



Facultad 1

Centro de Ideoinformática

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

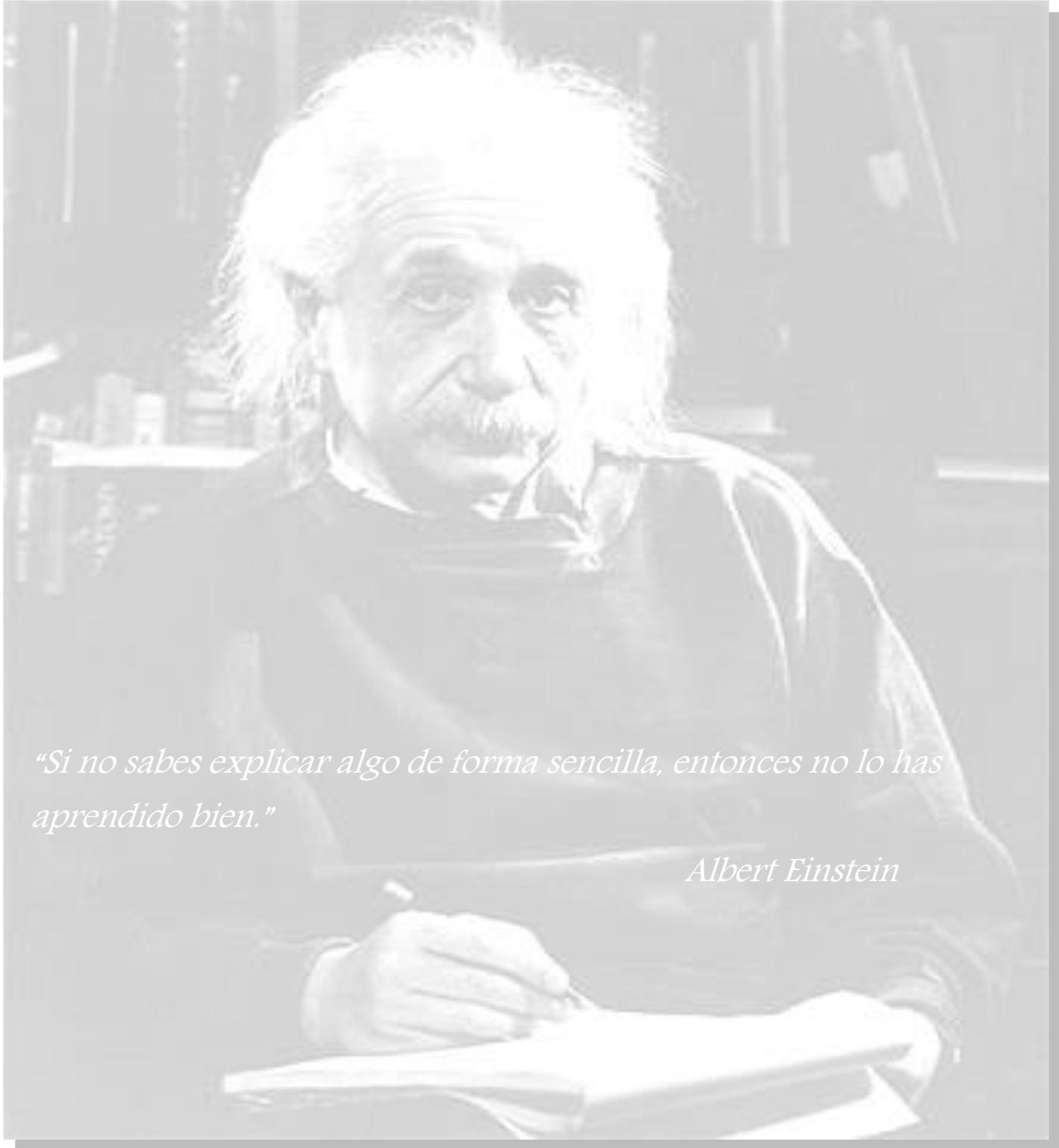
Título: Módulo para la gestión de boletines en aplicaciones web desarrolladas con Drupal 7.

Autor: Roberto Verdecia Conde

Tutores: MSc. Sahilyn Delgado Pimentel
Ing. Ramón Morales Álvarez

La Habana, Cuba

2016



“Si no sabes explicar algo de forma sencilla, entonces no lo has aprendido bien.”

Albert Einstein

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo al Centro de Ideoinformática de la Universidad de las Ciencias Informáticas, para que hagan el uso que estimen pertinente con este trabajo. Para que así conste firmo la presente a los _____ días del mes de _____ del año _____.

Firma del Autor
Roberto Verdecia Conde

Firma del Tutor
MSc. Sahilyn Delgado Pimentel

Firma del Tutor
Ing. Ramón Morales Álvarez

*A mi mamá, a mi papá a Juan Carlos y a mi
hermanita...para ellos, para que se sientan orgullosos.*

*A todos los amigos que de uno forma u otra han
ayudado a lograr este triunfo.*

*A mis tutores Ramón y Sahilyn, y a mí que
cuando nadie creía que yo podía, yo estaba seguro que
sí.*

RESUMEN

El departamento Soluciones Informáticas para Internet (SENIT) del Centro de Ideoinformática (CIDI) de la Universidad de las Ciencias Informáticas no cuenta con un boletín informativo para los portales web que desarrolla, por lo que se decidió desarrollar un módulo para la gestión de boletines informativos. Dicho módulo contribuirá a la mejora en la difusión de noticias de interés y la inmediatez de la información generada en los portales.

Bajo la metodología Agile Unified Process (AUP) el módulo fue diseñado e implementado con las herramientas y tecnologías usadas en la actualidad para el desarrollo de portales web, como son: CMS Drupal versión 7.x, PHP 5, CSS 3, Bootstrap 3.3, jQuery 1.9, Ajax, JavaScript 1.8, como servidor web se seleccionó Apache 2.2, como Sistema Gestor de Base de Datos PostgreSQL 9.4, el entorno de desarrollo integrado seleccionado fue NetBeans 8.0 y como herramienta de modelado VisualParadigm 8.0.

La aplicación posee funcionalidades específicas que permiten concentrar un grupo de características en el módulo desarrollado, que simplifica, facilita y agiliza su uso. Se le realizaron un grupo de pruebas al sistema, las cuales arrojaron resultados satisfactorios, asegurando la calidad del software desarrollado. El módulo para la gestión de boletines cumple con las características y funcionalidades mencionadas en la propuesta de solución por lo que se recomienda su pronta liberación para su uso en el centro.

Palabras clave: aplicación, boletines informativos, Drupal, entorno, gestión, portal web.

ÍNDICE

CAPÍTULO: 1.	Antecedentes históricos de la gestión de boletines informativos.....	5
1.1	Conceptos asociados	5
1.2	Características de un boletín informativo	5
1.3	Estructura de la información de un boletín informativo	6
1.4	Clasificación de boletines informativos	8
1.5	Herramientas para la gestión de listas de suscriptores	9
1.6	Herramientas para la confección de boletines informativos	11
1.7	Herramientas para el desarrollo	12
1.8	Lenguajes de programación	16
1.9	Tecnologías	18
1.10	Metodología de desarrollo	20
1.11	Conclusiones parciales	20
CAPÍTULO: 2.	Análisis y diseño del módulo para la gestión de boletines informativos.	22
2.1	Propuesta de solución.....	22
2.2	Modelo de dominio	22
2.3	Requisitos Funcionales	23
2.4	Requisitos no funcionales	28
2.5	Estilo arquitectónico	29
2.6	Patrones de diseño.....	31
2.7	Diagrama de clases del diseño	32
2.8	Conclusiones parciales	34
CAPÍTULO: 3.	Implementación y validación del módulo para la gestión de boletines informativos. .	35
3.1	Estándares de codificación.....	35
3.2	Modelo de datos	37
3.3	Diagrama de despliegue	38
3.4	Validación de la propuesta de solución	39
3.5	Conclusiones parciales	48
Conclusiones		49
Recomendaciones.....		50
Bibliografía		51
Anexos		55

ÍNDICE DE FIGURAS

FIGURA 1: DIAGRAMA DE CLASES DEL MODELO DE DOMINIO.....	23
FIGURA 2: CAPAS DE DRUPAL.....	31
FIGURA 3: DIAGRAMA DE CLASES DEL DISEÑO PARA SUSCRIBIRSE A BOLETÍN.....	33
FIGURA 4: MODELO DE DATOS.....	38
FIGURA 5: DIAGRAMA DE DESPLIEGUE.....	39
FIGURA 6: COMPORTAMIENTO DE NO CONFORMIDADES POR ITERACIÓN.....	44
FIGURA 7: RESULTADOS DE LAS PRUEBAS DE RENDIMIENTO.....	47

ÍNDICE DE TABLAS

TABLA 1: REQUISITOS FUNCIONALES.	24
TABLA 2: H.U. CREAR BOLETÍN.	25
TABLA 3: H.U. EDITAR BOLETÍN.	26
TABLA 4: H.U. SUSCRIBIRSE A BOLETÍN.	27
TABLA 5: CASO DE PRUEBA DE H.U. CREAR BOLETÍN.	41
TABLA 6: CASO DE PRUEBA DE H.U. EDITAR BOLETÍN.	41
TABLA 7: CASO DE PRUEBA DE H.U. SUSCRIBIRSE A BOLETÍN.	42
TABLA 8: RESULTADOS DE LA APLICACIÓN DE LA LISTA DE CHEQUEO PARA PRUEBAS DE SEGURIDAD.	45
TABLA 9: H.U. INSTALAR MÓDULO.	55
TABLA 10: H.U. DESINSTALAR MÓDULO.	55
TABLA 11: H.U. ELIMINAR BOLETÍN.	56
TABLA 12: H.U. MOSTRAR BOLETÍN.	57
TABLA 13: H.U. ENVIAR BOLETÍN.	57
TABLA 14: H.U. CANCELAR SUSCRIPCIÓN.	58
TABLA 15: H.U. MOSTRAR AYUDA DEL MÓDULO.	58
TABLA 16: H.U. MOSTRAR LA CANTIDAD DE USUARIOS QUE ACCEDEN DESDE UN BOLETÍN.	59
TABLA 17: H.U. MOSTRAR LOS BOLETINES MÁS UTILIZADOS.	60
TABLA 18: H.U. LISTAR LOS CONTENIDOS MÁS ACCEDIDOS DESDE UN BOLETÍN.	61
TABLA 19: H.U. LISTAR LOS TIPOS DE CONTENIDOS MÁS ACCEDIDOS DESDE UN BOLETÍN.	62
TABLA 20: H.U. MOSTRAR RESUMEN DE SUSCRIPTORES.	62

INTRODUCCIÓN

En las últimas décadas, las nuevas herramientas tecnológicas de la información y la comunicación han producido un cambio profundo en la manera en que los individuos se comunican e interactúan en el ámbito de los negocios. El uso de Internet está ampliamente extendido y es beneficioso para cualquier institución contar con presencia en la red, así como afrontar el reto de migrar sus servicios y aplicaciones a un entorno web que permita a sus usuarios acceder a ellos a través de la red. El correo electrónico en la actualidad, se ha convertido en la principal herramienta que tienen las empresas para comunicarse constantemente de forma directa con las demás instituciones y personas. Diariamente se envían una gran cantidad de correos electrónicos en todo el mundo; lo que hace que se transforme en un medio de mercadotecnia idóneo para que las organizaciones puedan gestionar la relación con sus clientes mediante la captación, el desarrollo y la fidelización de los mismos.

Uno de los medios de mercadotecnia utilizados a través del correo electrónico que podemos mencionar son los boletines informativos, los mismos son publicaciones periódicas que se envían por correo electrónico de forma masiva a través de un servidor web y que tratan sobre uno o varios temas de interés para sus audiencias. Entre las ventajas que se puede adquirir mediante la inserción de un boletín informativo en un portal web son: publicidad, mercadeo, invitación a eventos, promociones, encuestas y notificaciones (Uzcátegui, 2010). En el ámbito empresarial, la función principal del boletín informativo es ofrecer a sus clientes información interesante, novedosa, así como nuevos productos y servicios sobre el sector donde opera. Además, es una vía de comunicación directa entre usuarios e institución, con el objetivo de mejorar y fomentar lealtad de los clientes con la empresa, incrementar las ventas, así como adquirir nuevos clientes o convencer a los actuales que adquieran bienes o servicios de la entidad (Crexentia, 2014).

Contar con un boletín informativo por suscripción voluntaria permite establecer una presencia efectiva en la red. Al ser enviados periódicamente se convierten en un instrumento poderoso en los portales web donde son utilizados. Estos permiten desarrollar una comunidad de suscriptores estables, homogénea y creciente. Un boletín informativo, dotado de las características adecuadas tiene el potencial para aprovechar una parte importante del poder de compra de sus usuarios. Dichas características hacen de los boletines informativos una de las herramientas más sólidas, confiables y duraderas en cuanto a mercadotecnia en la actualidad (Mendoza, 2011).

En la Universidad de las Ciencias Informáticas (UCI), encargada de llevar la vanguardia en el desarrollo de

las nuevas tecnologías en el país, existen 14 centros de desarrollo de software. En el Centro de Ideoinformática (CIDI), precisamente en el departamento Servicios Informáticos para Internet (SENIT) se desarrollan portales web con el CMS Drupal 7.

El desarrollo de la web ha permitido que sea cada vez más frecuente que los sitios web, de instituciones o personales, cuenten con un boletín informativo debido a sus múltiples ventajas. El CMS Drupal 7 no está al margen de esta necesidad y para él se han desarrollado un grupo de módulos que permiten la gestión de boletines informativos, entre ellos se pueden destacar *Mime Mail*, *Simplenews* y *Newsletter*. Dicho departamento cuenta con la herramienta **Sympa** para la administración de listas de correo o *Mailing List Management* (MLM) por sus siglas en inglés, herramienta libre y de código abierto. Mientras que por otra parte no cuenta con un módulo para la gestión y creación única de boletines informativos con características propias y requeridas por los desarrolladores del departamento, como sería tener una funcionalidad que evite el envío de contenido repetido dentro de un mismo boletín informativo, que dicho boletín posea la capacidad de enviar archivos adjuntos, así como la posibilidad de que el usuario decida las categorías a las que desea suscribirse y no se vea atado a un boletín informativo con información que vaya más allá de sus preferencias, entre otras características ausentes en los módulos creados por el CMS Drupal 7.

Estos módulos requieren de un conocimiento avanzado para su configuración, resultando engorrosa y poco intuitiva para los administradores que la realizan en los portales donde son usados. Heredan un grupo de dependencias de otros módulos para realizar sus funciones, lo que trae como consecuencia que si uno de estos módulos de los que depende no funciona correctamente el módulo tampoco lo haga. También presentan problemas para la internacionalización, imposibilitando la interacción de usuarios de lenguas foráneas con el boletín informativo. Los módulos, además, contienen fallas en el envío de correos en formato **HTML**, lo que implica la degradación de la función principal del boletín informativo y la pérdida de interés de los usuarios que lo utilizan. Carecen de funciones que permitan enviar adjuntos, impidiendo el envío de información adicional y de interés en el boletín informativo. No generan reportes ni notificaciones con estadísticas de las suscripciones, suscriptores, ni el envío de contenidos, entorpeciendo un mejor análisis de la información generada por el boletín informativo. No posee una funcionalidad que impida enviar contenidos duplicados dentro de un mismo boletín, trayendo como consecuencias que el boletín informativo sea tratado por los servidores de correo electrónicos como **Spam** y no tengan el resultado esperado.

Por la problemática anteriormente expuesta el **problema científico** queda formulado por la siguiente interrogante: ¿Cómo gestionar los boletines electrónicos en aplicaciones web desarrolladas con Drupal 7 para

la divulgación de productos en los portales web en el departamento SENIT?

El **objeto de estudio** está constituido por la gestión de boletines informativos para aplicaciones web, enfocándose en el **campo de acción**: Gestión de boletines informativos para aplicaciones web en Drupal 7.

Para dar solución al problema se establece como **objetivo general**: Desarrollar un módulo para la gestión de boletines informativos en aplicaciones web desarrolladas con Drupal 7.

Como **objetivos específicos** se tienen:

- Construir el marco teórico conceptual de la investigación sobre la gestión de boletines informativos en aplicaciones web desarrolladas con Drupal 7.
- Definir las funcionalidades del módulo para la gestión de boletines informativos en aplicaciones desarrolladas con Drupal 7.
- Implementar las funcionalidades del módulo de gestión de boletines informativos en aplicaciones web desarrolladas con Drupal 7.
- Documentar las pruebas funcionales, rendimiento, seguridad e integración realizadas al módulo de gestión de boletines para aplicaciones web desarrolladas con Drupal 7.

En base al cumplimiento de los objetivos específicos se trazaron las siguientes **tareas de investigación**:

- Realización de un estudio sobre los sistemas para la gestión de boletines informativos.
- Selección de tecnologías, herramientas y estándares que se necesitan para implementar la propuesta de solución.
- Selección de la metodología de desarrollo.
- Definición de los requisitos funcionales y no funcionales de la propuesta de solución.
- Implementación de la propuesta de solución.
- Documentación de las pruebas realizadas al módulo.

Se toma como **Idea a defender**: La creación de un módulo para la gestión de boletines para el CMS Drupal 7 contribuirá a solucionar la divulgación de productos en los portales web en el departamento SENIT.

Métodos teóricos:

Histórico-Lógico: Se utilizó para realizar un estudio acerca de las soluciones informáticas relacionadas con el objeto de estudio, así como las tecnologías y estándares utilizados para su desarrollo.

