



Universidad de las Ciencias
Informáticas
Facultad 2

PLATAFORMA PARA LA CREACIÓN DE BOLETINES INFORMATIVOS PARA EL CENTRO DE TELEMÁTICA

**Trabajo de Diploma para optar por el Título de Ingeniero en
Ciencias Informáticas**

Autor:

Bárbaro Guillermo Barroso Gómez

Tutor:

Ing. Yenlys Guerra Dávila

La Habana, junio de 2017

“Año 59 de la Revolución”



Quiero poner
un ding en el
universo

Steve Jobs

DECLARACIÓN DE AUTORÍA

Declaro ser el único autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año ____.

Bábaro Guillermo Barroso Gómez

(Autor)

Ing. Yenlys Guerra Dávila

(Tutora)

DATOS DE CONTACTO

Autor:

Bábaro Guillermo Barroso Gómez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: bgbarroso@estudiantes.uci.cu

Tutor:

Ing. Yenlys Guerra Dávila

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: yguerra@uci.cu

A mis padres por siempre estar ahí para mi desde el primer momento
de mi vida.

A mis abuelos por enseñarme que aún en estos tiempos el
conocimiento más básico es útil.

A mis tías, en especial a Albis por confiar en que este día llegaría.

A Yusnelys por traerme esa paz que tanto necesitaba.

A mi tutora por demostrarme que nada es absoluto y que el ser
absoluto solo nos impide crecer como profesionales.

A Suany por demostrarme que en la vida todo se logra. Sea de una
manera o de otra.

A Humberto por las ideas por la ayuda y por ser tan buen oyente.

Al finalizar un trabajo tan arduo y lleno de dificultades como el desarrollo de una tesis es inevitable que te asalte un muy humano egocentrismo. Sin embargo, luego de unos minutos te das cuenta de que este resultado no hubiese sido posible sin la participación de personas que han facilitado las cosas para que este trabajo llegue a un feliz término. Por ello, es para mí un verdadero placer utilizar este espacio para ser justo y consecuente con ellas, expresándoles mis agradecimientos. Algunas personas se quejan de los Padres que tienen, pero yo en cambio agradezco mucho por los que la vida me brindó. Puede que no haya crecido entre riquezas materiales, pero ellos siempre me rodearon de mucho amor y gracias a ello he crecido muy feliz. Es por ustedes que he llegado hasta aquí. Sacrificaron mucho de sus vidas para que yo pudiera construir la mía y no saben lo eternamente agradecido que me encuentro. Les agradezco los consejos que siempre me han brindado, los abrazos que me han dado cuando más los necesitaba, los cuidados que tienen conmigo cuando me enfermaba, les agradezco todo eso y mucho más. Durante el desarrollo de esta tesis se presentaron varias situaciones las cuales pudieron llegar a ser en muchos momentos, motivos convincentes para rendirse o tirar la toalla, hubo momentos en los que la esperada y anhelada ayuda no parecía llegar hubo momentos en que el estrés casi me domina. Pero en esos mismos momentos resaltó dentro de mi vida esa persona que durante muchos días y meses fue mi compañera, mi amiga, mi consejera. Yusne gracias por estar ahí gracias por soportarme (que sé de muy buena fuente que no es tarea fácil) y te agradezco de la manera más sincera, e infinitamente por tu ayuda, y por tus incontables apoyos en mi vida. A mis amigos del día a día, a esos que puedo llamar familia por tenderme la mano y estar ahí Viset, Yaneisi (gracias por alimentarme) y por escucharme cuando lo necesitaba. A Judith, una de las primeras personas que conocí en esta universidad y me ayudo con las asignaturas. A aurora y Bouza, los padres de Judith por ayudarme cada vez que lo necesitaba incluso hasta el día de hoy. Felipe, que decir de Felipe... mi hermano de otra mamá... un grillete que arrastro desde el círculo infantil, como me dijiste en tu tesis "más de dos décadas soportándote". Pero es bueno saber que siempre estás ahí y que aun en los momentos difíciles puedo contar contigo. Gracias por nunca decirle no a ninguna de mis ideas locas y disculpa por arrastrarte a ellas y gracias por siempre ponerle ese toque ilustrado a mis obras más ocurrentes. Emilio gracias por ser mi consultor de ideas innovadoras y gracias por tenderme una mano siempre que pudiste y además gracias por demostrarme lo malo que era Wisbel en juegos de estrategia. Y Wisbel a ti gracias por

