'Guardar' (green) and 'regresar' (blue). The footer contains 'Logged in as: Web Master', copyright information 'Copyright © E-Consulta Es una marca registrada, propiedad de CONTRACORRIENTE 2021', and social media links for 'facebook' and 'Twitter'.

Figura 32. vista nuevo sitio

En la figura 33 se muestra la vista para editar un sitio registrado.

The screenshot shows the 'Editar Sitios' form. The header and sidebar are identical to the previous figure. The main content area is titled 'Editar Sitios'. The form fields are pre-filled: 'Selecciones el id del usuario' is set to 'brandón'; 'Seleccione la posición de la imagen' is set to 'centrada'; 'Titulo' contains 'prueba fecha'; 'Creado el' is set to '04/03/2021'; and 'Descripcion' contains 'perrote'. The image upload section shows a small image icon, the 'Seleccionar archivo' button, and the text 'Ningún archivo seleccionado' with the red error message 'imagen en formato jpg'. The 'Guardar' (green) and 'regresar' (blue) buttons are present at the bottom. The footer information is also consistent with the previous figure.

Figura 33. vista editar sitio.

En la figura 34 se muestra la vista al querer eliminar un sitio.

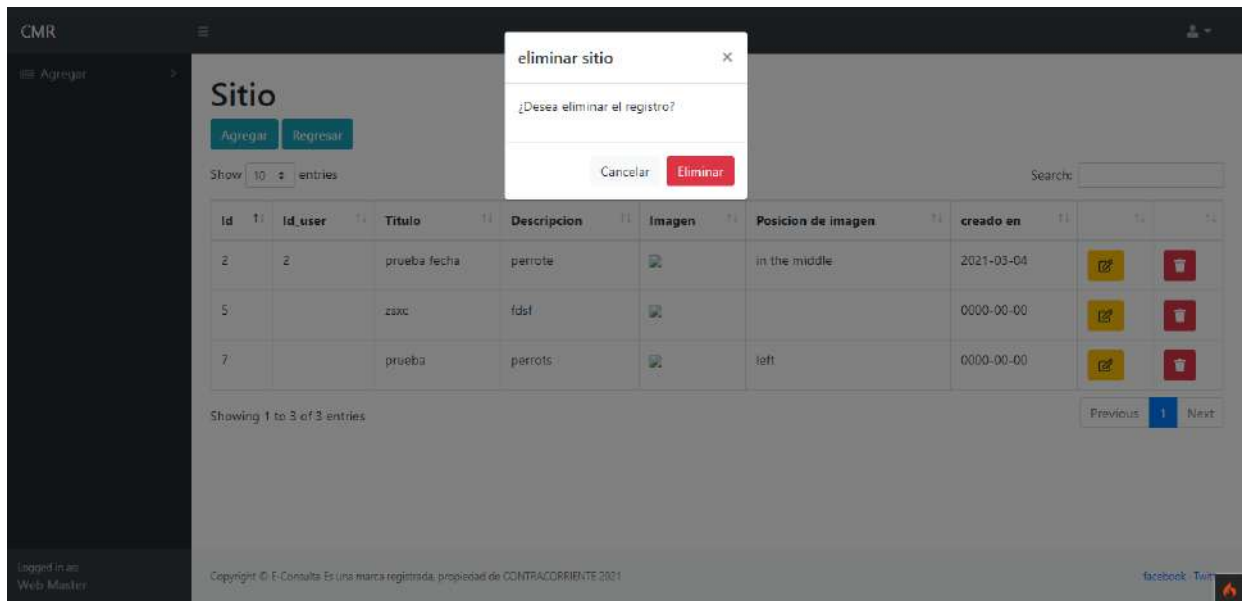


Figura 34. vista eliminar sitio.

3.4. Pruebas.

Se realizaron pruebas de usabilidad a tres sujetos donde se pidió que realizaran tres acciones dentro del sistema y posteriormente evaluaran la experiencia que tuvieron a realizar las tareas las cuales fueron Agregar un autor, editarlo y eliminarlo.