Analítico-Sintético: Empleado para el análisis de las partes esenciales referentes a los documentos relacionados con el objeto de estudio. La puesta en práctica de este método permitió el estudio de soluciones que poseen funcionalidades relacionadas con la gestión de boletines electrónicos para aplicaciones web.

Modelación: Se empleó con el fin de modelar los procesos, así como crear los artefactos propuestos por la metodología usada para desarrollar la propuesta de solución.

Inductivo-Deductivo: El proceso de inducción permitió generalizar las características presentes en otros sistemas de gestión de boletines utilizados en el mundo, para deducir conclusiones asociadas a las necesidades de la investigación.

Método empírico:

Entrevista: Este método se utilizó con el fin de obtener información de cómo se lleva a cabo el proceso de gestión de boletines electrónicos para aplicaciones web desarrolladas en Drupal 7. Además de conocer el estado de opinión del cliente pudiendo determinar cuáles son las principales deficiencias de dicho proceso.

El documento está estructurado de la siguiente forma:

Capítulo 1: Antecedentes históricos de la gestión de boletines informativos.

Se tratan conceptos fundamentales y de interés relacionados con la investigación, se realiza un estudio de homólogos, además, se analizan las herramientas y tecnologías más utilizadas en la actualidad con sus características fundamentales, se describe el lenguaje y la metodología escogidos para la realización de la investigación.

Capítulo 2: Análisis y diseño del módulo para la gestión de boletines informativos.

En este capítulo se hace referencia a la propuesta de solución del problema planteado. Se realiza el análisis y diseño del módulo para la gestión de boletines informativos desarrollado con Drupal 7, se describe la solución propuesta para el problema de investigación abordado, así como los requisitos funcionales y no funcionales para el proceso de desarrollo de la aplicación.

Capítulo 3: Implementación y validación del módulo para la gestión de boletines informativos.

Se expone la solución que se le dará al problema, a través de un conjunto de artefactos y diagramas generados al sistema. Se presentan los resultados de las pruebas realizada a la solución propuesta comprobando el correcto funcionamiento y calidad del mismo.

CAPÍTULO: 1. **Antecedentes históricos de la gestión de boletines informativos.**

En el presente capítulo se mostrará una descripción de los lenguajes de programación, la metodología y herramientas usadas en la actualidad haciendo énfasis en las utilizadas para el desarrollo del módulo. Se definen los principales conceptos asociados al dominio del problema y que se hacen necesarios para entender el desarrollo de la investigación y se realiza un estudio del arte de las tendencias actuales de los boletines informativos.

1.1 Conceptos asociados

Boletín: De acuerdo con (TheFreeDictionary, 2015), un boletín es una publicación periódica con diferentes funcionalidades, publicada generalmente por una institución.

HTML: Son las siglas designadas para *Hyper Text Markup Language*, que traducido al español significa Lenguaje de Marcas de Hipertexto. HTML es uno de los lenguajes utilizados en la informática, cuyo fin es el desarrollo de las páginas web (ConceptoDefinición, 2015).

Spam: Es un término informático con el que se le conoce a todo el contenido no deseado y no necesariamente malicioso (contenedor de virus) que se presenta en nuestras cuentas de correo electrónico o de mensajería instantánea, incluso en nuestros teléfonos (ConceptoDefinición, 2015).

1.2 Características de un boletín informativo

Los boletines informativos son publicaciones enviadas por correo electrónico y contienen una o múltiples noticias de interés para sus suscriptores. Estos presentan un grupo de características mostradas a continuación, que permiten estar acorde con las tendencias actuales en la creación de boletines informativos y transmitir el mensaje deseado (Uzcátegui, 2010).

Es autorizado: se hace imprescindible pedir permiso a los clientes para poder obtener las direcciones de correo electrónico y así construir listas certificadas que aprueben las comunicaciones enviadas por la empresa; de lo contrario, se corre el riesgo de ser catalogado como *Spam* y de crear una percepción negativa de la institución.

Tiene una frecuencia apropiada: No se debe saturar este medio con una gran cantidad de comunicaciones; ya que el destinatario de dichos mensajes puede sentirse bombardeado, causando el efecto contrario. Lo recomendado es enviar dos boletines mensuales; pudiendo enviarse un tercero

con información que genere bienestar al receptor.

Está muy segmentado: En base a la información demográfica y a los estudios de hábitos de consumo, se pueden hacer segmentaciones en las listas de contactos con el fin de que el contenido sea de mayor relevancia e interés para dichos contactos.

Posee un título adecuado: El título es muy importante debido a que determina que un boletín sea leído o no. Para que sea efectivo; debe: plasmar la idea principal, tener una extensión de menos de 50 caracteres, llamar la atención y atraer a la persona que lo está leyendo.

El remitente es real: La empresa que obtuvo la autorización para enviar el boletín debe ser la que esté como remitente en el correo electrónico que se va a enviar; manteniendo así, el mismo nombre para que la persona que recibe la comunicación pueda reconocerlo más fácilmente.

El contenido es acertado: El boletín debe contener un mensaje que sea coherente, claro y sencillo. La información debe ser lo suficientemente oportuna y pertinente como para generar valor.

1.3 Estructura de la información de un boletín informativo

Los boletines informativos mediante los avances tecnológicos y el desarrollo web se han vuelto más sofisticados en su diseño, lo que permite que los usuarios tengan la posibilidad de descargar algunos elementos adicionales en sus correos electrónicos.

El boletín informativo debe estar pensado teniendo en cuenta que los usuarios del correo electrónico destinan solo unos segundos para ver el contenido de dicho boletín. El diseño visual del boletín informativo será la primera impresión que reciba el lector al abrirlo, esto hace que sea de suma importancia tener un boletín bien elaborado. Debe existir un equilibrio entre las imágenes que el usuario reciba y el texto. Estos tienen un poder de llegada sobre los usuarios muy importante, pero la imagen siempre llega primero al usuario, al igual que el diseño en su conjunto (Fernández, 2010).

Para que el boletín informativo alcance su objetivo, debe contar con algunas características esenciales que atraigan la atención del lector en forma rápida, para ello debemos conocer que el tiempo que emplean los lectores para determinar si les interesa el contenido o no, es muy bajo. Por lo que es una buena práctica colocar el contenido más atractivo en los sitios más visibles del boletín. Esto, al igual que en el diseño de una página web, es en la parte superior, y en el caso de que el ancho del boletín lo permita, a la izquierda.

La estructura de un boletín electrónico no debe contar con más de dos o tres secciones, en primer lugar, porque seguramente los lectores del boletín no sigan más allá de los dos o tres primeros contenidos del mismo, y luego porque se debe tener en cuenta el tamaño del correo electrónico, para evitar que el posible lector tenga inconvenientes para descargarlo y abrirlo, es beneficioso que se trate de un archivo más bien pequeño y con un contenido de buena calidad y bien dirigido. El contenido debe estar basado en no más de tres o cuatro temas diferentes. Siempre es conveniente que se cuente con un tema central más destacado y el resto como complemento. El tema central debe resultar particularmente interesante para los lectores, haciendo que luego de la lectura del mismo el usuario sienta deseos de obtener más información. Los títulos en este tipo de información reducida adquieren especial importancia. Los mismos deben resultar llamativos para que, tras la lectura, se desee continuar con el resto del contenido (Fernández, 2010).

1.3.1 Diseño gráfico de los boletines informativos

Cuando se intenta atraer la atención visual de alguien, uno de los métodos empleados es la utilización del color. Al igual que en el diseño de la página web, el color es empleado con la finalidad de atraer la atención de los usuarios y conseguir un efecto visual agradable. Se debe tener en cuenta que el empleo del mismo debe seguir ciertas pautas institucionales. Por ejemplo: si se cuenta con una web con una paleta de colores determinada, es conveniente que la misma paleta de colores sea empleada en el diseño del boletín que se envía a los usuarios de la misma, de forma tal que los mismos relacionen fácilmente al boletín con la página. También es importante observar que, en varias ocasiones, el color es mal empleado y se consigue resaltar y destacar algunas secciones que no son de principal interés. La utilización de colores en fondos y fuentes, debe hacerse haciendo especial hincapié en el contraste para facilitar la lectura del contenido textual. Bajo ningún concepto es deseable que el boletín cuente con más de dos o tres colores, excepto en casos especiales que la temática así lo requiera (Fernández, 2010).

La utilización de imágenes, si bien no es una práctica no recomendada, debe hacerse con cuidado, ya que las mismas son los elementos que más afecta en el tamaño de los archivos. Las imágenes entre medianas y pequeñas consiguen ilustrar el contenido, balancear el diseño y hacerlo más agradable. Las imágenes ilustrativas no deberían ser más de dos o tres y se debe definir un tamaño en pixeles para que las imágenes muy grandes se vean bien en el boletín y que no aumente su tamaño. Tampoco es adecuado la utilización de enlaces en las imágenes, ya que los filtros anti-spam suelen utilizar esto como indicio de la presencia de Spam (Fernández, 2010).

1.3.2 Utilización de fuentes para la creación de boletines informativos

Además de ser necesario un buen contraste entre el fondo y el texto, éste tiene que tener un tamaño adecuado. De ahí que sea necesario el empleo de fuentes adecuadas, evitando las dificultades de lectura. Utilizar más de tres tipos de fuentes diferentes puede hacer que el aspecto visual del diseño se vea afectado en forma negativa, por lo que es aconsejable que se empleen no más de tres fuentes diferentes. Se debe tener en cuenta que los contenidos van a estar en páginas más reducidas que una página web normal, por lo que una gran cantidad de variantes de un mismo elemento puede hacer que todo el diseño se vea mal (Fernández, 2010).

Las fuentes más adecuadas para los bloques de texto son del tipo definido por la institución que genera el boletín informativo, igualmente pasa con los títulos. Debe tenerse en cuenta que, al igual que en el desarrollo de una página web, conviene emplear aquellas fuentes que constituyen estándares, de forma que cualquier navegador en cualquier sistema operativo cuente con ellas (Fernández, 2010).

1.3.3 Bordes, líneas y separadores

Si el diseño requiere que se empleen cualquiera de los elementos antes mencionados, deberá tenerse especial cuidado que los mismos no resulten una atracción para la vista del lector, por lo que conviene que sean finos y de colores muy atenuados. Tampoco resulta muy atractivo ver un diseño lleno de cuadros, por lo que el empleo de los bordes debe ser cuidadoso y respetando las líneas generales del diseño (Fernández, 2010).

1.4 Clasificación de boletines informativos

Boletín de texto y sin formato, destinados a cualquier persona con correo electrónico ya que cualquier persona que posea una cuenta de correo electrónico puede leer estos boletines, tiene como ventajas, que son extremadamente rápidos, fáciles de componer y enviar, este es un formato ideal para el envío con urgencia de información sensible a tiempo. Tienen como desventajas que no son para nada llamativos debido a que los boletines de noticias de texto sin formato no pueden contener colores, gráficos u otros elementos de diseño. Los lectores rápidamente pueden perder el interés al leer fila tras fila de texto, sin interrupciones visuales (Boletines Electrónicos, 2015).

Boletín PDF, son destinados a todas aquellas personas con correo electrónico y acceso a internet, tiene pocas limitaciones, esto se debe a que los boletines PDF pueden ser creados para parecerse a casi cualquier publicación impresa, es el mejor formato electrónico para los usuarios que van a imprimir y leer el boletín de

noticias fuera de la pantalla, es fácil de convertir la mayoría de los boletines impresos a una casi idéntica publicación en formato PDF. Acarrear que todos los lectores deben tener un software especial instalado en sus ordenadores para ver e imprimir un boletín informativo en PDF, fotos y gráficos se suele guardar en baja resolución para recibir los boletines en PDF y reducir al mínimo el tiempo de descarga. Esto significa que las fotos de la publicación y los gráficos se pueden ver borrosos cuando se imprima el boletín, Los archivos PDF son más grandes que cualquier otro tipo de archivo del boletín en línea. Además, al enviar estos archivos como adjuntos de correo electrónico, algunos lectores pueden no ser capaces de recibirlos. Esto se debe a que algunos proveedores de correo electrónico suelen limitar el tamaño de los archivos adjuntos que reciben sus clientes (Boletines Electrónicos, 2015).

El **boletín web**, destinado a aquellos que tienen acceso a la Web, tiene como sus principales ventajas: que puede ser tan atractivo y llamativo como cualquier otra página web, y puede incluir color, fotografías y gráficos, siendo ideal para leer en pantalla. Sin embargo, los lectores deben tener acceso a Internet para ver la publicación. También debe recordarse que, no todo el mundo que tiene una cuenta de correo electrónico tiene una conexión a Internet. En comparación con un boletín por correo electrónico, un boletín de noticias web no es tan conveniente porque los lectores tienen que ir a un sitio de Internet para poder verlo en lugar de que se les entregue a ellos (Boletines Electrónicos, 2015).

El **boletín de correo electrónico HTML**, es el tipo más atractivo de boletín informativo por correo electrónico, ya que puede incluir fotos, gráficos, imágenes, colores y otros elementos de diseño. Son una buena alternativa a los boletines impresos con un costo más elevado. Siendo este tipo de publicación ideal para las personas que van a leer todo el boletín en la pantalla. No obstante, si el boletín contiene fotos o gráficos incluidos en el correo electrónico HTML, los lectores deben tener una conexión abierta a Internet al mismo tiempo que leen el correo electrónico para verlos, esto puede ser un problema para las personas que vean el boletín de noticias en el trabajo porque muchas empresas restringen el acceso de sus empleados a Internet. A diferencia de un boletín de noticias web, todos los artículos se deben poner en la misma página, eso implica que un boletín con un gran número de artículos debe ser visto e impreso como una larga página (Boletines Electrónicos, 2015).

1.5 Herramientas para la gestión de listas de suscriptores

La lista de suscriptores es la pieza clave de todo blog que se precie y que se base en la fidelización de lectores; sea comercial o no. Esta lista registra las personas que están realmente interesadas en el contenido,

conocimiento o porque el usuario crea que puede obtener algo de la información que se brinda en el portal (Alonso, 2016).

Mailchimp

Existen numerosas herramientas para confeccionar y administrar listas de suscriptores, una de estas es *Mailchimp*. Con gran reconocimiento a nivel mundial y fundada en el 2001, cuenta con alrededor de 7 millones de usuarios, los cuales envían de forma conjunta unos 10 mil millones de correos electrónicos mensualmente. Funciona en la mayoría de los navegadores solo habilitando las cookies, pop-ups y JavaScript. Si no existe una lista de suscriptores, *Mailchimp* genera automáticamente un formulario de suscripción muy configurable que se puede publicar donde lo crea conveniente el administrador, este formulario permite recopilar toda la información necesaria para generar la lista. La herramienta es bastante segura ya que posee un sistema de doble autorización, dicha herramienta consiste en que una vez el usuario llene el formulario de suscripción correctamente y lo envíe al administrador, este, lo acepta y le reenvía un correo electrónico con un enlace para confirmar su suscripción. Esto permite que los correos enviados a través de la lista de suscriptores no sean tratados como *spam*. No permite a piratas informáticos la creación de suscripciones falsas teniendo una lista de suscriptores estable y que la información llegue solo a los usuarios interesados (MailChimp, 2015).

Sympa

Sympa es el gestor de listas de correo electrónico usado en nuestra Universidad, es una plataforma bilingüe (español e inglés), presenta una interfaz web que permite la gestión fácil e intuitiva para usuarios y administradores de listas de correo. Contiene también una gestión automática de *bounce*¹. Mantiene un historial de mensajes enviados. Crea la posibilidad de habilitar la compartición de archivos en la web. La gestión de suscripciones (altas o bajas) es cómoda. Además, es un gestor de licencia libre lo que permite mantener las políticas de la institución sobre el uso de software libre (Universidad de Murcia, 2016).

MailChimp y **Sympa** son herramientas para la creación de listas de correos que poseen disímiles ventajas para la organización que las utilice. En el caso de la primera herramienta mencionada presenta algunas desventajas, ya que su versión gratuita permite hasta 2000 suscriptores, de ahí en adelante deja de serlo y hay que costear una licencia para su uso, **MailChimp** se encuentra en inglés solamente, siendo un problema

¹ Funcionalidad que permite la devolución de mensajes al remitente, dirigidos a los suscriptores cuya cuenta de correo dejó de existir.

para usuarios que no dominen dicho idioma, además, la creación del formulario de suscripción no es intuitiva dificultándole la confección del mismo a usuarios con poca experiencia en administración de listas de suscripciones. **Sympa** por otro lado es el gestor usado para la confección de listas de correos de nuestra Universidad, sin embargo, también presenta desventajas como: no exporta las listas de suscriptores en formato PDF, es poco intuitiva al momento de la confección de listas y el proceso de configuración resulta complicado sin un conocimiento previo en la gestión de listas de suscriptores.

1.6 Herramientas para la confección de boletines informativos

A continuación, se muestran las características de algunas herramientas para la confección de boletines informativos, y algunos módulos para el CMS Drupal que permiten la gestión y el envío de estos boletines informativos. Este análisis se hace con el objetivo de encontrar puntos comunes o elementos que puedan ayudar al desarrollo de este módulo que se desea implementar.