enseñarme que siempre hay que tener ganas de hacer cosas nuevas. Siempre en un aula hay conjuntos de personas que te llegan, siempre hay uno que te cae mal, siempre está el que te quiere bien, el que te quiere mal. Yo me llevo de mis compañeros la experiencia de haber compartido con ellos estos años y a todos les digo que este donde este, siempre tendrán una puerta abierta y una mano amiga para ayudarlos. A todas gracias por soportarme. Especial mención a los muchachos del Madrid (Yeniel, Ronald, Alberto, Fernando y Humberto que ya tubo merecida mención en la dedicatoria de esta tesis y que hoy debería estar aquí defendiéndola, pero por cosas de la vida no se pudo), a las niñas del aula Daynelita un beso mi amor, gracias por enseñarme la parte incoherente de esta vida, Sandra, Yeni, Yilian, Yanira, Yuliet, Yunexy. A la gente del apartamento Yosiel, Julien, Yoan, Luis Angel, Duriel. A la gente de la radio por dejarme aprender de ellos a Sandy por enseñarme a comunicar de manera breve y concisa las ideas, a papo por hacernos reír a todos y darnos lecciones de vida y de puntualidad, a Jose por darnos la confianza y por estar ahí para ayudar, A Rosaibis por dejarse arrastrar y por criticar todo lo posible. Ahora la parte difícil las personas que me ayudaron crecer durante estos seis años en la UCI y que me enseñaron, me educaron y me mostraron que aun de los momentos más oscuros y las malas experiencias se pueden sacar cosas positivas. A la profe Rosa por ser mi mamá desde el día que puse un pie en la UCI. Al profe Hugo que no está aquí pero que me ayudó y me aconsejó siempre que lo necesité. A la profe Madelin Haro por mostrarme lo que puede llegar a hacer un ingeniero. A la profe Lianne y al profe Yubismel por enseñarme a enseñar y a transmitir mejor mis conocimientos. A la profe Yenlis por alarme las orejas y nunca pero nunca estar de acuerdo con mis ideas gracias a eso siempre pude mejorar gracias por ayudarme a crecer y demostrarme que el absolutismo no conduce a nada, sinceramente no imagino a alguien mejor como mi tutor en este momento. Profe Suany a Ud. gracias por darme ese espacio para demostrar lo que podía llegar a hacer y gracias por siempre escuchar mis propuestas e ideas más tontas y darles forma y gracias por siempre hacerme sentir parte del proceso. A Luis Manuel por ser mi Wikipedia personal de códigos, tendencias y herramientas web. A los profes Yoanny, Antonio y Diannet por ayudarme a mejorar con cada crítica y recomendación acertada durante los cortes. A Saylin mi profe guía. A Yoiler por enseñarme a ser justo e imparcial y demostrarme que siempre hay un peldaño de conocimiento aun por escalar. A la profe Idalis por enseñarme a nunca dejar de investigar. A la profe Arianna por el público más exigente al que me enfrente en cada una

de mis obras. A todos los que conocí pero que a estas horas no atino a recordar no se sientan desplazados es que tengo mala memoria. Pero les aseguro que de una manera u otra ya a cada uno de Uds. les he dado mis más sinceros agradecimientos. No por ello deja de ser válido que justo antes de terminar cerremos con un simple...

A todos y cada uno de Uds.... Gracias.

RESUMEN

En la denominada sociedad de la información es común el intercambio de datos entre las personas a través de un vínculo con la tecnología. Uno de los métodos usado para ello, lo constituyen los boletines informativos en formato PDF, los cuales han alcanzado un gran auge hoy día gracias a las posibilidades que acarrea para cualquier negocio su distribución a través de una lista de correo. Dichos boletines son producto de un proceso extenso en el cual varias personas deben converger para su realización, lo cual atrasa de manera considerable su llegada al lector, haciéndolo ineficiente.