Para poder evaluar a los sujetos se realizaron las siguientes preguntas:

- ¿Considera que el sistema es facil de utilizar?
- ¿Le gustaría utilizar este sistema frecuentemente?
- ¿La página de inicio se distingue de las demás páginas?
- ¿podria hacer uso del sistema sin tener que aprender cosas nuevas?
- ¿Podria hacer uso del sistema sin la necesidad de alguna una ayuda?
- ¿las diferentes funciones del sistema se encuentran muy bien integradas?
- ¿las personas podrian aprendender a utilizar el sistema rápidamente?
- ¿Considera que el sistema es seguro?
- ¿Considera que el sistema es consistente?
- ¿Se sientio comodo durante el uso del sistema?

En la encuesta existen 5 respuestas disponibles donde el encuestado tiene que elegir un valor entre 1 a 5. donde los valores se consideran como:

1: Totalmente en desacuerdo, 2: En desacuerdo, 3: Neutral, 4: De acuerdo, 5: Totalmente de acuerdo

Para evaluar el sistema se hizo uso de esta puntuación donde si se obtiene el 100% significa que el sistema es perfecto, si se obtiene hasta el 70% son considerados buenos y se considera que cualquier valor inferior al 50% es muy ineficiente en términos de usabilidad.

En la figura 35 se muestra la primera prueba de usabilidad que se realizó, el porcentaje que se obtuvo fue de 82%.

Id: Cp-01		Pruebas de usabilidad "generador de minisitios"				
Propósito: verificar registro, modificar y eliminar registro.						
Autor de caso: Empleado 1		escala				
no.pregunta	Pregunta	1. Totalmente en desacuerdo	2. En Desacuerdo	3. Ni de acuerdo ni en desacuerdo	4. De acuerdo	5. Totalmente de acuerdo
1	¿Considera que el sistema es facil de utilizar?					x
2	¿Le gustaria utilizar este sistema frecuentemente?			x		
3	¿La página de inicio se distingue de las demás páginas?				x	
4	¿podria hacer uso del sistema sin tener que aprender cosas nuevas?				x	
5	¿Podria hacer uso del sistema sin la necesidad de alguna una ayuda?				x	
6	¿las diferentes funciones del sistema se encuentran muy bien integradas?			x		
7	¿las personas podrian aprender a utilizar el sistema rápidamente?					x
8	¿Considera que el sistema es seguro?			x		
9	¿Considera que el sistema es consistente?					x
10	¿Se siento comodo durante el uso del sistema?					x
				total = 41/50	82%	

Figura 35. Prueba usabilidad 1.

En la figura 36 se muestra la segunda prueba de usabilidad que se realizó, el porcentaje que se obtuvo fue de 80%.

Id: Cp-02		Pruebas de usabilidad "generador de minisitios"				
Propósito: verificar registro, modificar y eliminar registro.						
Autor de caso: Empleado 2		escala				
no.pregunta	Pregunta	1. Totalmente en desacuerdo	2. En Desacuerdo	3. Ni de acuerdo ni en desacuerdo	4. De acuerdo	5. Totalmente de acuerdo
1	¿Considera que el sistema es facil de utilizar?				x	
2	¿Le gustaria utilizar este sistema frecuentemente?			x		
3	¿La página de inicio se distingue de las demás páginas?					x
4	¿podria hacer uso del sistema sin tener que aprender cosas nuevas?			x		
5	¿Podria hacer uso del sistema sin la necesidad de alguna una ayuda?					x
6	¿las diferentes funciones del sistema se encuentran muy bien integradas?				x	
7	¿las personas podrian aprender a utilizar el sistema rápidamente?				x	
8	¿Considera que el sistema es seguro?			x		
9	¿Considera que el sistema es consistente?					x
10	¿Se siento comodo durante el uso del sistema?				x	
				total 40/50=	80%	

Figura 36. Prueba usabilidad 2

En la figura 37 se muestra la tercera prueba de usabilidad que se realizó, el porcentaje que se obtuvo fue de 92%.