Microsoft Publisher

Es una herramienta para la gestión de boletines que permite crear, personalizar y compartir con facilidad una amplia variedad de publicaciones. Con Microsoft Publisher, se puede comunicar fácilmente el mensaje contenido en el boletín en diferentes tipos de publicaciones, lo cual permite ahorrar tiempo y dinero. Ya sea que se necesite crear folletos, boletines, postales, tarjetas de felicitación o boletines electrónicos, puede obtener resultados de alta calidad sin tener experiencia en diseño gráfico, permite el remplazo de imágenes y al mismo tiempo conservar el diseño de la página. Esta tecnología ofrece guías visuales para ayudar a alinear y ubicar el contenido sin esfuerzos, aunque el diseño final siempre queda en manos del diseñador. (Uzcátegui, 2010).

Mime Mail

Módulo desarrollado por los colaboradores del CMS Drupal con el que se le puede dar a los desarrolladores un grupo de mejoras como son: proporciona acciones del sistema simples para enviar correo electrónico con imágenes y archivos adjuntos, permite enviar mensajes en formato HTML, si el boletín presenta gráficos los incluye como archivo adjunto al enviarlo, si la dirección del destinatario está disponible y prefieren texto plano, el HTML se convertirá en texto plano y se envía tal cual. De lo contrario, el correo electrónico será enviado en formato HTML con una alternativa de texto plano. El módulo se adapta a las características y estilo del sitio (CSS) al incluirlo (Micka, 2005).

Newsletter

El módulo **Newsletter** tiene como objetivo la gestión de boletines. Está especialmente recomendado para personas que necesitan algo más que una lista de boletín de noticias para su sitio. Presenta un grupo de características como la de poseer una lista de suscriptores configurable, puede enviar boletines en formato HTML o texto plano, el boletín de noticias tiene una programación definida para su envío, contiene una administración centralizada y configuración en una misma página (Liakos, 2011).

Simplenews

Este módulo envía boletines informativos a una lista de usuarios suscriptos. Tanto los usuarios anónimos como los autenticados pueden optar por formatos de boletines diferentes. Soporta la traducción de nodos y taxonomías a varios idiomas. Los suscriptores pueden elegir el idioma en que recibirán el boletín (según la configuración de la cuenta). El módulo *Translate* es necesario para la traducción del boletín. Este módulo puede generar 2 tipos de boletines, los boletines de texto plano y los boletines en formatos HTML. (Dietiker, 2006).

El estudio realizado a las herramientas y módulos anteriormente mencionados evidencia que estos sistemas han sido creados para dar una solución genérica a la gestión de boletines informativos. Se aprovecharon algunas características de estos sistemas para el desarrollo de la propuesta de solución como en el caso de **Microsoft Publisher**, se pudo observar las pocas dependencias que genera, pero al no ser un software de licencia libre no se pudo estudiar el código fuente para un mayor análisis. De los módulos **MimeMail**, **Simplenews** y **Newsletter**, el autor se pudo nutrir con un grupo de características para la implementación del módulo para la gestión de boletines informativos a desarrollar, como fueron: la funcionalidad que permite el envío de boletines en formato HTML y en texto plano, la adaptabilidad de los módulos en diferentes entornos, la reutilización de código estudiando su código fuente. Como principales desventajas se les acarrea a estos módulos el gran número de dependencias que generan con respecto a otros módulos, imposibilitando su autonomía al momento de realizar su función.

1.7 Herramientas para el desarrollo

Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto de software que reúna los requisitos del cliente. Las herramientas para el desarrollo de software surgieron para intentar dar solución a los problemas de software mediante aplicaciones informáticas (Malavé, 2013).

1.7.1 CMS Drupal

Drupal en su versión 7, es un sistema de gestión de contenidos o *Content Management System*, (CMS) por sus siglas en inglés. Este se utiliza para crear sitios web dinámicos con gran variedad de funcionalidades. Drupal es un software libre, escrito en PHP, que cuenta con una amplia y activa comunidad de usuarios y desarrolladores que colaboran conjuntamente en su mejora y ampliación. Esta ampliación es posible gracias a que se trata de un sistema modular con una arquitectura muy consistente, que permite que los módulos creados por cualquier desarrollador puedan interactuar con el núcleo del sistema y con los módulos creados por otros miembros de la comunidad. Con Drupal es posible implementar una gran variedad de sitios web: un blog personal o profesional, un portal corporativo, una tienda virtual, una red social o comunidad virtual, etc (Buitrago, 2011).

1.7.2 Entorno de desarrollo

Netbeans 8.0

El ambiente de desarrollo integrado o entorno de desarrollo interactivo, en inglés *Integrated Development Environment* (IDE), es la principal herramienta de trabajo del programador, es el conjunto de herramientas y procesos de programación utilizados para crear el programa o producto de software. Es aquel en el que los procesos y las herramientas se coordinan para ofrecer a los desarrolladores una interfaz ordenada y cómoda visión del proceso de desarrollo, o al menos los procesos de la escritura de código, probándolos y empaquetándolos para su uso.

Como **IDE** se seleccionó el Netbeans por las ventajas que posee respecto a otros entornos y las características que lo distinguen como se muestran a continuación: permite a los desarrolladores escribir, depurar, compilar y ejecutar programas. Está escrito en Java, pero soporta otros lenguajes de programación como C, C++, Python, CSS, PHP, HTML y Ruby. Es libre y gratuito sin restricciones de uso, tiene una interfaz amigable e intuitiva, posee todas las herramientas para crear aplicaciones profesionales ya sean de escritorio, empresariales, web y móviles. Es multiplataforma y posee una creciente comunidad de usuarios. Cuenta con integración de base de datos, permitiendo escribir instrucciones de eliminar y actualizar. Sus funciones están previstas por módulos (NetBeans, 2015).

1.7.3 Servidor web

APACHE 2.2

Como la solución tiene que estar orientada a la web, se necesita hacer uso de un servidor web, el cual se

puede definir como un programa que escucha las peticiones de los usuarios o navegantes y las atiende o satisface, implementa el protocolo HTTPS (*Hypertext Transfer Protocol Secure*), este, se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones por parte de un cliente (navegador web), habiendo recibido dichas peticiones responde a estas adecuadamente y de forma segura, a través de una página web que se exhibirá en el navegador o mostrando un mensaje en caso de algún error.

De entre los tipos de servidores web que existen se escoge APACHE, por los beneficios que trae consigo, así como las características que lo hacen uno de los servidores más empleados del mundo. A continuación, se reflejan estas características y ventajas: flexible, rápido y eficiente. Puede ser utilizado en varios sistemas operativos. Continuamente actualizado y adaptado a los nuevos protocolos HTTP y HTTPS, modular, puede ser adaptado a diferentes entornos y necesidades, con los módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos. Tecnología gratuita, de código fuente abierto. Permite personalizar la respuesta ante los posibles errores que se den en el servidor. Permite la creación de archivos de registro a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor (Apache, 2013).

1.7.4 Sistema Gestor de Bases de Datos

PostgreSQL 9.4

Es un sistema gestor de bases de datos relacionales de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado gran popularidad dada su fiabilidad, integridad de datos y la corrección de los mismos. Es multiplataforma, ejecutándose en los principales sistemas operativos. Soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o vídeos. Es altamente escalable, tanto en la enorme cantidad de datos que puede manejar como en el número de usuarios concurrentes que puede acomodar. Por sus características, garantiza la integridad de los datos, la velocidad de acceso y consultas a la base de datos. Incorpora una estructura de datos y permite la declaración de funciones propias. Permite la gestión de diferentes usuarios concurrentes, además de los permisos asignados a cada uno de ellos (PostgreSQL, 2013).

1.7.5 Administración de Bases de Datos

PgAdmin III

Es una herramienta de código abierto para la administración de sistemas gestores de bases de datos

PostgreSQL y derivados (*EnterpriseDB, PostgresPlus, Advanced, Server y Greenplum Database*). La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible para varios sistemas operativos, incluyendo Microsoft Windows, Linux, FreeBSD, Mac OSX y Solaris.

Presenta varias características, a continuación, se muestran algunas de ellas:

- Interfaz administrativa y gráfica.
- Herramienta de consulta SQL.
- Editor de código procedural.
- EL panel de administración está diseñado para responder a las necesidades de la mayoría de los usuarios, desde los que escriben simples consultas SQL hasta los que desarrollan bases de datos complejas (Cartes, 2015).

1.7.6 Visual Paradigm

Visual Paradigm para el Lenguaje Unificado de Modelo o UML por sus siglas en inglés. Es una herramienta para el desarrollo de aplicaciones utilizando modelado UML ideal para ingenieros de software, analistas y arquitectos de sistemas que están interesados en construcción de sistemas a gran escala que necesiten confiabilidad y estabilidad durante el proceso de software. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad, hace posible la documentación del proyecto, mapeo relacional de objetos para Java, .NET y PHP. Permite construir diagramas de diversos tipos, código inverso y generar código desde diagramas (software, 2015).

1.7.7 Acunetix Web Vulnerability Scanner

Es una herramienta capaz de escanear sitios web en busca de posibles fallos de seguridad que puedan poner en peligro la integridad de la página publicada en Internet. Esta aplicación ejecuta una serie de pruebas, totalmente configurables por el usuario, para identificar las vulnerabilidades tanto en la programación de la página como en la configuración del servidor (Acunetix, 2015).

1.7.8 Apache Jmeter

La aplicación Apache Jmeter es un software de código abierto, creado en java y diseñado para probar el comportamiento funcional de un sitio web y medir su rendimiento. También puede ser utilizada para probar el rendimiento en recursos estáticos y dinámicos, lenguajes dinámicos como PHP, Java, ASP.NET, objetos

Java, bases de datos y consultas. Se puede utilizar para simular una carga pesada en un servidor, grupo de servidores, la red o el objeto para probar su resistencia o para analizar el rendimiento general bajo diferentes tipos de carga (Jmeter, 2016).

1.8 Lenguajes de programación

Un lenguaje de programación es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, también para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación (CCM, 2016).

1.8.1 Lenguajes del lado del cliente

La programación del lado del cliente permite la validación de datos en la máquina cliente antes de enviarlos al servidor. Esto proporciona informes inmediatos de errores a los usuarios en caso de alguna falla mientras siguen en esa página de formulario y sin necesidad de volver atrás tras recibir un mensaje de error. Puede resultar necesario acceder a una base de datos para validar determinados valores, aunque no suelen disponer de un acceso directo a la base de datos desde la máquina del cliente (Web, 2015).

Existen varios lenguajes del lado cliente entre los que se encuentran HTML, CSS, Java Script, Applets de Java, Visual Basic Script, Flash, se fundamentan a continuación los elegidos para el desarrollo del sistema por las características y ventajas que lo destacan de acuerdo al entorno en el que se está trabajando.

Java Script 1.8

Java Script es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débiles y dinámicos. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de Java Script del lado del servidor. Java Script se diseñó con una sintaxis similar al lenguaje de programación C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y Java Script no están relacionados y tienen semánticas y propósitos diferentes. Todos los navegadores modernos interpretan el código Java Script integrado en

las páginas web. Para interactuar con una página web se provee al lenguaje Java Script de una implementación del Modelo de Objetos del Documento o *Document Object Model* **DOM** por sus siglas en inglés (Arias, 2008).

JavaScript es una excelente solución para poner en práctica la validación de datos de un formulario en el lado del cliente. Si un usuario omite escribir un campo obligatorio en el formulario, una función de validación en JavaScript puede desplegar en pantalla un mensaje *popup* o alerta para hacerle saber a este acerca de la omisión. Este tipo de funcionalidades son más ventajosas que tener una rutina de validación del lado del servidor para controlar el error, dado que el servidor en este caso no tiene que hacer ningún tipo de procesamiento de información adicional. Una rutina en PHP podría ser escrita para lograr la misma tarea, pero un formulario desarrollado en JavaScript no permitiría que la información se enviase a menos que se complete correctamente el formulario (Duarte, 2012).

CSS 3

Las hojas de estilo en cascada o *Cascading Style Sheets* (CSS) por sus siglas en inglés, es una tecnología muy usada para el diseño web, es de gran utilidad para el programador web permitiéndole darles formato y estilo a varias páginas web al mismo tiempo (Bos, 2015). Esta versión de CSS mejorará la accesibilidad del documento, reduciendo la complejidad de su mantenimiento y permitiendo visualizar el mismo documento en infinidad de dispositivos diferentes. Las nuevas características de CSS 3 respecto a sus versiones anteriores incluyen soporte para selectores adicionales, sombras, esquinas redondeadas, múltiples fondos, animaciones y transparencia, permitiendo una mayor elegancia y estética en aquellas aplicaciones donde se utilice (Khalid, 2014).

1.8.2 Lenguajes de programación del lado del servidor

Son aquellos lenguajes que son reconocidos, interpretados y ejecutados por el mismo servidor, y enviados al cliente en un formato comprensible para él (Web, 2015).

PHP 5.5.9

El lenguaje de programación del lado servidor que se utilizará será PHP por las siguientes características: es el lenguaje en que está programado el CMS Drupal, es gratuito e independiente de la plataforma, además rápido, con una gran librería de funciones y mucha documentación. Desarrollado en política de código abierto. Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos. Capacidad de conexión con la mayoría de los

manejadores de base de datos: MySQL, PostgreSQL, Oracle, entre otras. Capacidad de expandir su potencial utilizando módulos. Soporta en cierta medida la orientación a objeto, clases y herencia. Está diseñado específicamente para ser un lenguaje más seguro y permita escribir programas en una interfaz de entrada común o *Common Gateway Interface* (CGI) por sus siglas en inglés, siguiendo algunas prácticas correctas de programación (de la Torre, 2015).

1.9 Tecnologías

La programación para muchos es considerada un arte, pero en sus inicios se tornaba un poco difícil para la mayoría de las personas. Con el tiempo se han ido descubriendo y desarrollando formas, guías generales, estilos y tecnologías de programación, con el objetivo de resolver aquellos problemas (Gonzalez, 2015).

AJAX

El término AJAX se presentó por primera vez el 18 de febrero de 2005 por Jesse James Garrett. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación web que estaba apareciendo. En realidad, el término AJAX es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML". El uso de AJAX permitirá mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano. Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor (LibrosWeb, 2015).

JQuery 1.9

jQuery es una biblioteca de JavaScript, creada por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Estas son algunas de sus características:

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- Soporta extensiones.
- Presenta varias utilidades, como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- Compatible con los navegadores Mozilla Firefox, Internet Explorer, Safari, Opera y Google Chrome (Alvarez, 2012).

Bootstrap 3.3.6

Bootstrap es una herramienta *Open Source* para el desarrollo rápido de aplicaciones web de código abierto. Desarrollado por Twitter permite dar forma a un sitio web mediante librerías CSS que incluyen tipografías, botones, cuadros, menús y otros elementos que pueden ser utilizados en cualquier sitio web. Bootstrap es una excelente herramienta para crear interfaces de usuario limpias y totalmente adaptables a todo tipo de dispositivos y pantallas, sea cual sea su tamaño. Además, Bootstrap ofrece las herramientas necesarias para crear cualquier tipo de sitio web utilizando los estilos y elementos de sus librerías.

Desde su aparición se ha vuelto compatible con el desarrollo web *responsive*² y entre otras características se han reforzado las siguientes:

- Soporte adecuado (casi completo) con HTML5 y CSS3, permitiendo ser usado de forma flexible para el desarrollo web con excelentes resultados.
- Establece *Media Queries* para 4 tamaños de dispositivos diferentes dependiendo del tamaño de su pantalla.
- También permite insertar imágenes *responsive*, es decir, con solo insertar la imagen con la clase “img-responsive” las imágenes se adaptarán al tamaño.

Bootstrap es compatible con la mayoría de los navegadores web del mercado, y más desde la versión 3. Actualmente es totalmente compatible con los siguientes navegadores:

² Capacidad de adaptar la apariencia de las páginas web a la resolución de los dispositivos donde se estén visualizando.

- Google Chrome (en todas las plataformas).
- Safari (tanto en iOS como en Mac).
- Mozilla Firefox (en Mac y en Windows).
- Internet Explorer (en Windows y Windows Phone).
- Opera (en Windows y Mac).

Todas estas características hacen que Bootstrap sea una excelente opción para desarrollar aplicaciones web totalmente adaptables a cualquier tipo de dispositivo (Fontela, 2015).

1.10 Metodología de desarrollo

AUP-UCI

Para todo proceso de desarrollo de software es necesario contar con una metodología de desarrollo que lo guíe. En el mundo no existe una metodología estándar para guiar todos estos procesos de desarrollo, ya que las metodologías deben ser consecuentes y configurables a las necesidades de las organizaciones donde se lleve a cabo el desarrollo de software. En nuestra universidad se decidió realizar una variación de la metodología del Proceso Unificado Ágil o *Agile Unified Process* (AUP) por sus siglas en inglés, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la universidad.

La nueva versión de la metodología consiste en tres fases: Inicio, Ejecución y Cierre. En Inicio se lleva a cabo la planeación del proyecto. Se realiza un estudio inicial de la organización que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo, costo y decidir si se ejecuta o no el proyecto. La fase de Ejecución se divide en las disciplinas Requisitos, Análisis y Diseño, Implementación, Pruebas internas, Pruebas de liberación y Pruebas de aceptación. En la última fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto (eXcriba, 2016).

1.11 Conclusiones parciales

Los diferentes sistemas homólogos estudiados presentan características y funcionalidades que pueden ser utilizadas en la confección del módulo. Dicho estudio permitió identificar estilos y tendencias actuales en el desarrollo de listas de suscriptores y módulos para la confección de boletines informativos.

La definición de las herramientas y tecnologías, así como de los lenguajes de programación tanto del lado del cliente como del servidor, bajo la supervisión de la metodología para el desarrollo de software sentaron

las bases para el diseño e implementación del módulo para la gestión de boletines informativos.