En el presente trabajo se desarrolla una herramienta que permite informatizar el proceso generación y distribución de un boletín informativo. En la propuesta se utiliza como metodología de desarrollo de *software* XP, como tecnologías del lado del servidor el lenguaje de programación Python en conjunto con el Framework Django y el plug-in Django Rest Framework, además de la Librería ReportLab para el trabajo y dibujo del boletín en formato PDF y la librería Pillow para el trabajo con las imágenes. Además, como tecnologías del lado del cliente se utilizaron HTML, CSS, JavaScript, jQuery, siguiendo las pautas establecidas por la estrategia marcaria de la Universidad de las Ciencias Informáticas para la rama de productos XILEMA y así crear una aplicación cliente que interactúa con la API en el servidor y ofrece sus funcionalidades de una forma amigable a los usuarios.

Palabras clave: información, boletín informativo, ReportLab, Pillow, maquetador web.

Índice

Introducción.....	9
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	14
1.1 PROCESO DE CREACIÓN DE BOLETINES	14
1.2 MAQUETACIÓN DE BOLETINES. ESTADO DEL ARTE	15
1.2.1 Plataforma web BEE	16
1.2.2 Plataforma web WIX.com	16
1.2.3 Peurple DS.....	17
1.2.4 Microsoft Publisher.....	18
1.2.5 Adobe InDesign	18
1.2.6 Adobe Acrobat	19
1.3 ESTADO DEL ARTE. RESULTADOS DEL ESTUDIO.....	19
1.4 METODOLOGÍA DE DESARROLLO DE SOFTWARE	21
1.4.1 Fundamentación de la metodología de desarrollo seleccionada.....	21
1.5 TECNOLOGÍAS Y HERRAMIENTAS DE DESARROLLO	22
1.5.1 Tecnologías del lado del servidor.....	22
1.5.2 Tecnologías del lado del Cliente	26
1.5.3 Herramientas de desarrollo	27
1.6 ANÁLISIS DE LA POLÍTICA DE COMUNICACIÓN INSTITUCIONAL Y EL MANUAL DE IDENTIDAD DE LA UCI.....	27
1.7 CONCLUSIONES DEL CAPÍTULO	30
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN.....	31
2.1 ANÁLISIS DE EL PROCESO DE CREACIÓN Y MONTAJE DEL BOLETÍN DEL CENTRO DE TELEMÁTICA	31
2.2 CARACTERÍSTICAS DE LA PROPUESTA DEL SISTEMA	32
2.3 CARACTERÍSTICAS NO FUNCIONALES DEL SISTEMA	33
2.4 CONCLUSIONES DEL CAPÍTULO	35
CAPÍTULO 3: EXPLORACIÓN, PLANIFICACIÓN Y DISEÑO	36
3.1 EXPLORACIÓN.....	36
3.1.1 Historias de usuario	36
3.2 PLANIFICACIÓN.....	43
3.2.1 Estimación del esfuerzo por historias de usuario.....	43
3.2.2 Plan de iteraciones	44

3.2.3 Plan de entregas.....	45
3.3 DISEÑO.....	45
3.3.1 Patrón arquitectónico.....	46
3.3.2 Patrones de diseño.....	47
3.3.3 Tarjetas CRC.....	49
3.3 CONCLUSIONES DEL CAPÍTULO.....	51
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.....	52
4.1 ESTÁNDARES DE CODIFICACIÓN.....	52
4.2 ITERACIONES.....	53
4.2.1 ITERACIÓN 1.....	53
4.2.2 ITERACIÓN 2.....	59
4.2.3 ITERACIÓN 3.....	63
4.3 CONCLUSIONES DEL CAPÍTULO.....	66
CONCLUSIONES.....	68
RECOMENDACIONES.....	69
REFERENCIAS BIBLIOGRÁFICAS.....	70
BIBLIOGRAFÍA.....	72

Introducción

Actualmente gran parte de la humanidad está marcada por un rápido e intenso desarrollo tecnológico. Las tecnologías de la información y las comunicaciones (TIC) forman parte indispensable en la vida del ser humano. La informática, a nivel mundial, supone una nueva revolución tecnológica, pues han sido numerosos los avances alcanzados en dicha ciencia en materia de comunicación y gestión del conocimiento, debido a la necesidad de información y de nuevos saberes que tiene la sociedad actual.