Id: Cp-03		Pruebas de usabilidad "generador de minisitios"				
Propósito: verificar registro, modificar y eliminar registro.						
Autor de caso: Empleado 1.		escala				
no. pregunta	Pregunta	1. Totalmente en desacuerdo	2. En Desacuerdo	3. Ni de acuerdo ni en desacuerdo	4. De acuerdo	5. Totalmente de acuerdo
1	¿Considera que el sistema es facil de utilizar?					x
2	¿Le gustaría utilizar este sistema frecuentemente?					x
3	¿La página de inicio se distingue de las demás páginas?					x
4	¿podría hacer uso del sistema sin tener que aprender cosas nuevas?					x
5	¿Podría hacer uso del sistema sin la necesidad de alguna una ayuda?			x		
6	¿las diferentes funciones del sistema se encuentran muy bien integradas?					x
7	¿las personas podrían aprender a utilizar el sistema rápidamente?				x	
8	¿Considera que el sistema es seguro?				x	
9	¿Considera que el sistema es consistente?					x
10	¿Se siento comodo durante el uso del sistema?					x
		total 46/50 =			92%	

Figura 37. Prueba usabilidad 3

En la tabla 7 se muestra el resultado que obtuvo el encuestado de la prueba 1

Nombre de proyecto: generador de minisitios		Id: Caso de prueba : CP-01	
Autor de caso : Empleado de la empresa.		Propósito: verificar el registro, Edición y eliminación de autor	
#	acciones	Salida esperada	Salida obtenida
01	Registrar datos de autor.	Mostrar el autor en la tabla de autores.	Muestra datos de los autores registrados.
02	Edición a los datos de autor.	Mostrar el autor editado en la tabla de autores.	Muestra el autor editado en la tabla de autores.
03	Eliminar autor	Eliminar autor de la tabla de autores	Eliminación de autor en la tabla de autores
Resultado: Aprobado			

Tabla 7. Evaluación de usabilidad CP-01

En la tabla 8 se muestra el resultado que obtuvo el encuestado de la prueba 2

Nombre de proyecto: generador de minisitios		Id: Caso de prueba : CP-02	
Autor de caso : Empleado de la empresa.		Propósito: verificar el registro, Edición y eliminación de autor	
#	acciones	Salida esperada	Salida obtenida
01	Registrar datos de autor.	Mostrar el autor en la tabla de autores.	Muestra datos de los autores registrados.
02	Edición a los datos de autor.	Mostrar el autor editado en la tabla de autores.	Muestra el autor editado en la tabla de autores.
03	Eliminar autor	Eliminar autor de la tabla de autores	Eliminación de autor en la tabla de autores
Resultado: Aprobado			

Tabla 8. Evaluación de usabilidad CP-02.

En la tabla 9 se muestra el resultado que obtuvo el encuestado de la prueba 3

Nombre de proyecto: generador de minisitios		Id: Caso de prueba : CP-03	
Autor de caso : Empleado de la empresa.		Propósito: verificar el registro, Edición y eliminación de autor	
#	acciones	Salida esperada	Salida obtenida
01	Registrar datos de autor.	Mostrar el autor en la tabla de autores.	Muestra datos de los autores registrados.
02	Edición a los datos de autor.	Mostrar el autor editado en la tabla de autores.	Muestra el autor editado en la tabla de autores.
03	Eliminar autor	Eliminar autor de la tabla de autores	Eliminación de autor en la tabla de autores
Resultado: Aprobado			

Tabla 9. Evaluación de usabilidad CP-03.

En la figura 38 se muestra el porcentaje total de cada una de las preguntas contestadas en las pruebas.