CAPÍTULO: 2. Análisis y diseño del módulo para la gestión de boletines informativos.

En este capítulo se explica la propuesta de solución, se identifican las funcionalidades específicas del módulo que ocurren a la hora de generar el boletín a través de la modelación del negocio, además se realiza un levantamiento de requisitos funcionales y no funcionales a partir del modelo construido, basándose en los objetos relevantes del problema para que guíen el proceso de desarrollo de software.

2.1 Propuesta de solución

El módulo para la gestión de boletines informativos, permitirá la administración completa de todos los datos de los usuarios que se suscriban a dicho boletín. El módulo podrá crear, modificar y eliminar categorías en la suscripción de un usuario. También mostrará reportes y notificaciones de la cantidad de usuarios suscriptos a un mismo boletín, los boletines más usados, los contenidos más accedidos desde un boletín, los tipos de contenidos más accedidos desde un mismo boletín y un resumen de suscriptores. Además, posibilitará el envío de archivos adjuntos dentro de un mismo boletín para complementar la información en caso de ser necesaria.

2.2 Modelo de dominio

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan los eventos que suceden en el entorno en el que trabaja el sistema. En el modelo de dominio se describe mediante diagramas UML (especialmente mediante diagramas de clases). De manera general el modelo del dominio ayuda a los usuarios, clientes, desarrolladores e interesados a utilizar un vocabulario común para poder entender el contexto en que se sitúa el sistema. En este caso se decidió realizar un modelo de dominio, ya que no se logró identificar con claridad los procesos del negocio, con fronteras bien establecidas (Larman, 2016).

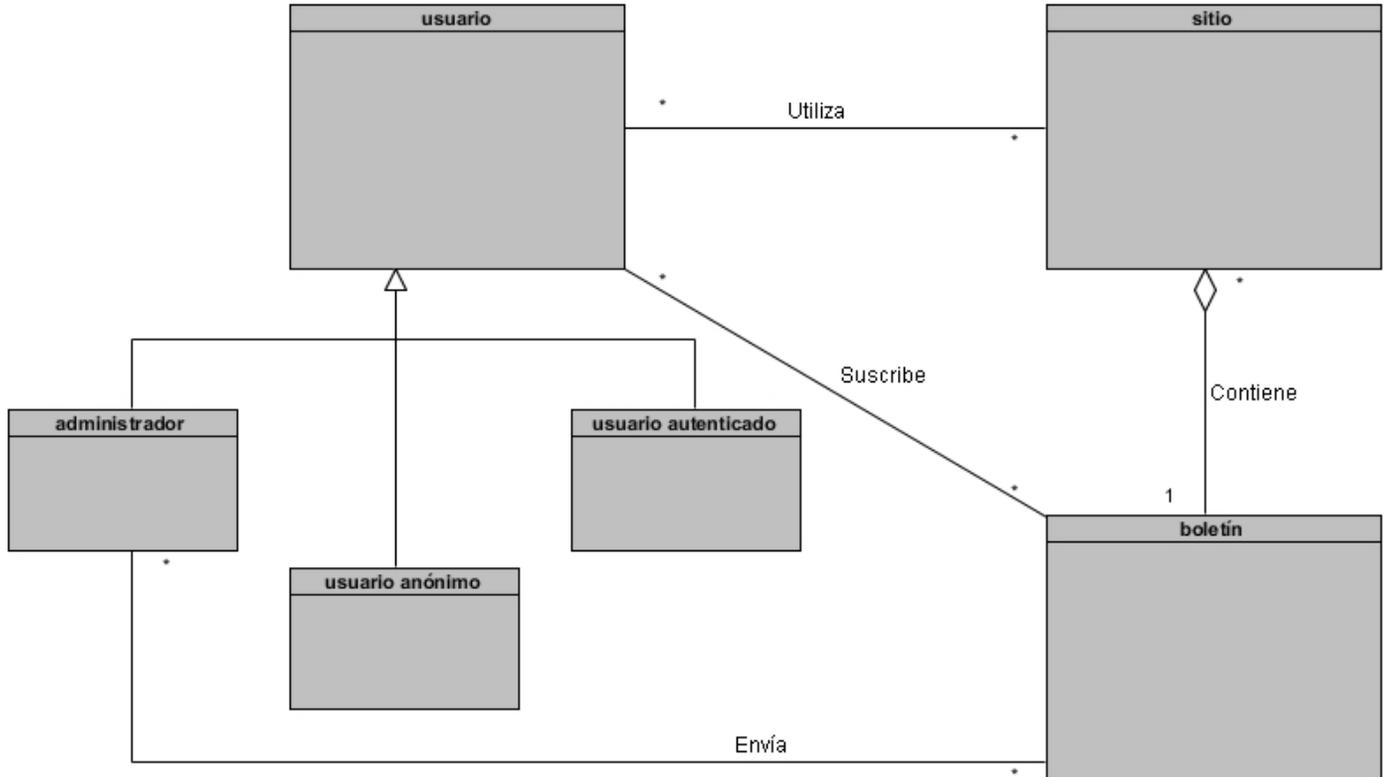


Figura 1: Diagrama de Clases del Modelo de Dominio.

2.2.1 Descripción de Clases del Modelo de Dominio

A continuación, se describen las clases que intervienen en el proceso y sus relaciones:

Usuario: Grupo de clientes con atributos y permisos determinados.

Sitio: Portal web en que se desplegará el módulo implementado.

Boletín: Módulo implementado con características requeridas.

Administrador: Usuario con permisos plenos en el módulo desarrollado.

Usuario autenticado: Usuarios con permisos definidos por el administrador para el módulo.

Usuario anónimo: Usuarios con permisos definidos por el administrador para el módulo.

2.3 Requisitos Funcionales

La ingeniería de requisitos del software es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refinan en detalle los requisitos del sistema y el papel asignado al software. El análisis de requisitos es una tarea de ingeniería del software que cubre el vacío entre la definición del software a

nivel sistema y el diseño de software. El análisis de requerimientos permite al ingeniero de sistemas especificar las características operacionales del software (función, datos y rendimientos), indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software (Pressman, 2002).

A continuación, se definen un grupo de requisitos funcionales (**RF**) que se determinaron para el desarrollo del software:

Tabla 1: Requisitos funcionales.

RF1	Instalar módulo.
RF2	Desinstalar módulo.
RF3	Crear boletín.
RF4	Editar boletín.
RF5	Eliminar boletín.
RF6	Mostrar boletín.
RF7	Enviar boletín.
RF8	Suscribirse a boletín.
RF9	Cancelar suscripción.
RF10	Mostrar ayuda del módulo.
RF11	Mostrar la cantidad de usuarios que acceden desde un boletín.
RF12	Mostrar los boletines más utilizados.
RF13	Listar los contenidos más accedidos desde un boletín.
RF14	Listar los tipos de contenidos más accedidos desde un boletín.
RF15	Mostrar resumen de suscriptores.
RF16	Mostrar resumen de cancelación de suscriptores.

2.3.1 Historias de usuario

La metodología **AUP-UCI** utiliza para describir los requisitos de software las Historias de Usuario. Estas utilizan las funcionalidades del sistema como entrada principal para su desarrollo. Contienen la información suficiente para que los desarrolladores pueden producir una estimación razonable del esfuerzo para desarrollar una funcionalidad (eXcriba, 2016). Los tiempos de estimación y real así como las prioridades dadas a los requisitos que se muestran en las Historias de Usuarios, están definidos en el

departamento SENIT. A continuación, se muestran las historias de usuario utilizadas en la propuesta de solución, el resto se muestran en los **Anexos**:

Tabla 2: H.U. Crear boletín.

Historia de usuario	
Número: 3	Nombre del requisito: Crear boletín
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración
Prioridad: Alta	Tiempo Estimado: 0.38 días
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días
<p>Descripción: Los usuarios que tengan rol administrador o editor pueden crear boletines, para ello deben llenar los siguientes campos:</p> <ul style="list-style-type: none"> ➤ Título (Obligatorio. Campo de texto. Long min 1 caracter. Permite todos los caracteres) ➤ Descripción (Opcional. Área de texto. Long min 1 caracter. Permite todos los caracteres) ➤ Contenidos (Obligatorio. Campo de autocompletamiento. Permite todos los caracteres) 	
<p>Observaciones:</p> <ul style="list-style-type: none"> ➤ Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha creado satisfactoriamente el boletín. ➤ Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error. ➤ Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben de llenarse. ➤ El campo “contenidos” solo admite valores que existan en el sistema. 	
Prototipo de interfaz	

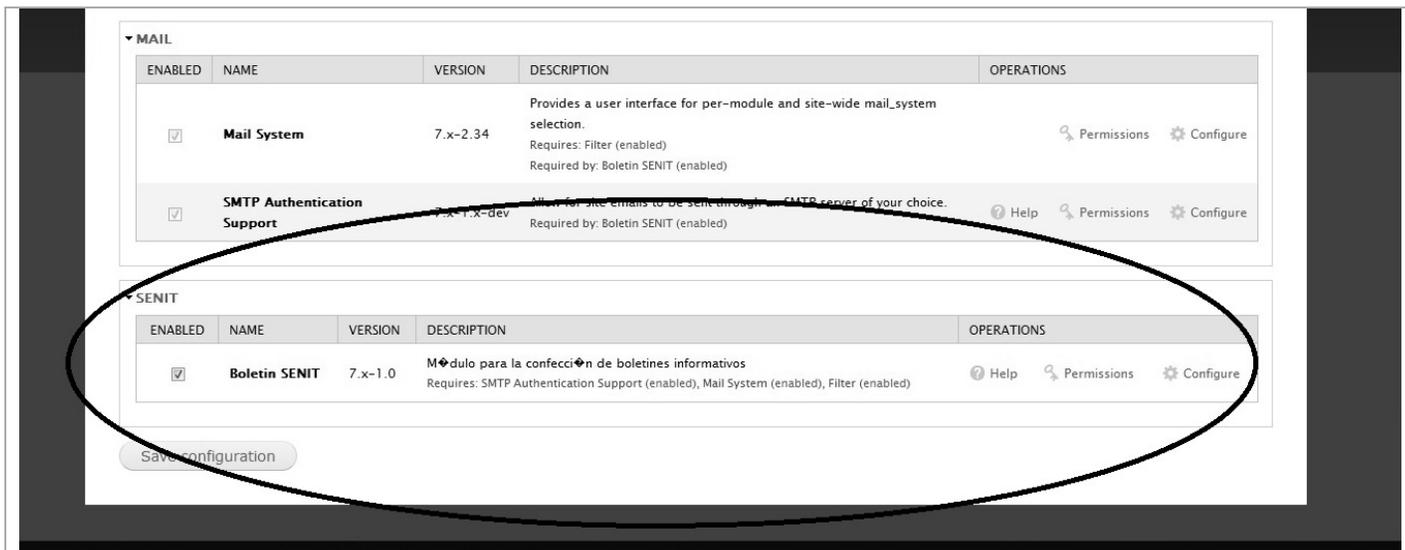


Tabla 3: H.U. Editar boletín.

Historia de usuario	
Número: 4	Nombre del requisito: Editar a boletín
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración
Prioridad: Media	Tiempo Estimado: 0.25 días
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días
<p>Descripción: Los usuarios que tengan rol administrador o editor pueden editar los boletines existentes en el sistema, modificando todos sus campos.</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> ➤ Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha editado satisfactoriamente el boletín. ➤ Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error. ➤ Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben de llenarse. 	
Prototipo de interfaz:	

Boletín SENIT Dashboard **Content** Structure Appearance People Modules Configuration Reports Help Hello admin Log out

SHOW ONLY ITEMS WHERE

status any Filter

type any

UPDATE OPTIONS

Publish selected content Update

TITLE	TYPE	AUTHOR	STATUS	UPDATED	OPERATIONS
fgfdgd	Boletín SENIT	admin	published	05/30/2016 - 10:04	edit delete
ciencia	Boletín SENIT	admin	published	05/30/2016 - 09:45	edit delete
Prueba	Boletín SENIT	admin	published	05/30/2016 - 09:27	edit delete
dasdasd	Boletín SENIT	admin	published	05/30/2016 - 08:45	edit delete
sdasdasdas	Article	admin	published	05/30/2016 - 08:44	edit delete

Tabla 4: H.U. Suscribirse a boletín.

Historia de usuario	
Número: 8	Nombre del requisito: Suscribirse a boletín
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración
Prioridad: Baja	Tiempo Estimado: 0.13 días
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días
Descripción: Los usuarios pueden suscribirse a los boletines existentes en el sistema mediante su correo electrónico.	
Observaciones:	
<ul style="list-style-type: none"> ➤ La dirección de correo electrónico introducida debe ser válida, una vez realizada la suscripción, el sistema mostrará un mensaje con instrucciones. 	

Bloque para los boletines

Mantente informado con nuestros boletines

Correo *

Boletin SENIT

Ciencia

Cultura

Deporte

Sociedad

Operaciones

Darse de baja

Actualizar

Suscribirse

Prototipo de interfaz:

2.4 Requisitos no funcionales

Un requisito no funcional o atributo de calidad es en la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento (PRESSMAN, 2002).

A continuación, mostramos un grupo de requisitos no funcionales (**RNF**) que se determinaron para el módulo a desarrollar:

➤ Usabilidad

RNF1: Se debe lograr una estructura simple y lo más intuitiva posible para el bloque de suscripción.

RNF2: Se utilizarán tamaños de letra apropiados para personas con alguna discapacidad visual.

➤ Seguridad

RNF3: El sistema será utilizado solo por los usuarios con permisos acordes a su nivel de autorización, usuario administrador, usuario registrado y usuario anónimo.

➤ **Interfaz**

RNF4: El menú de configuración del módulo presentará una interfaz agradable, utilizando colores claros, en tonalidad azul.

RNF5: Los textos serán presentados en tipografías compatibles con varios sistemas operativos y se resaltarán los encabezados en negrita.

➤ **Operacional**

RNF6: El módulo necesitará un entorno de trabajo compuesto por:

- ✓ Se utilizará como servidor web Apache versión 2.2.7. o superior.
- ✓ Lenguaje de programación PHP versión 5.5.9. o superior.
- ✓ CMS Drupal versión 7.x. o superior.

➤ **Soporte y Mantenimiento**

RNF7: Las operaciones de mantenimiento se deberán realizar en el momento menos concurrido del sitio (pasada la media noche) y no deberán extenderse por más de 3 horas.

RNF8: El sistema funcionará en varios sistemas operativos como Windows XP,7,8,8.1 y 10, Linux en todas las distribuciones basadas en Debian, Ubuntu, OpenSuse y Fedora.

➤ **Hardware**

RNF9: El servidor de aplicaciones web y de base de datos deben poseer como mínimo un CPU Dual Core a 2.20 GHz con 2 Gb de ram.

2.5 Estilo arquitectónico

La afirmación de que el CMS Drupal presenta una arquitectura de n-capas, viene dada por la combinación de dos elementos fundamentales. El primero, que esta arquitectura es en realidad un estilo de programación donde el objetivo principal es separar los diferentes aspectos del desarrollo y permitir intercambiar porciones de la aplicación sin necesidad de modificarla completa. El segundo, que es la característica fundamental del CMS Drupal, es su adaptabilidad mediante la adición de módulos, que son al final porciones de la aplicación que se modifican, sin tener que realizar cambios en el resto de la aplicación, además de dividir su contenido en una serie de elementos básicos nodos (*nodes*), módulos (*modules*), bloques y menús (*blocks & menus*), permisos de usuario (*user permissions*) y plantillas (*templates*) según (Mifsud, 2011), como se describe en la Figura 2.

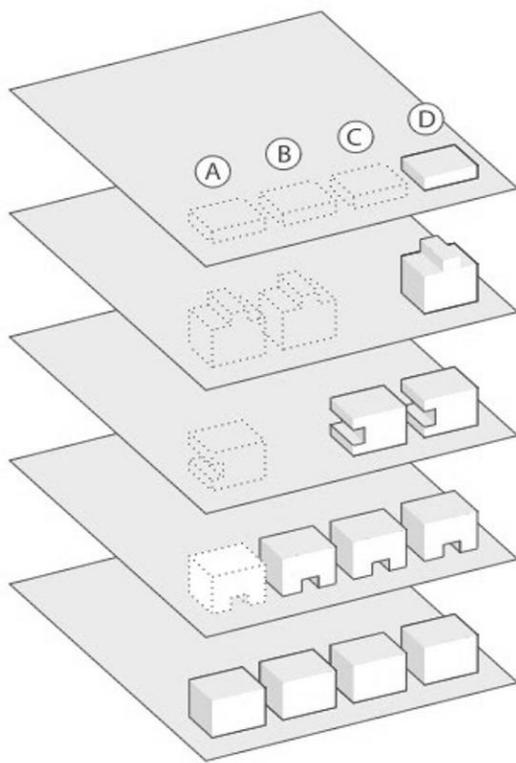
Capa 1 (Datos): Los nodos o datos son los elementos básicos en que Drupal 7 almacena la información del portal. Esto se puede ver evidenciado en el módulo desarrollado mediante las 3 tablas creadas en la base de datos del portal web donde se aloja dicho módulo.

Capa 2 (Módulos): Los módulos son los elementos que operan y otorgan funcionalidad a Drupal 7 permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada sitio web. Son como *Plug-Ins* que se instalan en el sitio web proporcionándole nuevas funcionalidades. El módulo para la gestión de boletines informativos desarrollado, se puede observar en la carpeta **NombreDelPortal/sites/all/modules/boletín_informativo**. Dentro de este directorio se puede encontrar un archivo con el propio nombre del módulo y **boletín_informativo.info**, en la que se pueden encontrar un grupo de características como las dependencias que genera, la versión y el bloque donde se encontrará ubicado.