La tecnología y la creación de redes interconectadas influyen fuertemente en el escenario económico, en las sociedades, individuos, grupos, corporaciones y organizaciones educativas, reflejándose la necesidad de generar nuevos comportamientos en lo que respecta a la forma tradicional de trabajar, comunicarse, convivir, relacionarse y aprender (Rivas 2012). En la denominada Sociedad de la Información es común el intercambio de datos entre las personas a través de un vínculo con la tecnología. La aparición de internet ha supuesto un cambio importante en la sociedad, facilitando la forma de acceder, organizar y comunicar información relevante con mayor eficiencia (Martinez 2014). Uno de los métodos usados para ello, lo constituyen los boletines informativos en formato PDF¹ los cuales han alcanzado un gran auge hoy día gracias a las posibilidades que acarrea para cualquier negocio su distribución a través de una lista de correo.

Entre las principales ventajas que presenta este tipo de tecnología está el hecho de que, a diferencia de otros canales como las redes sociales, el contacto con el suscriptor es directo. Es un canal que, bien usado, permite construir una relación de confianza con el suscriptor que puede ser la base de futuras ventas o relaciones de colaboración. La calidad de la información que se les facilita a las personas suscritas es superior a la encontrada mediante la navegación entre sitios esporádicos de la red a través de los buscadores.

Muchos expertos en *marketing* afirman que adquirir un nuevo cliente cuesta entre seis y siete veces más que conservar un cliente existente. Mantener el contacto con

¹ PDF: Sigla del inglés Portable Document Format (formato de documento portátil)

los clientes a través de un boletín informativo brinda la posibilidad de que puedan pensar en su empresa en primer lugar cuando tengan que hacer otra compra o hacer una recomendación (Microsoft 2015).

En Cuba se lleva a cabo un proceso de informatización que evidencia el uso de soluciones informáticas con el objetivo de automatizar los procesos de negocio en las instituciones. La Universidad de las Ciencias Informáticas (UCI), tiene la misión de producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación y servir de soporte a la industria cubana de la informática (UCI 2002). Dentro de su infraestructura docente-productiva se encuentra el Centro de Telemática (TLM), el cual tiene como misión el desarrollo de sistemas y servicios informáticos en la rama de las telecomunicaciones y la seguridad informática (TLM 2010).

Este centro cuenta con un grupo de especialistas y estudiantes los cuales pertenecen a los proyectos asociados a su línea de desarrollo; los mismos poseen la necesidad de mantenerse informados, de manera periódica, acerca de los estados de los proyectos, además de mostrar las líneas y avances investigativos del mismo, así como otras informaciones de interés para el área. Así mismo no hay forma de brindar a los clientes, tanto regulares como potenciales, información sobre los productos y servicios del centro. Para suplir estas deficiencias se define el proceso de conformación de un boletín de tipo electrónico el cual permitirá al Centro contar con un recurso de información y *marketing*.

El boletín debe ser conformado por un grupo de personas con conocimiento de diseño los cuales deben crear los recursos y estructura del mismo para cada edición, lo cual toma bastante tiempo y atrasa de manera considerable la publicación de cada edición ya que cada diseño e información a publicar debe ser revisada y aprobada, esto implica que debe generarse el boletín un considerable número de veces, además repercute en que la información no llegue de la manera más inmediata a los suscriptores disminuyendo su eficiencia.

Teniendo en cuenta los siguientes elementos:

- La importancia del uso del boletín como recurso de información y *marketing* en una institución.

- El amplio uso de los boletines en las instituciones cubanas.
- La necesidad de investigar las herramientas existentes para gestionar el diseño y composición de un boletín.
- La importancia de informatizar el mecanismo de recopilación y montaje de la información en un boletín haciendo uso de las tecnologías *web*.