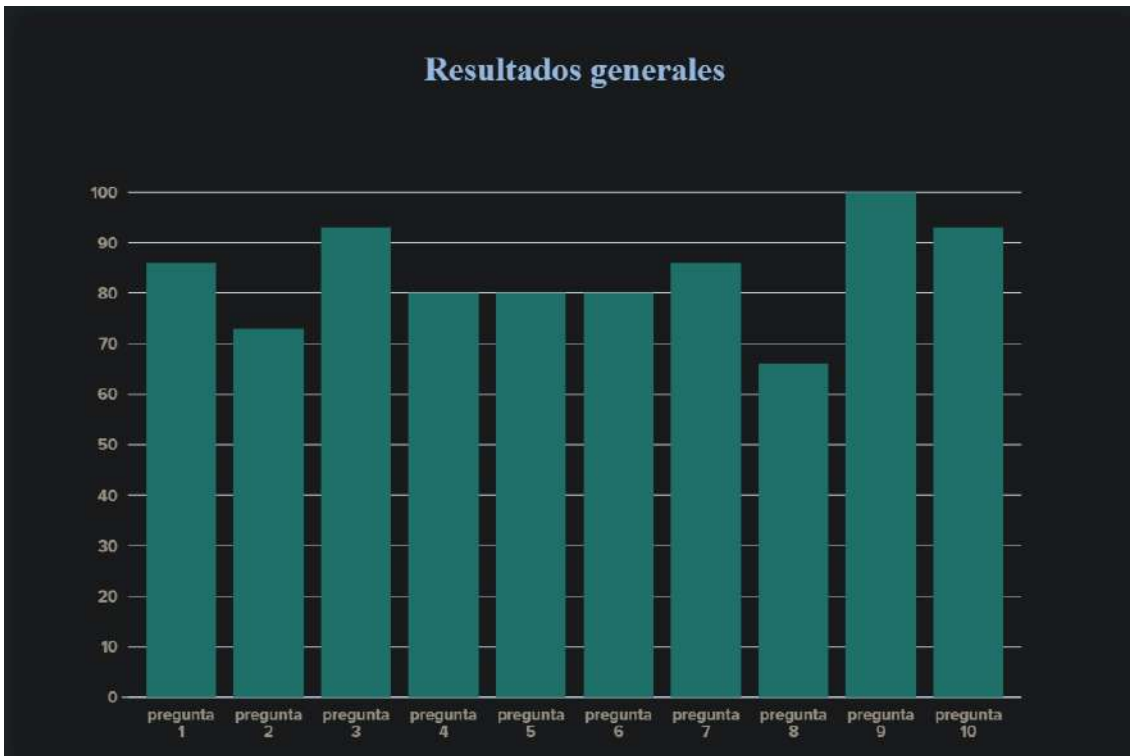


Figura 38. Resultados generales de las pruebas.

Además, para el apartado de "Contenidos" se muestra de una forma diferente ya que es la unión de todos los apartados anteriores, en este apartado se realizaron las pruebas del sistema, ya que depende de los apartados anteriores y requiere estén completos y sin errores para poder realizar la tarea de crear un nuevo contenido correctamente. En la primera versión del sistema se ingresaban los datos como se muestra en la figura 39.

POS Code p

Agregar

Contenido

Seleccione el nombre usuario
Nombre usuario

Seleccione el nombre del autor
Nombre autor

Seleccione el nombre de la categoría
Nombre categoría

Seleccione el nombre del sitio
Nombre sitio

Título

Cuerpo

Slug

Resumen

Fecha de noticia
dd/mm/aaaa

Creado el
dd/mm/aaaa

Imagen
Seleccionar archivo | Ningún archivo seleccionado

Guardar regresar

Logged in as: Web Master

Copyright © E-Contalta Es una marca registrada, propiedad de CONTRACORRIENTE 2021

facebook - Twitter

Figura 39. Primera versión para ingresar contenido nuevo

En la figura 40 se muestra como se ve un contenido insertado correctamente lo cual significa que los demás apartados funcionan correctamente y permite crear correctamente un contenido