Capa 3 (Menú y Bloques): Estos permiten estructurar y organizar los contenidos en la página web. La comunicación entre esta capa y la capa de nodos se puede ver a través del bloque de suscripción del módulo y los reportes que genera el mismo.

Capa 4 (Permisos de usuario): Es una capa importante en Drupal 7 pues controla los usuarios y los permisos. Actualmente, la mayor parte de sitios web son multiusuario, por lo que la seguridad y control de los usuarios es un punto clave para garantizar la integridad de la información almacenada. Los permisos generados para los 3 tipos de usuarios (administrador, autenticado y anónimo), pueden observarse en el panel de navegación del portal web donde se alojará el módulo desarrollado.

Capa 5 (Plantilla): Es la capa que establece la apariencia gráfica y el estilo de la información que se le muestra al usuario. Esta separación entre información y aspecto gráfico permite cambiar el diseño u apariencia del sitio web sin necesidad de modificar los contenidos, lo que es muy práctico si lo único que se quiere es renovar la apariencia de un sitio web. El módulo es perfectamente adaptable a las plantillas que contiene Drupal 7, por lo que no fue necesario la creación de nuevas plantillas.



5) Plantillas

4) Permisos de usuarios

3) Bloques y menús

2) Módulos

1) Nodos

Figura 2: Capas de Drupal.

2.6 Patrones de diseño

Drupal 7 hace uso de varios patrones de diseño, estos permiten diseñar sistemas seguros y que a su vez cumplan con los estándares de diseño establecidos por normas internacionales para el desarrollo de aplicaciones web según (Mifsud, 2011). Entre los que podemos mencionar:

Singleton (Instancia única): Los módulos en Drupal pueden considerarse como objetos que no encapsulan datos, lo que separa un módulo de otro es el conjunto de funciones que contiene, por lo que debe considerarse como una clase con una instancia *singleton*. (Matos, 2010). La propuesta de solución utiliza este patrón al ser un módulo para Drupal 7.

Observer (Observador): Este patrón es usado en Drupal 7 para controlar la forma en que se realizan las interacciones entre nodos, pues cuando se hace una modificación en un nodo, Drupal 7 notifica de este cambio a todos los nodos que se vean afectados por él. Los módulos que implementan un *hook* determinado por evento de inserción o actualización de una determinada entidad, son declarados como

observadores de dichas entidades con las que interactúan (Matos, 2010). En la propuesta de solución se utiliza el **hook _node_insert**, que se ejecuta al insertar un tipo de contenido en específico, por lo que el módulo es declarado como observador ya que se encuentra a la espera de un evento de inserción. Se utiliza además en la implementación del **hook_node_update**.

Command (Comando): Drupal 7 utiliza este patrón para reducir el número de funciones que son necesarias para la aplicación, pasando la operación como un parámetro, junto con los argumentos y así agilizar la ejecución de ciertas tareas en el sistema (Matos, 2010). Muchos de los *hook* implementados en la propuesta de solución utilizan el patrón de *command* para reducir el número de funciones que son necesarias para funcionar, pasando la operación como un parámetro, junto con los argumentos. El propio sistema de *hook* utiliza este patrón, por lo que los módulos no tienen que definir cada *hook*, sino más bien sólo los que se necesiten implementar.

Chain of Responsibility (Cadena de responsabilidades): El sistema de menús de Drupal 7 sigue este patrón. En cada solicitud de la página, el menú del sistema determina si hay un módulo para gestionar la solicitud y si el usuario tiene acceso a los recursos solicitados. Para ello, el mensaje se pasa a la opción del menú correspondiente a la vía de la solicitud. Si el elemento de menú no puede manejar la petición, se pasa a otro. Esto continúa hasta que un módulo se encarga de la petición, un módulo niega el acceso para el usuario, o la cadena se ha agotado (Matos, 2010). La implementación del **hook_menu** es la evidencia de la utilización de este patrón en la propuesta de solución.

Bridge (Puente): La capa de abstracción de base de datos de Drupal se implementa de una forma similar al patrón bridge, los módulos necesitan ser escritos en una manera que es independiente del sistema de base de datos que se utiliza, en el caso de la propuesta de solución la implementación de cada uno de los *hook*³ se realiza independiente del gestor de bases de datos utilizado (Matos, 2010). En el módulo desarrollado se utilizaron algunos de estos *hook* entre los que se encuentran el **hook_menu**, **hook_permission**, **hook_node_insert** y **hook_node_update** que realizan consultas a la base de datos sin importar de qué tipo sea esta.

2.7 Diagrama de clases del diseño

El diagrama de clase se elabora para tener en cuenta los detalles concretos de la implementación del sistema. Es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del

³ Funciones que permiten la comunicación con todo el núcleo de Drupal, pueden ser eventos u operaciones con campos, usuarios.

sistema se especifica con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones (Oliveros, y otros, 2015). A continuación, se representan los paquetes que conforman los módulos de Drupal.

Themes: Incluye las distintas plantillas. Cuando se desee incluir un nuevo diseño, se copia la plantilla dentro de esta carpeta.

Includes: Contiene un conjunto de ficheros indispensables para el correcto funcionamiento del CMS, como, por ejemplo, la capa de abstracción o conexión que provee las funcionalidades de acceso a la base de datos de Drupal.

Modules: Incluye todos los módulos. Cuando se desee incorporar un nuevo módulo solo se tiene que copiar en dicha carpeta.

Scripts: Contiene un conjunto de ficheros indispensables para el funcionamiento. Fundamentalmente orientados al aspecto visual, CSS y JavaScript.

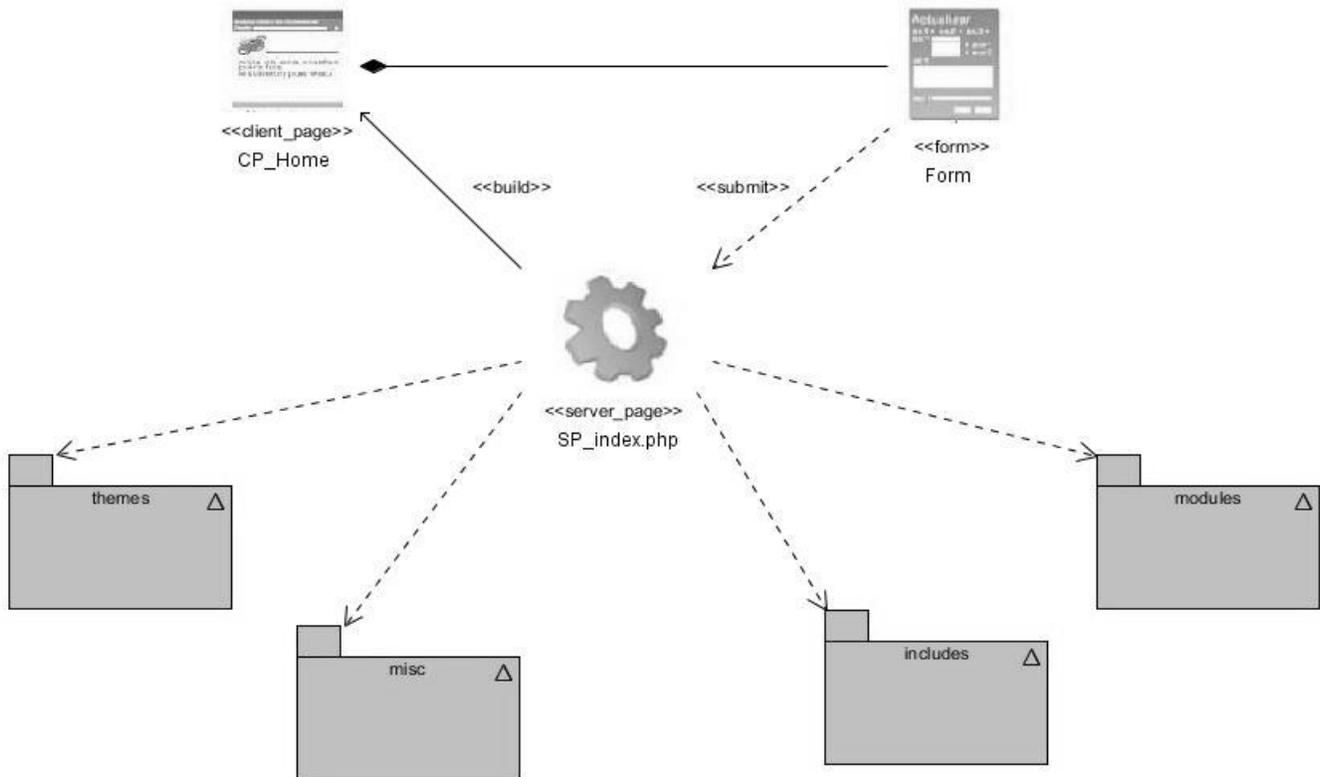


Figura 3: Diagrama de clases del Diseño para suscribirse a boletín.

2.8 Conclusiones parciales

Se realizó la descripción de la propuesta de solución logrando una mayor comprensión del sistema a implementar.

El levantamiento de requisitos funcionales y no funcionales sirvió de guía para el desarrollo de la aplicación, pues permitieron conocer el tiempo estimado para desarrollar las distintas funcionalidades y de este modo satisfacer las necesidades detectadas.

Los artefactos generados según la metodología de desarrollo utilizada, la arquitectura y los patrones de diseño descritos, constituyeron una guía fundamental para la construcción de la propuesta de solución.

CAPÍTULO: 3. Implementación y validación del módulo para la gestión de boletines informativos.

Desarrollar un sistema con la calidad requerida depende en gran medida de que se aplique correctamente un plan de pruebas. Las funcionalidades deben ser probadas desde el inicio y en cada una de las iteraciones dando seguimiento a los cambios en el sistema. El sistema desarrollado fue sometido a pruebas funcionales y de integración. En este capítulo se describe este proceso además de los diagramas del sistema, los estándares de codificación empleados durante la implementación, así como algunos de los resultados alcanzados.

3.1 Estándares de codificación

La forma de escribir código es propia de cada programador y completamente diferente a la forma de cualquier otro. De la forma usada depende la facilidad para entender el código y retomar ciertas partes realizadas por otros integrantes, así como la depuración de las mismas. Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Los usos de estos estándares ayudan a mejorar el proceso de codificación haciéndolo más eficiente, además de servirle a los programadores como punto de referencia. Para conseguir que el código se encuentre en buen estado y que cualquier persona del equipo pueda modificar cualquier parte de este, es imprescindible que el estilo de codificación sea consistente (Marín, 2003).

Etiquetas de apertura y cierre

Cuando se escribe en PHP, siempre se deben utilizar las etiquetas “<?php y ?>”, y en ningún caso la versión corta “<? y ?>”. En general se omite la etiqueta de cierre de PHP (?>) al final de los archivos .module y .inc. Esta convención evita que se puedan quedar olvidados espacios no deseados al final del archivo (después de la etiqueta de cierre ?>), que serían identificados como salida HTML y podrían provocar un error típico que genera el siguiente mensaje, “*Cannot modify header information-headers already sent by*”. Por tanto la etiqueta de cierre final del archivo (?>) es opcional en Drupal.

Uso de comillas

Se pueden usar tanto las comillas simples (‘cadena’) como las comillas dobles (“cadena”) para delimitar las cadenas de caracteres. Las comillas dobles son necesarias si se desean incluir variables dentro de las

cadenas de texto. Por ejemplo, “<h1>\$title</h1>”. También se recomienda el uso de las comillas dobles cuando el texto puede incluir alguna comilla simple.

Arrays

Los valores dentro de un *array* (o matriz) se deben separar por un espacio (después de la coma que los separa). El operador “=>” debe separarse por un espacio a ambos lados. Cuando la línea de declaración del *array* supera los 80 caracteres, cada elemento se debe escribir en una única línea. En este último caso, la coma de separación del último elemento también se escribirá, aunque no existan más elementos. De esta forma se evitan errores al añadir nuevos elementos al vector.

Uso de punto y coma (;) en código PHP

Aunque PHP permite escribir líneas de código individuales sin el terminador de línea (;), como por ejemplo “<?php print \$title ?>”. En Drupal es siempre obligatorio: “<?php print \$title; ?>”.

Concatenación de cadenas

Emplear siempre un espacio entre el punto y las partes concatenadas de una cadena para mejorar la legibilidad. Al concatenar variables simples se pueden usar comillas dobles y agregar la variable dentro. En otro caso, es necesario usar comillas simples. Al usar el operador de “concatenación-asignación” (‘.=’), también es necesario incluir un espacio antes y después del operador.

Operadores

Todos los operadores binarios (operadores que están entre dos valores), como +, -, +=, !=, ==, >, deben tener un espacio antes y después del operador, para facilitar la lectura.

Estructuras de control

Debe existir un espacio entre el comando que define la estructura (*if*, *while*, *for*) y el paréntesis de apertura.

- La llave de apertura ({) se situará en la misma línea que la definición de la estructura, separada por un espacio.
- Usar siempre las llaves { } aún en los casos en que no sea obligatorio su uso.
- Las estructuras *else* y *elseif* se escribirán en la línea siguiente al cierre de la sentencia anterior.

Declaración de funciones

Las funciones deben tener el nombre del grupo o módulo como prefijo, para evitar conflictos de nombres entre funciones de distintos módulos.

Variables

Las variables serán declaradas haciendo uso del guion bajo (`_`) para separar las palabras, del nombre de las variables que contenga más de dos palabras.

Nomenclatura de los módulos

El nombre de un módulo no debe contener guiones bajos para evitar conflictos de espacios de nombres.

Nomenclatura de los ficheros

Los nombres de los ficheros deben estar en minúsculas.

Nomenclatura de métodos

Los métodos y propiedades de clases deben usar "*lowerCamelCase*".

Nomenclatura de funciones

Los nombres de las funciones estarán en minúsculas y basados en el nombre del módulo del que forman parte. Los guiones bajos se utilizarán para separar las palabras descriptivas del nombre. Después del nombre del módulo, la función debe nombrarse con el verbo y el objeto a los que hace referencia.

Comentar el código

En este apartado se debe diferenciar entre los comentarios para aclarar determinados fragmentos de código, que se insertan en cualquier punto del mismo y los comentarios de documentación. Estos últimos, suelen escribirse al principio de un archivo o de cada función, y se utilizan para generar documentación de ayuda a través de aplicaciones que extraen la información a partir de las etiquetas empleadas.

En el primero de los casos se suelen emplear */** para comentarios en varias líneas y *//* para comentarios de una única línea. Se deben escribir frases completas, comenzándolas con mayúscula y terminándolas con un punto. En caso de que en el comentario se haga referencia a una constante, esta deberá escribirse en mayúscula (por ejemplo, TRUE o FALSE).

3.2 Modelo de datos

El modelo de datos es uno de los principales elementos del diseño de bases de datos que consiste en representar: objetos (entidades que existen y que se manipulan), atributos (características básicas de estos objetos) y relaciones (forma en que se enlazan los distintos objetos entre sí). Los modelos de datos

aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos (Codd, 2013). Seguidamente se muestra una descripción de las clases del modelo de datos:

Suscriptores: En esta tabla se guardan el correo electrónico del usuario que se ha suscrito al boletín, así como las categorías escogidas.

Contenido_enviar: En dicha tabla se almacenarán los identificadores de los contenidos a enviar, que no son más que los identificadores de las categorías que han seleccionado todos los usuarios que se han suscrito al boletín.

Contenido_enviado: Esta tabla es una especialización de **contenido_enviar**, en la que se almacena todo el contenido enviado en los boletines.

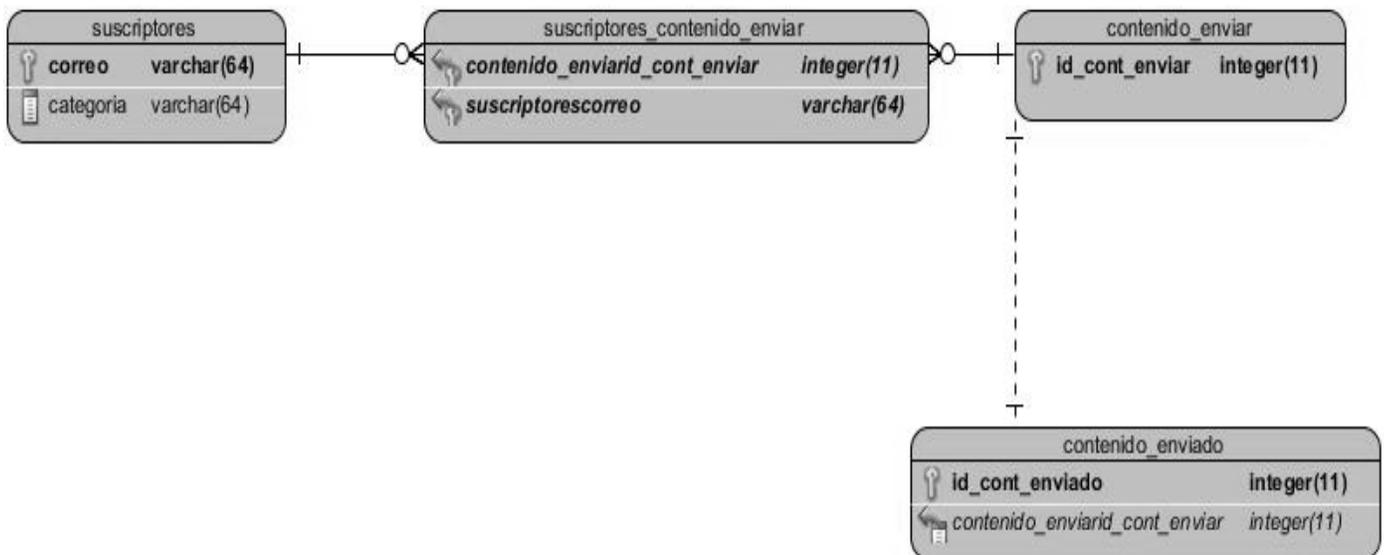


Figura 4: Modelo de datos

3.3 Diagrama de despliegue

El diagrama de despliegue es utilizado para capturar los elementos de configuración del procesamiento y las conexiones entre dichos elementos. Es utilizado también para visualizar la distribución de los componentes de software en los nodos físicos. La relación entre los nodos es denominada como protocolos de comunicación (Sarmientos, 2013).