Se plantea como **problema a resolver**: ¿Cómo gestionar el proceso de generación del boletín para el centro de Telemática de la Facultad dos en la Universidad de las Ciencias Informáticas?

Se define como **objeto de estudio**: el proceso de generación de un boletín. Estableciéndose como **objetivo general** de la presente investigación desarrollar una aplicación que permita gestionar el proceso de generación del boletín para el centro de Telemática de la Facultad dos de la Universidad de las Ciencias Informáticas, definiéndose como **campo de acción**: el proceso de generación del boletín para el centro de Telemática de la facultad número dos de la Universidad de las Ciencias Informáticas.

Para dar cumplimiento al objetivo planteado, se definen las siguientes **tareas de investigación**:

- Análisis de las herramientas y tecnologías existentes que permiten el diseño y composición de un boletín para establecer similitudes con la investigación en curso.
- Análisis de los procesos que intervienen en el diseño y composición de un boletín, para tener un mejor dominio del problema a resolver.
- Estudio de los elementos correspondientes a la metodología de desarrollo de *software* seleccionada con el objetivo de llevar a cabo un desarrollo organizado.
- Caracterización de las herramientas y tecnologías seleccionadas que permitan una correcta implementación del sistema.
- Análisis de los diferentes tipos de pruebas propuestos por la metodología escogida para validar el correcto funcionamiento del sistema desarrollado.

Para facilitar el cumplimiento del objetivo propuesto y de las tareas de investigación se emplean métodos teóricos y empíricos de la investigación científica.

Métodos teóricos:

- **Analítico-Sintético:** Permite realizar el estudio teórico de la investigación, facilitando el análisis de documentos y la extracción de los elementos más importantes acerca del funcionamiento de las aplicaciones que permiten el diseño y montaje de boletines informativos en la actualidad.
- **Histórico-Lógico:** Este método permite estudiar todo lo relacionado con los sistemas de diseño y montaje de boletines, para así obtener un conocimiento histórico de su desarrollo y comportamiento a nivel internacional como nacional.

Métodos empíricos:

- **Medición:** El sistema obtenido como resultado de la investigación será medido a través las pruebas unitarias y de aceptación garantizando el correcto funcionamiento del mismo.

En aras de estructurar el proceso de desarrollo del presente trabajo de una manera coherente y organizada, se dividió el mismo en un total de cuatro capítulos:

Capítulo 1. Fundamentación Teórica: Se abordan los elementos teóricos más importantes que sirven de base para la realización de la investigación; también se proporciona una descripción de las herramientas y metodología de desarrollo de *software* a utilizar para dar solución al problema planteado.

Capítulo 2. Propuesta de solución: Se describe la propuesta de solución creada, además del flujo actual de los procesos a automatizar. Se detallan las funcionalidades y características que tendrá la propuesta del sistema, lo que posibilitará un mayor entendimiento del mismo.

Capítulo 3. Exploración, Planificación y Diseño: Se detallan las historias de usuario definidas, que recogen todas las funcionalidades a implementar. Se muestran además otros artefactos propios de la metodología de desarrollo, que se generan en estas etapas.

Capítulo 4. Implementación y Pruebas: Se expone todo lo relacionado a los procesos de implementación y pruebas del sistema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se realiza un análisis acerca de las herramientas y procesos para la conformación de boletines informativos tanto a nivel internacional como nacional. Se hará una descripción los conceptos fundamentales asociados al dominio del problema y se hace un análisis entre las soluciones existentes y la propuesta que se realiza. Se analizará el proceso de creación de boletines y sus principales etapas. Se hace un estudio de los maquetadores *web* de boletines describiendo sus elementos fundamentales y características. También se describen las principales características de las tecnologías, metodologías, lenguajes y herramientas utilizadas para el desarrollo de la aplicación, utilizados para dar solución al problema planteado.