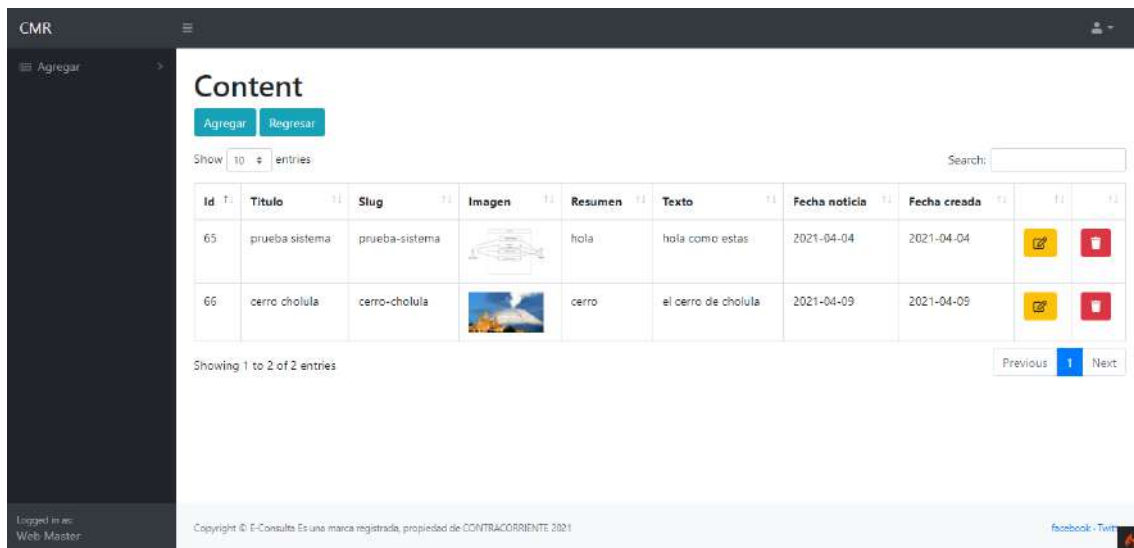


Figura 40. vista de contenidos ingresados.

3.5 Mantenimiento.

Tomando en cuenta la retroalimentación del usuario se decido cambiar la forma de ingresar el apartado de “cuerpo” ya que los periodistas al escribir el cuerpo de una noticia usan diferentes tipos de letras, párrafos, negritas y demás acciones por lo cual se optó por el uso de una herramienta la cual nos brinda las cualidades anteriormente comentadas y muchas más. La aplicación es TinyMCE 5 la cual es un editor de texto enriquecido, potente y flexible que puede integrarse en aplicaciones web asi como se muestra en la figura 41.

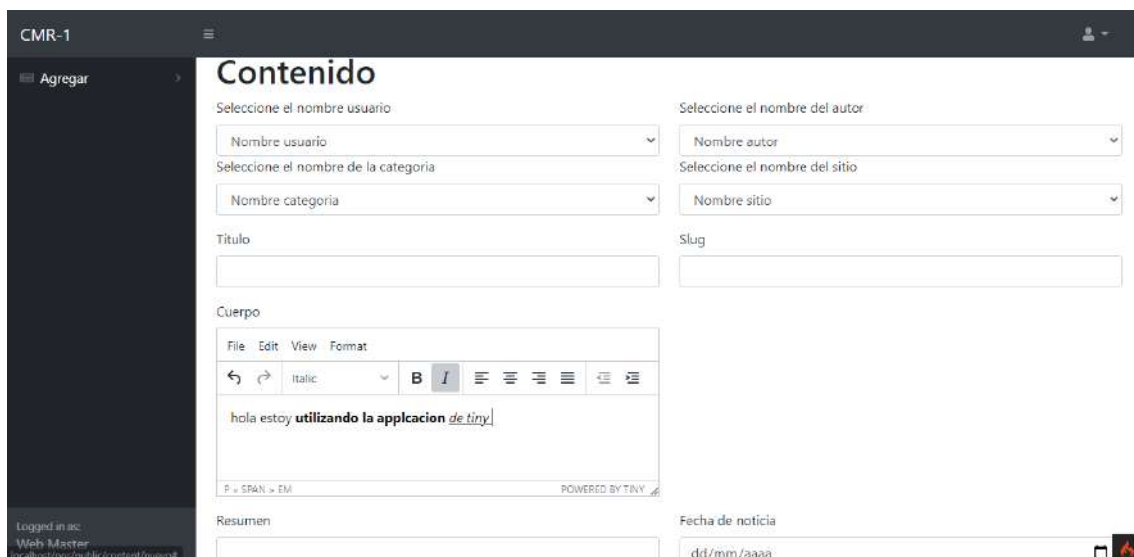


Figura 41. Segunda versión en el apartado de contenido.