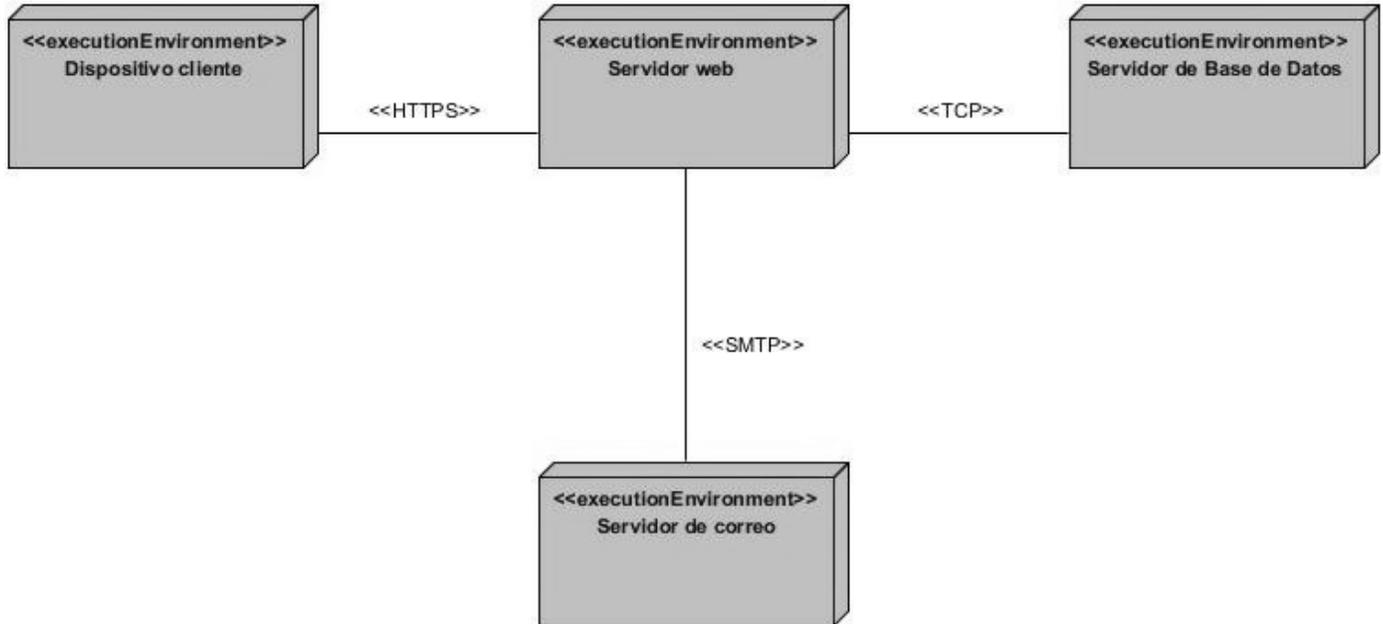


Figura 5: Diagrama de despliegue.

3.3.1 Descripción de los nodos.

Nodo Cliente: Representa una computadora desde la cual el usuario podrá acceder a la aplicación.

Nodo Servidor web: Representa una estación donde estará montado el servidor Apache sobre el cual correrá la aplicación a la cual accederán los usuarios a través de la máquina cliente.

Nodo Servidor de base de datos: Representa el servidor donde estará el Sistema Gestor de Base de Datos que dará respuesta a las peticiones hechas por la aplicación.

Nodo Servidor de correo: Representa el servidor de correo electrónico el cual prestará el servicio de envío del correo que contendrá el boletín.

3.4 Validación de la propuesta de solución

El aseguramiento de la calidad del software se ha convertido en una necesidad prioritaria y en una tarea vital en el desarrollo de cualquier sistema informático por la necesidad de garantizar que el producto cumpla con los requisitos especificados y que no presente errores. Por esta razón es necesario establecer una estrategia de prueba. Esta consiste en realizar un conjunto de pruebas como pruebas funcionales haciendo uso de la técnica de partición de equivalencia, pruebas de seguridad, pruebas de integración incremental y las pruebas de rendimiento para determinar el correcto funcionamiento del sistema. Existen

varios tipos de pruebas que suelen ser utilizadas, dentro de las que se pueden mencionar:

3.4.1 Pruebas funcionales

Las pruebas funcionales, son el objetivo final que debe cumplir la aplicación antes de ser liberada, su principal tarea es garantizar que los requerimientos hayan sido cumplidos y que el sistema sea operacional. Una de estas pruebas son las de **caja negra**, usando la técnica de partición de equivalencia se llevará a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa. Los casos de prueba de caja negra pretenden demostrar que:

- ✓ Las funciones del software son operativas.
- ✓ La entrada se acepta de forma correcta.
- ✓ Se produce una salida correcta.
- ✓ La integridad de la información externa se mantiene (Pressman, 2005).

Como resultado de las pruebas de aceptación se obtendrán artefactos descritos en tablas, estas contarán con los siguientes campos:

Código: identificador de la prueba realizada sugerente a la Historia de Usuario a la que hace referencia.

Nombre: nombre de la prueba a realizar.

Nombre del probador: nombre de la persona que realiza la prueba.

Descripción: se describe la funcionalidad que se desea probar.

Condiciones de Ejecución: mostrará las condiciones que deben cumplirse para poder llevar a cabo el caso de prueba, estas condiciones deben ser satisfechas antes de la ejecución del caso de prueba para que se puedan obtener los resultados esperados.

Entradas/Pasos de Ejecución: descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tiene en cuenta cada una de las entradas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.

Resultado esperado: breve descripción del resultado que se espera obtener con la prueba realizada.

Evaluación de la prueba: acorde al resultado de la prueba realizada se emitirá una evaluación sobre la misma. Esta evaluación tendrá uno de los dos valores que a continuación se describen:

- ✓ Satisfactorio
- ✓ Insatisfactorio

A continuación, se muestra una propuesta de diseño de caso de prueba por historias de usuarios, las demás se pueden encontrar en los **Anexos**.

Tabla 5: Caso de Prueba de H.U. Crear boletín.

Caso de Prueba		
Código: CDPHU1	Nombre: Crear boletín.	
Nombre del probador: Roberto Verdecia Conde.		
Descripción: Funcional.		
Condiciones de Ejecución: Estar autenticado y tener los permisos necesarios.		
<p>Entrada / Pasos de ejecución: El usuario accede al sistema.</p> <p>El sistema permite seleccionar una “Estructura” en la pestaña de administración “Bloque” donde se puede encontrar el bloque “Boletín SENIT” y se ubica en la parte deseada del portal.</p> <p>El sistema permite agregarle categorías al boletín haciendo clic en la opción “Contenido” dentro de la funcionalidad “Adicionar contenido” del boletín.</p>		
Escenarios:	Resultados Esperados	Evaluación de la Prueba:
EC1. Se agregan todas las categorías del boletín.	El sistema muestra el boletín en el lugar previamente seleccionado.	Satisfactoria.

Tabla 6: Caso de Prueba de H.U. Editar boletín.

Caso de Prueba	
Código: CDPHU4	Nombre: Editar boletín.
Nombre del probador: Roberto Verdecia Conde.	
Descripción: Funcional.	

Condiciones de Ejecución: Estar autenticado y tener los permisos necesarios.

Entrada / Pasos de ejecución: El usuario accede al sistema.

El sistema muestra en “Estructura”-> “Tipo de contenido”-> “Boletín SENIT” su contenido.

El usuario escribe el título al boletín, confecciona el cuerpo del boletín y selecciona el adjunto en caso de ser necesario.

El usuario guarda la configuración del boletín editado.

Escenarios:	Resultados Esperados	Evaluación de la Prueba:
EC1. Seleccionar las categorías a editar y modificar.	El sistema reconoce nuevos datos y si no hay ninguna incongruencia devuelve un mensaje “Su nueva configuración ha sido salvada”.	Satisfactoria.

Tabla 7: Caso de Prueba de H.U. Suscribirse a boletín.

Caso de Prueba	
Código: CDPHU8	Nombre: Suscribirse a boletín.
Nombre del probador: Roberto Verdecia Conde.	
Descripción: Funcional.	
Condiciones de Ejecución: Ser usuario del sistema donde esté desplegado el boletín.	

Entrada / Pasos de ejecución: El usuario accede al sistema.

El sistema muestra en la esquina inferior derecha un bloque con el boletín.

El usuario teclea su correo electrónico en un campo de texto obligatorio (**Correo***), selecciona las categorías a las que quiere suscribirse al boletín mediante unos *checkbox* y da clic en el botón Suscribirse.

El sistema envía un correo confirmando su suscripción.

Escenarios:	Resultados Esperados	Evaluación de la Prueba:
EC1. Seleccionar las categorías e incluir en el campo (Correo*) un correo válido.	El sistema almacena su correo y las categorías correspondientes en la base de datos para su posterior envío.	Satisfactoria.
EC2. Seleccionar las categorías e incluir en el campo (Correo*) un correo inválido.	El sistema se encarga de mostrar un mensaje donde advierte que la dirección de correo no es válida.	Insatisfactoria.
EC4. No seleccionar las categorías e incluir en el campo (Correo*) un correo inválido.	El sistema muestra un mensaje advirtiendo que no se pueden guardar los datos si no se introducen correctamente.	Insatisfactoria.
EC5. Seleccionar las categorías y dejar el campo (Correo*) en blanco.	El sistema muestra un mensaje advirtiendo que no puede dejar el campo (Correo*) en blanco.	Insatisfactoria.

Resultados de las pruebas funcionales

Para la validación de los requisitos funcionales se realizaron tres iteraciones de pruebas funcionales al módulo. En la Figura 6 se muestran los resultados obtenidos en cada una de las iteraciones de pruebas realizadas al módulo para la gestión de boletines.

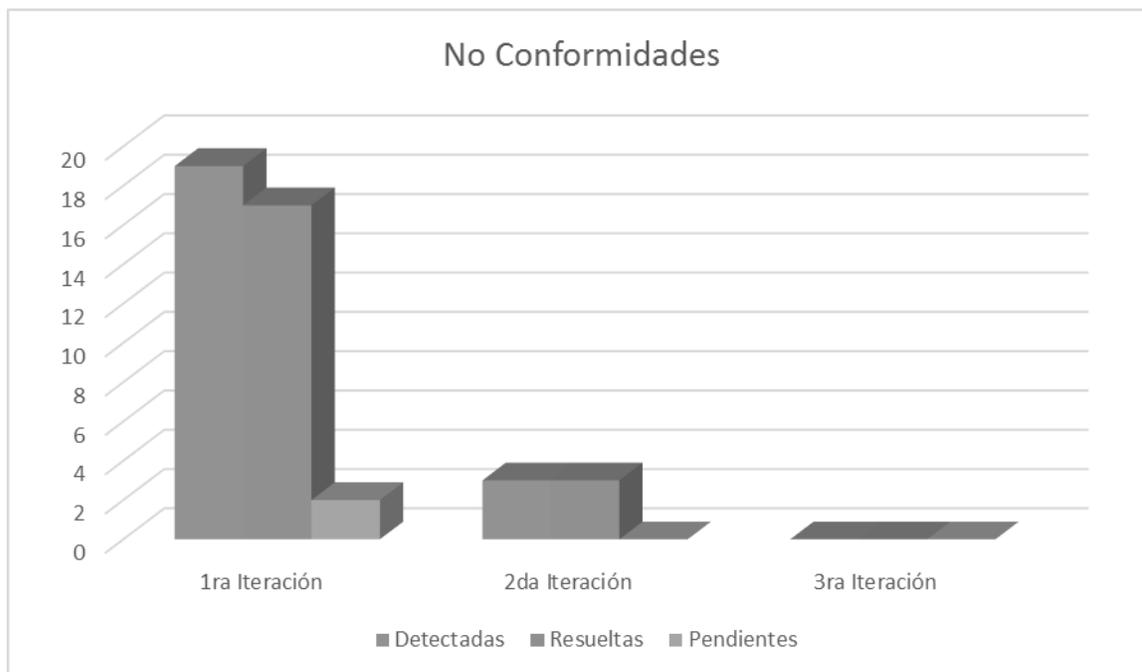


Figura 6: Comportamiento de no conformidades por iteración.

Se encontraron 19 no conformidades en la primera iteración de las cuales 17 fueron resueltas y quedando 2 no conformidades pendientes. En la segunda iteración se encontraron 3 no conformidades, una nueva y las restantes de la primera iteración, las cuales fueron resueltas en su totalidad. Se realizó una tercera iteración para la validación, arrojando como resultado 0 no existieron no conformidades en el módulo desarrollado.

Algunas de las no conformidades encontradas durante el proceso de prueba fueron las siguientes:

- Datos incorrectos guardados en las bases de datos sin validación previa.
- Mensajes de error que no corresponden con los errores definidos en los casos de pruebas.
- Errores ortográficos.
- Errores de concordancia.
- Errores de validación de los campos.

3.4.2 Pruebas de seguridad

Las pruebas de seguridad garantizan que los usuarios estén restringidos a funciones específicas o que su acceso esté limitado únicamente a los datos que están autorizados a acceder. Sólo aquellos usuarios

autorizados a acceder al sistema son capaces de ejecutar las funcionalidades disponibles. El objetivo fundamental de este tipo de pruebas es comprobar los niveles de seguridad lógicos del sistema (García, 2014).

Resultados de las pruebas de seguridad

Para evaluar la seguridad de la aplicación se hizo uso de la herramienta Acunetix WVS, la cual devolvió los siguientes resultados.

Tabla 8: Resultados de la aplicación de la lista de chequeo para pruebas de seguridad.

Categoría de vulnerabilidad	Cantidad
Pruebas de autorización	33
Pruebas de gestión de sesiones	2
El método TRACE está activado	6

Todos los errores corresponden al sistema Drupal donde se encuentra alojado el módulo y no constituyen problemas de seguridad para el mismo.

3.4.3 Pruebas de integración

La prueba de integración es una técnica sistemática para construir la estructura de un programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. Existen dos tipos de integración: no incremental e incremental. En el primer caso se combinan todos los módulos y se prueba el programa en su conjunto, como es lógico pensar el resultado puede ser caótico con un gran número de fallos y la consiguiente dificultad para identificar el módulo que los provocó. Por su parte, en la integración incremental el programa se construye y se prueba en pequeños segmentos en los que los errores son más fáciles de aislar y corregir (García, 2014).

Resultados de las pruebas de integración

Para el uso de la prueba se utilizó una técnica incremental, la misma permite probar el programa en pequeñas porciones haciendo más fácil encontrar errores. El módulo para la gestión de boletines desarrollado para Drupal 7, se integró en diferentes portales web como el Portal web de la FEU y al Portal web de la UJC. Estos presentan diferentes gestores de bases de datos como PostgreSQL y MySQL,

manteniendo el correcto funcionamiento de los portales y concluyeron que el módulo presentaba un funcionamiento correcto en todos los casos probados.

3.4.4 Pruebas de rendimiento

Mediante las pruebas de rendimiento es posible hallar tendencias y comportamientos para los elementos de una aplicación, los cuales generan bajo rendimiento. Este tipo de pruebas permiten identificar cuellos de botella, capacidad de concurrencia de usuarios, tiempos de respuesta de operaciones de negocio a nivel de sistema, establecer un marco de referencia para pruebas futuras, determinar el cumplimiento de los objetivos de rendimiento y requerimientos no funcionales, entre otros. **Pruebas de estrés:** Mediante las pruebas de estrés es posible identificar la capacidad de respuesta de un sistema bajo condiciones de carga extrema, representadas por una alta concurrencia de usuarios y/o procesos, una vez realizadas las pruebas de estrés se podrá conocer el punto de quiebre del aplicativo en términos de capacidad de respuesta, con lo cual será posible establecer acciones de optimización en diferentes niveles para asegurar una mejor capacidad de concurrencia de usuarios y/o procesos que se verá reflejada en una óptima operación de negocio. **Pruebas de carga:** Mediante la ejecución de las pruebas de carga es posible identificar la capacidad de recuperación de un sistema cuando es sometido a cargas variables tanto de usuarios como de procesos. Al realizar las pruebas de carga se puede determinar el tiempo de respuesta de todas las transacciones críticas del sistema y encontrar cuellos de botella de la aplicación (V&vquality, 2016).

Para realizar las pruebas de rendimiento en el sistema se empleó la herramienta Apache Jmeter, para un total de 50 muestras que se le hicieron al módulo, el tiempo mínimo de respuesta fue de 0 segundos y el máximo de 8 segundos.