1.1 PROCESO DE CREACIÓN DE BOLETINES

Un boletín informativo es un informe, impreso o digital, que contiene información e ideas, el cual es distribuido en forma regular a un grupo de personas interesadas. Los boletines son, típicamente, de dos a ocho páginas de extensión, y varían considerablemente en costo, calidad y contenido.(Karen Salas 2017)

En el ámbito empresarial, la función principal del Boletín informativo o *Newsletter* es la de ofrecer a sus clientes información interesante sobre lo que su empresa conoce, su negocio. Es decir, novedades sobre el sector donde opera, información sobre sus nuevos productos o servicios y noticias relacionadas.(Crexentia 2015)

Teniendo en cuenta que el público al que van dirigidos los boletines es muy variable y la mayoría de las veces sectorial se entiende que los mismos son diversos y el proceso para su creación tiende a variar.(Lledó 2016) Hay muchas formas de producir un buen boletín, pero existen un conjunto de reglas o pautas esenciales para su conformación.

Se deberá contestar una serie de preguntas respecto al boletín, que ayudarán a conformar el manifiesto por el cual se elaborarán todas las ediciones.

- ¿Cuál es el propósito primario del boletín?
- ¿Quién es la audiencia?
- ¿Con cuánta frecuencia se quiere que aparezca el boletín?

- ¿Cuál será la frecuencia de los plazos de entrega?
- ¿Quién diseñará y editará el boletín?
- ¿Quién escribirá los artículos?
- ¿Cómo se le distribuirá el boletín a la audiencia?(Karen Salas 2017)

Una vez contestadas estas preguntas se procede a la elaboración del manifiesto y la conformación de los grupos editoriales que integrarán el equipo de realización.

Los grupos editoriales son:

- Informativo: Se encarga de generar todo el contenido del boletín, dígame artículos e informaciones.
- Diseño: Se encarga de generar todo el contenido gráfico y organizativo del boletín. Además de ser los responsables de conformar el boletín.
- Revisión: Se encarga de revisar que todo contenido y material gráfico generado en el boletín para cada edición este en correspondencia con lo establecido en el manifiesto y cumpla con los requisitos de calidad requeridos.
- Publicación: Se encarga de distribuir o publicar el boletín ya sea de manera impresa o digital.(Crexentia 2015)

Los pasos generales o el flujo de trabajo que siguen estos grupos son los siguientes:

- Decidir qué aparecerá en la publicación y los elementos que se incluirán cada uno de estos contenidos.
- Diseñar el formato del boletín.
- Redactar los artículos para el boletín.
- Editar los artículos revisando el contenido, el estilo y los espacios.
- Generar la copia final.
- Revisar la copia final por última vez para ver si hay algún error de edición.
- Enviar por correo o distribuir el boletín.(Lledó 2016)

1.2 MAQUETACIÓN DE BOLETINES. ESTADO DEL ARTE

Diversas son las herramientas que hoy están a disposición de los usuarios para la elaboración de boletines electrónicos, tanto para la *web* como para el entorno de escritorio. Su uso y explotación por parte de las nuevas empresas es vital para el

desarrollo del *marketing* electrónico. A continuación, se realizará un análisis acerca de las principales aplicaciones para la maquetación de boletines informativos en formato PDF, sus similitudes y diferencias en la forma en la que gestionan la información, respecto a la solución que se desea dar en el presente trabajo.

1.2.1 Plataforma *web* BEE

BEE es un editor *web*, perteneciente a la empresa MailUP², que utiliza la filosofía *Drag and Drop*³ para maquetar boletines informativos electrónicos para correos electrónicos de diseño adaptable. Cuenta con una interfaz que permite a cualquier persona crear un hermoso mensaje de correo electrónico. Una vez que el mensaje se ha creado, puede generar una vista previa, vista de prueba o descargarlo en formato HTML⁴.(SPA 2016)

Esta herramienta privativa no cuenta con un sistema de recopilación de información, además no permite la importación de imágenes propias. La información y datos de negocio introducidos en la plataforma, son propensos a ser usados por los administradores para operaciones de *marketing* de la propia empresa. El costo de uso de la herramienta es a través de una cuota mensual que se pacta con los administradores una vez adquirido el servicio, y se calcula según el número de prestaciones que desee utilizar el usuario. El costo por el paquete completo de aplicaciones son 250 USD al mes incluyendo servicios de *hosting*. (SPA 2016)