Otra de las cosas que se obtuvieron en la retroalimentación con los clientes fue que el inicio de sesión era con correo y ellos necesitaban que se iniciara sesión con nombre de usuario y su contraseña por lo cual se realizó el cambio que se muestra en la figura 42.

Copyright ©. E-Consulta Es una marca registrada, propiedad de CONTRACORRIENTE 2021

facebook · Twitter

Figura 42. Segunda versión en el inicio de sesión.

4. Conclusiones y recomendaciones

Durante el desarrollo del proyecto existieron múltiples conflictos desde el aprendizaje del framework codeigniter y el funcionamiento del modelo vista controlador, hasta el cómo iba a funcionar el sistema.

En conclusión: en el desarrollo del proyecto se obtuvo mucho conocimiento desde el uso de una metodología, el mismo desarrollo del sistema, redacción del documento, uso de nuevas herramientas para facilitar el desarrollo del sistema, en resumen, las cosas aprendidas ayudaron a mejorar la forma en cómo desarrollar y plantear un proyecto de forma correcta y eficaz.

5. Referencias bibliográficas

- [1] ARMENTA-BENITEZ, Brayan, RODRIGUEZ-ESPINOZA, Indelfonso, MEDINA-MUÑOZ, Luis Arturo y GONZALEZ-LÓPEZ, Samuel. Aplicación del modelo de prototipos: Caso de estudio Software RedbotGamesShop. Revista de Simulación Computacional. 2018. Fecha de consulta: 16/abril/2021
- [2] URL: <https://metodologia.es/prototipado/> Blog web dedicado a las metodologías de desarrollo y mantenimiento de software y sistemas de información. Fecha de consulta: 22/febrero/2021
- [3] URL: https://www.ecured.cu/Modelo_de_prototipos enciclopedia donde se pueden encontrar diversos artículos. Fecha de consulta: 22/febrero/2021
- [4] URL: <https://helisulbaransistemas.blogspot.com/2014/09/paradigmas-en-el-desarrollo-de-software.html> Blog web donde tiene una gran variedad de artículos relacionadas a la tecnología y la programación. Fecha de consulta: 22/febrero/2021
- [5] centro de investigación en matemáticas, A.C. (s. f.). Arquitectura para colaboración(1.a.ed.,Vol.1).<https://cimat.repositorioinstitucional.mx/jspui/bitstream/1008/206/2/TE%20326.pdf>
- [6] URL: <https://desarrolloweb.com/articulos/codeigniter.html> Blog enfocado en el aprendizaje de desarrollo web. Fecha de consulta 22/febrero/2021
- [7] Suehring, S., Converse, T., & Park, J. (2009). La biblia de PHP 6 y MySQL / PHP 6 and MySQL 6 Bible (Tra ed.). Anaya Multimedia.
- [8] Gilfillan, I. (2003). La Biblia de MySQL / MySQL 4 (Har/Cdr Tr ed., Vol. 4). Anaya Multimedia.
- [9] Bravo, C. “Sistema de información para Encuestas Simples basado en Computación Móvil Multiplataforma” [En Línea] Proyecto de Tesis para Licenciatura en Computación, Facultad de Ciencias, Universidad Central de Venezuela, 2014
- [10] Herrera, E. “Arrancar con HTML5. Curso de programación” 1ra Edición Alfaomega Grupo Editor, México, (2011)



Universidad Politécnica de Puebla
Ingeniería en Informática

Uriel Castillo kuri
Rigoberto Martínez López
Rebeca Rodríguez Huesca

Este documento se distribuye para los términos de la
Licencia 2.5 Creative Commons (CC-BY-NC-ND 2.5 MX)