Resultados obtenidos de las pruebas de rendimiento

Para las pruebas de rendimiento se utiliza el software Apache Jmeter v2.8.4. Para ello se definen las propiedades de las PC utilizadas tanto la cliente como la utilizada como servidor.

Hardware de prueba (PC-cliente):

- Tipo de procesador AMD a6 3.70GHz.
- Memoria Ram: 8 GB.
- Tipo de Conexión: Ethernet 100Mbps.

Hardware de prueba (PC-servidor):

- Tipo de procesador: AMD a6 3.70GHz.
- Memoria Ram: 8 GB.
- Tipo de Conexión: Ethernet 100Mbps.

Software instalado en ambas PC:

- Tipo de servidor web: Apache versión 2.2.
- Plataforma: SO Windows y GNU Linux versión Xubuntu.
- Servidor de BD: PostgreSQL versión 9.5.

Interpretación de resultados obtenidos de las pruebas de rendimiento

En la Figura 7 se muestran los resultados obtenidos con la herramienta, luego de definido el *hardware* y el *software* se configuran los parámetros del Apache Jmeter, simulando un total de 50 usuarios concurrentes con un período de suscripción de tres segundos, se esperaba un promedio de suscripción de 8 segundos y los resultados muestran un promedio de 4.8 segundos, demostrando así que el módulo desarrollado responde satisfactoriamente ante las pruebas realizadas.

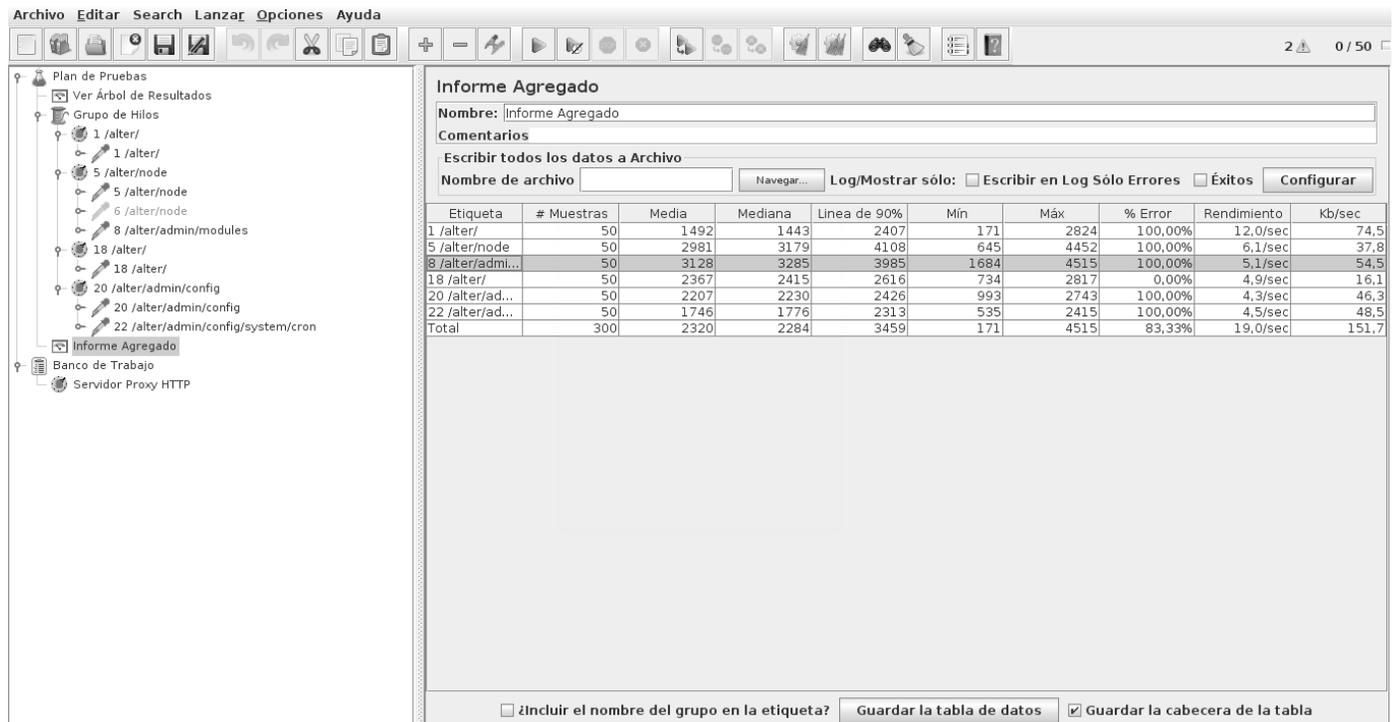


Figura 7: Resultados de las pruebas de rendimiento.

3.5 Conclusiones parciales

La utilización de estándares de codificación de código permitió adoptar una estructura homogénea que facilita la comunicación y asegura la calidad, menos errores y un fácil mantenimiento.

Se mostró a través del modelo de despliegue los recursos físicos necesarios para el despliegue del sistema.

Las pruebas realizadas al sistema permitieron corregir errores y fallos en el funcionamiento de la aplicación, así como validar el correcto funcionamiento del sistema, lo que permite asegurar que la propuesta de solución es estable y segura.

Conclusiones

Con el estudio de homólogos y la selección de las tecnologías y herramientas usadas durante el proceso de desarrollo del software, permitió una solución informática viable al problema planteado. Bajo una metodología ágil, simplista y concentrada en actividades de valor como lo es **AUP**, se establecieron las bases para el diseño e implementación del módulo para la gestión de boletines.

Una vez definida la propuesta de solución e identificados los requisitos funcionales y no funcionales, así como el estilo arquitectónico y los patrones de diseño, el proceso de desarrollo de software concurre con plena comodidad para el equipo de trabajo. Con las diferentes pruebas y técnicas utilizadas para validar el módulo desarrollado, quedó demostrado que dicho módulo es una solución funcional, segura, se integra adecuadamente y posee un rendimiento acorde con los resultados esperados.

Con el desarrollo del módulo, quedan cumplidos los objetivos de la investigación y es la solución adecuada para atender las necesidades del centro. El módulo se encuentra listo para crear boletines y sean enviados hacia sus destinatarios correspondientes, mejorando así la divulgación de la información dentro de cualquier portal web donde se instale dicho módulo.

Recomendaciones

Para optimizar el módulo para la gestión de boletines informativos y permitir una mayor aceptación por parte de los clientes una vez concluido el trabajo, el autor recomienda:

- ✓ Implementar una funcionalidad que permita al usuario definir la frecuencia con que desee recibir los boletines informativos.
- ✓ Generar una tabla en la base de datos que almacene todos los reportes generados por el módulo para la gestión de boletines desarrollado.

Bibliografía

- Acunetix. 2015.** Acunetix. *Acunetix*. [En línea] 2015. <http://www.acunetix.com/>.
- Alvarez, Miguel. 2012.** Nxnet. *Nxnet*. [En línea] 12 de 1 de 2012. [Citado el: 2 de 6 de 2016.] <https://nxnethosting.com/blog/jquery-que-es-que-ventajas-y-desventajas-tiene/>.
- Apache. 2013.** The Apache. *The Apache*. [En línea] 2013. [Citado el: 3 de 12 de 2015.] https://httpd.apache.org/docs/2.4/new_features_2_4.html.
- Arias, David. 2008.** catedraprogramacion. *catedraprogramacion*. [En línea] 16 de 10 de 2008. <http://catedraprogramacion.forosactivos.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos>.
- Bos, Bert. 2015.** CSS current work. [En línea] 2015. [Citado el: 5 de 12 de 2015.] <http://www.w3.org/Style/CSS/current-work>.
- Buitrago, Jorge. 2011.** Drupal. *Drupal*. [En línea] 11 de 5 de 2011. [Citado el: 14 de 4 de 2016.] <https://groups.drupal.org/node/148379>.
- Cartes, Cesar. 2015.** ArPuG. *ArPuG*. [En línea] 2015. [Citado el: 5 de 12 de 2015.] <http://www.postgresql.org.ar/trac/wiki/PgAdmin?format=pdfarticle>.
- CCM. 2016.** CCM. *CCM*. [En línea] 3 de 2016. [Citado el: 1 de 6 de 2016.] <http://es.ccm.net/contents/304-lenguajes-de-programacion>.
- Codd, Edgar. 2013.** Medievals Trucos. *Medievals Trucos*. [En línea] 18 de 7 de 2013. [Citado el: 5 de 5 de 2016.] <https://medievalstrucos.wordpress.com/2013/07/18/12-reglas-de-codd-para-bases-de-datos-relacionadas/>.
- ConceptoDefinición. 2015.** conceptodefinicion.de. *conceptoDefinicion.de*. [En línea] 26 de 1 de 2015. [Citado el: 31 de 5 de 2016.] <http://conceptoDefinicion.de/html/>.
- Crexentia. 2014.** Crexentia. *Crexentia*. [En línea] 28 de 1 de 2014. [Citado el: 30 de 11 de 2015.] <http://www.crexentia.com/la-importancia-del-boletin-informativo-en-la-web-corporativa/>.
- de la Torre, Anibal. 2015.** qdelat. [En línea] 2015. [Citado el: 5 de 12 de 2015.] http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
- Dietiker, Miro. 2006.** Drupal. [En línea] 6 de 7 de 2006. [Citado el: 2 de 12 de 2015.] <https://www.drupal.org/project/simplenews>.
- Duarte, Eugenio. 2012.** Capacity. *Capacity*. [En línea] 19 de 9 de 2012. [Citado el: 1 de 6 de 2016.]

<http://blog.capacityacademy.com/2012/10/19/que-es-javascript-ventajas-y-desventajas/>.

Electronicos, Boletines. 2015. Boletines Electronicos. *Boletines Electronicos*. [En línea] 2015. [Citado el: 30 de 11 de 2015.] <https://sites.google.com/site/uelbosqueboletineselectronicos/discusion>.

Eva Maria. 2010. Consumoteca. [En línea] 31 de 5 de 2010. [Citado el: 30 de 11 de 2015.] <http://www.consumoteca.com/telecomunicaciones/internet/newsletter/>.

eXcriba. 2016. eXcriba. *eXcriba*. [En línea] 2016. [Citado el: 2 de 6 de 2016.] <http://excriba.prod.uci.cu/page/user/rconde/>.

Fernández, Andrés. 2010. la webera.es. [En línea] 17 de 5 de 2010. [Citado el: 30 de 11 de 2015.] <http://www.lawebera.es/promocionar/disen-boletin-electronico.php>.

Fontela, Álvaro. 2015. Raiola Networks. [En línea] 2015. [Citado el: 6 de 12 de 2015.] <https://raiolanetworks.es/blog/que-es-bootstrap/>.

García, Ana. 2014. javiergarzas.com. *javiergarzas.com*. [En línea] 4 de 7 de 2014. [Citado el: 6 de 5 de 2016.] <http://www.javiergarzas.com/2014/07/tipos-de-pruebas-10-min.html>.

Gonzalez, Maria. 2015. Prezi. *Prezi*. [En línea] 1 de 31 de 2015. [Citado el: 1 de 6 de 2016.] <https://prezi.com/h9w8asfupsle/tecnologias-para-el-desarrollo-de-aplicaciones-web/>.

Jmeter, Apache. 2016. Apache Jmeter. *Apache Jmeter*. [En línea] 2016. [Citado el: 2 de 6 de 2016.] <http://jmeter.apache.org/>.

Khalid. 2014. exp3rto. *exp3rto*. [En línea] 25 de 8 de 2014. [Citado el: 1 de 6 de 2016.] <http://www.exp3rto.com/introduccion-a-css3-nuevas-caracteristicas-capacidades-y-recursos/>.

Larman, Craig. 2016. DSpace. [En línea] 2 de 2016. [Citado el: 17 de 2 de 2016.] <http://dspace.ucbscz.edu.bo/dspace/handle/123456789/13197>.

Liakos, Paris. 2011. Drupal. [En línea] 7 de 7 de 2011. [Citado el: 1 de 12 de 2015.] <https://www.drupal.org/project/newsletter>.

LibrosWeb. 2015. LibrosWeb. *LibrosWeb*. [En línea] 2015. [Citado el: 6 de 12 de 2015.] http://librosweb.es/libro/ajax/capitulo_1.html.

MailChimp. 2015. MailChimp Email Marketing and Email List Manager. [En línea] 2015. [Citado el: 30 de 11 de 2015.] <http://mailchimp.com/about/>.

mailpro. 2015. mailpro. [En línea] 2015. [Citado el: 30 de 11 de 2015.] https://es.mailpro.com/10_razones.asp.

- Malavé, Simón. 2013.** Prezi. *Prezi*. [En línea] 16 de 1 de 2013. [Citado el: 14 de 4 de 2016.] <https://prezi.com/cvlck6fg226/tecnicas-y-herramientas-para-el-desarrollo-de-software/>.
- Marín, David. 2003.** Estándares. [En línea] 5 de 5 de 2003. [Citado el: 14 de 4 de 2016.] <http://www.aspl.es/fact/files/aspl-fact/estandares-node2.html>.
- Matos, Liber. 2010.** POO y patrones de diseño en Drupal. [En línea] 2010. [Citado el: 16 de 2 de 2016.] <http://jitcode.blogspot.com/2010/06/poo-y-patrones-de-diseno-en-drupal.html>.
- Mendoza, Alvaro. 2011.** mercadeoglobal.com. *mercadeoglobal.com*. [En línea] 2011. [Citado el: 31 de 5 de 2016.] <http://mercadeoglobal.com/blog/ventajas-boletin-electronico/>.
- Micka, Allie. 2005.** Drupal. [En línea] 16 de 11 de 2005. [Citado el: 1 de 12 de 2015.] <https://www.drupal.org/project/mimemail>.
- Mifsud, Elvira. 2011.** Observatorio Tecnológico. [En línea] 7 de 12 de 2011. [Citado el: 16 de 2 de 2016.] <http://recursostic.educacion.es/observatorio/web/es/software/software-general/1022%20monograficodrupal-7?start=2>.
- Mirabal, Liss. 2012.** SlideShare. *SlideShare*. [En línea] 21 de 10 de 2012. [Citado el: 10 de 5 de 2016.] <http://es.slideshare.net/Lismirabal/requerimientos-funcionales-y-no-funcionales>.
- Murcia, Universidad de. 2016.** UM. *UM*. [En línea] 2016. [Citado el: 2 de 6 de 2016.] <https://www.um.es/atika/contenidos/listas/documentacion/nuevogestlist.html>.
- NetBeans. 2015.** NetBeans. [En línea] 2015. [Citado el: 2 de 12 de 2015.] <https://netbeans.org/community/releases/80/>.
- Oliveros, Alejandro y Antonelli, Rubén Leandro. 2015.** Sedici. *Sedici*. [En línea] 10 de 2015. [Citado el: 2 de 6 de 2016.] <http://sedici.unlp.edu.ar/handle/10915/50349>.
- PostgreSQL. 2013.** PostgreSQL. *PostgreSQL*. [En línea] 21 de 4 de 2013. [Citado el: 14 de 4 de 2016.] <http://www.postgresql.org/about/>.
- Pressman. 2002.** Análisis de requisitos del software. *Análisis de requisitos del software*. [En línea] 2002. [Citado el: 2 de 6 de 2016.] <http://yaqui.mx.l.uabc.mx/~molguin/as/IngReq.htm>.
- . 2005. BibliotecaUCI. *BibliotecaUCI*. [En línea] 2005. [Citado el: 2 de 6 de 2016.] <http://bibliodoc.uci.cu/pdf/reg02689.pdf>.
- Rouse, Margaret . 2015.** [En línea] 2 de 2015. [Citado el: 18 de 2 de 2016.] <http://whatis.techtarget.com/definition/framework>.

- Sarmientos, Johana. 2013.** umldiagramadespliegue. *umldiagramadespliegue*. [En línea] 7 de 4 de 2013. [Citado el: 2 de 6 de 2016.] <http://umldiagramadespliegue.blogspot.com/>.
- software. 2015.** software.com.ar. *software.com.ar*. [En línea] 2015. [Citado el: 18 de 2 de 2016.] <http://www.software.com.ar/p/visual-paradigm-para-uml>.
- TheFreeDictionary. 2015.** TheFreeDictionary. *TheFreeDictionary*. [En línea] 2015. [Citado el: 30 de 11 de 2015.] <http://es.thefreedictionary.com/bolet%C3%ADn>.
- Uzcátegui, Jhuliana . 2010.** PuroMarketing. *PuroMarketing*. [En línea] 2010. [Citado el: 30 de 11 de 2015.] <http://www.puromarketing.com/23/6852/caracteristicas-debe-tener-boletin-electronico-exitoso.html>.
- V&vquality. 2016.** V&vquality. *V&vquality*. [En línea] 2016. <http://vyvquality.com/pruebas-rendimiento/>.
- Web, Programacion. 2015.** Programacion Web. [En línea] 2015. [Citado el: 5 de 12 de 2015.] <https://programacionwebisc.wordpress.com/2-3-lenguajes-de-programacion-del-lado-del-servidor/>.

Anexos

Anexo 1

Tabla 9: H.U. Instalar módulo.

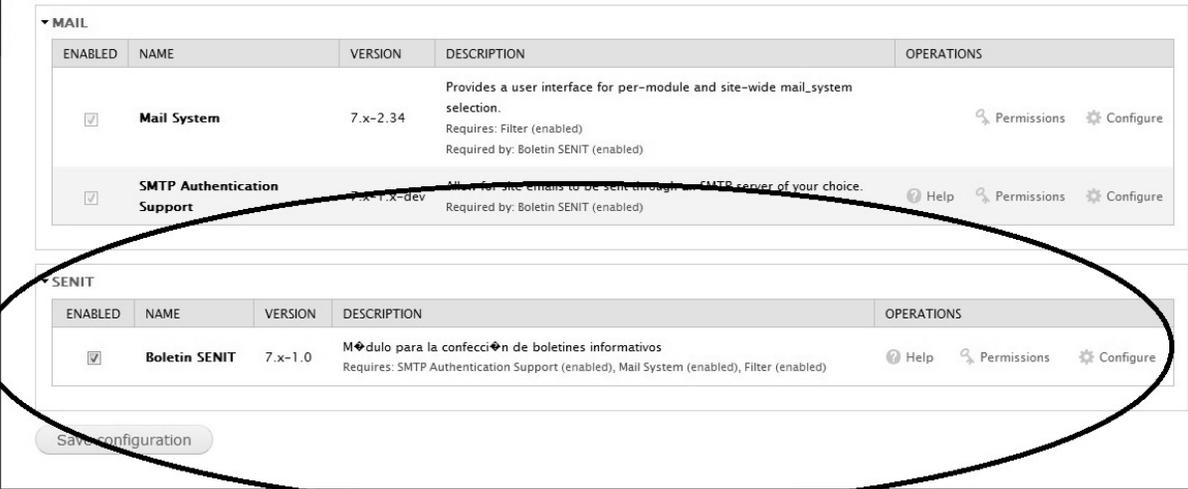
Historia de usuario	
Número: 1	Nombre del requisito: Instalar módulo.
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración
Prioridad: Media	Tiempo Estimado: 0.25 días
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días
Descripción: Se instala el módulo “boletín informativo” en el portal web.	
Observaciones: <ul style="list-style-type: none">➤ Si se instala el módulo correctamente el CMS Drupal lo muestra mediante un mensaje de aceptación.	
Prototipo de interfaz	
	

Tabla 10: H.U. Desinstalar módulo.

Historia de usuario	
Número: 2	Nombre del requisito: Desinstalar módulo.
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración
Prioridad: Media	Tiempo Estimado: 0.25 días
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días

Descripción: Se desinstala el módulo “boletín informativo” en el portal web.

Observaciones:

- Si se desinstala el módulo correctamente el CMS Drupal lo muestra mediante un mensaje de aceptación.

Prototipo de interfaz:

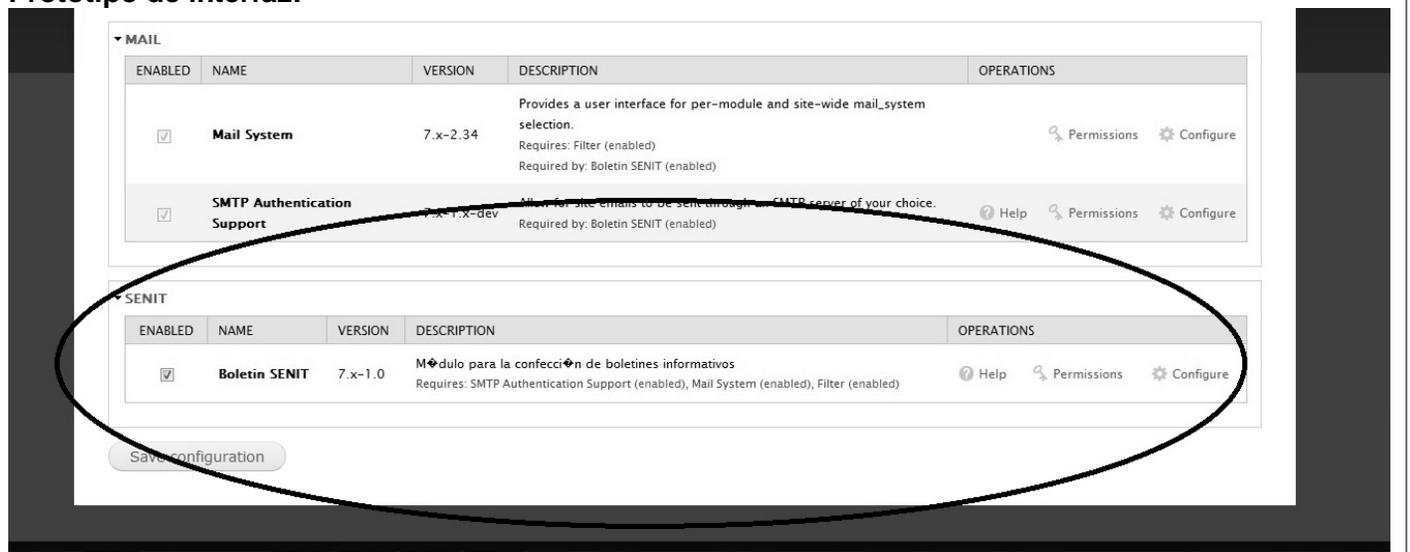


Tabla 11: H.U. Eliminar boletín.

Historia de usuario	
Número: 5	Nombre del requisito: Eliminar boletín.
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración
Prioridad: Media	Tiempo Estimado: 0.25 días
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días
Descripción: Al eliminar el boletín, se elimina la Taxonomía, el Tipo de Contenido y el Bloque que conforma el boletín.	

Observaciones:
Prototipo de interfaz: No aplica.

Tabla 12: H.U. Mostrar boletín.

Historia de usuario	
Número: 6	Nombre del requisito: Mostrar boletín.
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración
Prioridad: Media	Tiempo Estimado: 0.13 días
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días
<p>Descripción: Los usuarios que tengan rol administrador o editor podrán visualizar los boletines existentes en el sistema, en la interfaz de administración. Los usuarios anónimos podrán visualizar los boletines después de suscribirse, en su correo electrónico.</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> ➤ Los usuarios anónimos visualizarán los contenidos del boletín en su correo electrónico, donde se mostrarán con sus datos. ➤ Los usuarios autenticados visualizarán el boletín en la interfaz de administración, donde se mostrarán solo los enlaces de los contenidos que contiene. 	
Prototipo de interfaz: No aplica.	

Tabla 13: H.U. Enviar boletín.

Historia de usuario	
Número: 7	Nombre del requisito: Enviar boletín.
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración
Prioridad: Alta	Tiempo Estimado: 0.13 días
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días

<p>Descripción: El usuario con rol administrador podrá enviar los boletines a los usuarios que estén suscritos a los mismos.</p>
<p>Observaciones:</p> <ul style="list-style-type: none"> ➤ Para enviar el boletín al menos debe existir un usuario suscrito. ➤ Un boletín puede ser enviado solo una vez.
<p>Prototipo de interfaz: No aplica.</p>

Tabla 14: H.U. Cancelar suscripción.

Historia de usuario	
Número: 9	Nombre del requisito: Cancelar suscripción.
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración
Prioridad: Media	Tiempo Estimado: 0.13 días
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días
<p>Descripción: Los usuarios podrán darse baja de los boletines a los que se ha suscrito.</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> ➤ Si el usuario no se ha suscrito a ningún boletín no podrá realizar la acción de darse baja. ➤ Para realizar la acción de darse baja al menos debe haber recibido un boletín. 	
<p>Prototipo de interfaz: No aplica.</p>	

Tabla 15: H.U. Mostrar ayuda del módulo.

Historia de usuario	
Número: 10	Nombre del requisito: Mostrar ayuda del módulo.
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración
Prioridad: Baja	Tiempo Estimado: 0.125 días
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días

Descripción: Los usuarios podrán consultar la ayuda del módulo ubicada a la derecha de dicho módulo.

Observaciones:

- El usuario puede consultar la ayuda general o la ayuda del administrador por separados.

Prototipo de interfaz:

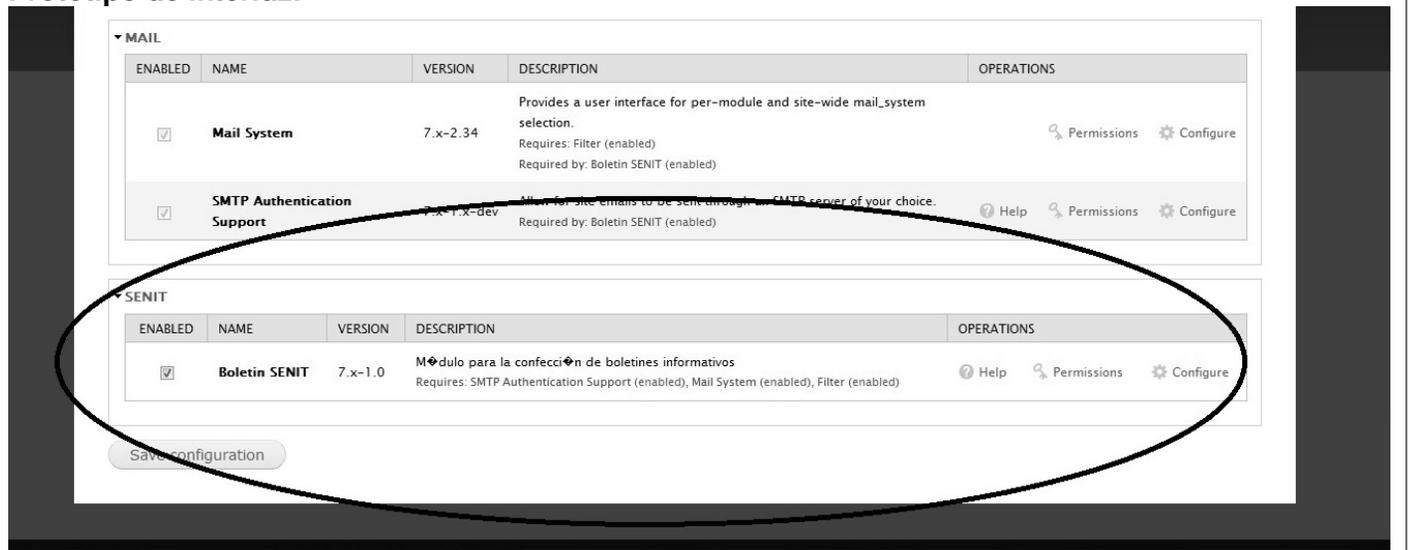


Tabla 16: H.U. Mostrar la cantidad de usuarios que acceden desde un boletín.

Historia de usuario	
Número: 11	Nombre del requisito: Mostrar la cantidad de usuarios que acceden desde un boletín.
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración
Prioridad: Alta	Tiempo Estimado: 0.13 días
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días
<p>Descripción: Se podrá observar mediante esta funcionalidad una tabla con la cantidad de usuarios que acceden desde un mismo boletín.</p>	

Observaciones:

Prototipo de interfaz:

The screenshot shows a web application interface for 'Boletín SENIT'. The navigation menu includes: Dashboard, Content, Structure, Appearance, People, Modules, Configuration, Reports, Help. The user is logged in as 'admin'. The page title is 'Resumen de suscriptores'. Below the title is a table with the following data:

USUARIO	CORREO	SUSCRITO	CATEGORIA
Anónimo	operez@uci.cu	Si	Cultura Deporte
admin	rconde@estudiantes.uci.cu	Si	Boletín SENIT Ciencia Cultura Deporte Sociedad

Tabla 17: H.U. Mostrar los boletines más utilizados.

Historia de usuario	
Número: 12	Nombre del requisito: Mostrar los boletines más utilizados.
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración
Prioridad: Alta	Tiempo Estimado: 0.13 días
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días
Descripción: Muestra una tabla con los boletines más utilizados..	
Observaciones:	
Prototipo de interfaz:	

TITULO	CATEGORIA	CANTIDAD DE USUARIOS
El deporte	Deporte	2
Lines and paragraphs break automatically.	Sociedad	1
dasdasd	Ciencia	1
Prueba	Boletin SENIT	1
ciencia	Ciencia	1
fgfdgd	Cultura	2

Tabla 18: H.U. Listar los contenidos más accedidos desde un boletín.

Historia de usuario													
Número: 13	Nombre del requisito: Listar los contenidos más accedidos desde un boletín..												
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración												
Prioridad: Alta	Tiempo Estimado: 0.13 días												
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días												
Descripción: Se podrá observar los contenidos más accedidos desde un mismo boletín.													
Observaciones:													
Prototipo de interfaz:													
<p>The screenshot shows a web application interface with a navigation menu at the top (Boletin SENIT, Dashboard, Content, Structure, Appearance, People, Modules, Configuration, Reports, Help) and a user profile (Hello admin, Log). Below the navigation is a breadcrumb trail: Home » Administration » Boletin SENIT. The main heading is 'Contenido mas accedido desde un boletín.' followed by a table with three columns: TITULO, FECHA, and CANTIDAD. The table contains three rows of data.</p> <table border="1"> <thead> <tr> <th>TITULO</th> <th>FECHA</th> <th>CANTIDAD</th> </tr> </thead> <tbody> <tr> <td>Articulo de prueba título</td> <td>2016-May-24 09:00:53</td> <td>5</td> </tr> <tr> <td>Página de prueba título</td> <td>2016-May-24 09:01:20</td> <td>5</td> </tr> <tr> <td>sdasdasdas</td> <td>2016-May-30 12:44:41</td> <td>1</td> </tr> </tbody> </table>		TITULO	FECHA	CANTIDAD	Articulo de prueba título	2016-May-24 09:00:53	5	Página de prueba título	2016-May-24 09:01:20	5	sdasdasdas	2016-May-30 12:44:41	1
TITULO	FECHA	CANTIDAD											
Articulo de prueba título	2016-May-24 09:00:53	5											
Página de prueba título	2016-May-24 09:01:20	5											
sdasdasdas	2016-May-30 12:44:41	1											

Tabla 19: H.U. Listar los tipos de contenidos más accedidos desde un boletín.

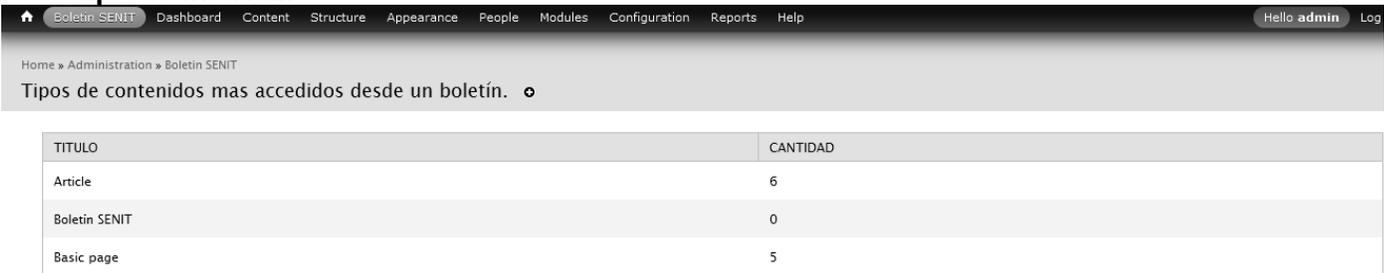
Historia de usuario									
Número: 14	Nombre del requisito: Listar los tipos de contenidos más accedidos desde un boletín.								
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración								
Prioridad: Alta	Tiempo Estimado: 0.13 días								
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días								
Descripción: Se podrá observar los tipos contenidos más accedidos desde un mismo boletín.									
Observaciones:									
Prototipo de interfaz:									
 <p>The screenshot shows a web application interface. At the top, there is a navigation menu with items like 'Boletín SENIT', 'Dashboard', 'Content', 'Structure', 'Appearance', 'People', 'Modules', 'Configuration', 'Reports', and 'Help'. On the right, it says 'Hello admin' and 'Log'. Below the menu, there is a breadcrumb trail: 'Home » Administration » Boletín SENIT'. The main heading is 'Tipos de contenidos mas accedidos desde un boletín.' followed by a dropdown arrow. Below this is a table with the following data:</p> <table border="1"> <thead> <tr> <th>TITULO</th> <th>CANTIDAD</th> </tr> </thead> <tbody> <tr> <td>Article</td> <td>6</td> </tr> <tr> <td>Boletín SENIT</td> <td>0</td> </tr> <tr> <td>Basic page</td> <td>5</td> </tr> </tbody> </table>		TITULO	CANTIDAD	Article	6	Boletín SENIT	0	Basic page	5
TITULO	CANTIDAD								
Article	6								
Boletín SENIT	0								
Basic page	5								

Tabla 20: H.U. Mostrar resumen de suscriptores.

Historia de usuario	
Número: 15	Nombre del requisito: Mostrar resumen de suscriptores.
Programador: Roberto Verdecia Conde	Iteración Asignada: Primera Iteración
Prioridad: Alta	Tiempo Estimado: 0.13 días
Riesgo en Desarrollo: Gespro/Plan de riesgo	Tiempo Real: 0.125 días
Descripción: Mostrará a través de una tabla todo los usuarios suscritos al boletín.	

Observaciones:

- En uno de los campos de la tabla se podrá observar si el usuario esta suscrito aun o canceló su solicitud.

Prototipo de interfaz:

The screenshot shows a web interface for 'Boletín SENT'. The top navigation bar includes links for Dashboard, Content, Structure, Appearance, People, Modules, Configuration, Reports, and Help. A user is logged in as 'admin'. The breadcrumb trail is 'Home » Administration » Boletín SENT'. The main heading is 'Resumen de suscriptores'. Below this is a table with the following data:

USUARIO	CORREO	SUSCRITO	CATEGORIA
Anónimo	operez@uci.cu	Si	Cultura Deporte
admin	rconde@estudiantes.uci.cu	Si	Boletín SENT Ciencia Cultura Deporte Sociedad