1.2.2 Plataforma *web* WIX.com

Wix.com es una plataforma de desarrollo *web* basada en la nube que permite a los usuarios crear sitios *web* HTML5 y sitios móviles a través del uso de su filosofía de *Drag and Drop*. Los usuarios pueden agregar funcionalidades como *plug-ins*⁵ sociales, *e-commerce*⁶, formularios de contacto, *marketing* a través de boletines

² MailUP: Empresa italiana líder en el campo del marketing electrónico.

³ Drag and Drop: En las interfaces gráficas de usuario, arrastrar y soltar es un gesto en el que el usuario selecciona un objeto virtual "agarrándolo" y arrastrándolo a una ubicación diferente o a otro objeto virtual.

⁴ HTML: Es el lenguaje de marcado estándar para crear páginas *web*.

⁵ Plug-in: En la informática, un plug-in es un componente de software que agrega una característica específica a un programa de computadora existente.

⁶ E-commerce: El comercio electrónico (comercio electrónico o EC) es la compra y venta de bienes y servicios, o la transmisión de fondos o datos, a través de una red electrónica, principalmente Internet.

informativos por correo electrónico a sus sitios *web*, usando una variedad de aplicaciones desarrolladas por Wix y de terceros. Los usuarios deben comprar paquetes *premium*⁷ para conectar sus sitios a sus propios dominios, eliminar los anuncios Wix, agregar capacidades de comercio electrónico, comprar almacenamiento de datos adicional y ancho de banda, entre otros. (PCWorld 2012)

Esta herramienta privativa no cuenta con un sistema de recopilación de información, desde Cuba no está permitido el registro y su uso está restringido a usuarios registrados. La información y datos de negocio introducidos a la plataforma, son propensos a ser usados por los administradores para operaciones de *marketing* de la propia empresa. El costo de licencia de la herramienta es a través del uso de las plantillas para generar los boletines ya que el sistema solo permite la aplicación de las plantillas que el mismo ofrece y estas cuestan entre 9.89 y 59.90 USD cada una según el nivel de complejidad que requiera el usuario para su publicación. (PCWorld 2012)

1.2.3 Peurple DS

Peurple DS Publishing Suite es una solución privativa y de costo, perteneciente a la empresa SpryLab, eficiente para la edición digital en una aplicación *desktop* o en la *web*. Todas sus herramientas están totalmente interconectadas y proporcionan una experiencia de usuario única al poder compartir información y estilos. Además, todos los sistemas se pueden conectar fácilmente. Está especialmente diseñado para optimizar los flujos de trabajo y garantizar un proceso de publicación electrónica eficiente y rentable. La *suite* le permite utilizar una variedad de procesos de producción completamente automáticos, y permite combinar diferentes tipos de flujos de trabajo, para ofrecer los mejores resultados a cada tipo de proyecto. (Sprylab 2016)

Esta herramienta privativa es una *suite* integrada que permite el desarrollo colaborativo de proyectos de diseño. Cuenta con un poderoso sistema de manejo y organización de la información, pero no cuenta con un sistema de plantillas por lo que requiere que estas sean hechas a mano y que sean elaboradas con productos

⁷ Premiun: Es un bono, un regalo o una suma adicional al precio, salarios, intereses o similares.

de la Suite de Adobe ya que no permite la integración con otros editores de vectores e imágenes que no pertenezcan a esta *suite*. Su sistema de negocio se basa en una plataforma de uso completo la cual no permite el desglose de paquetes de servicios, por lo que debe ser adquirida de manera íntegra. Su costo total es de 650 USD al mes. (Sprylab 2016)

1.2.4 Microsoft Publisher

Microsoft Publisher (anteriormente Microsoft Office Publisher) es la aplicación de autoedición de Microsoft. A menudo es considerado como un programa para principiantes, que difiere del procesador de textos Microsoft Word en que se hace hincapié en el diseño y la maquetación de las páginas, más que en el proceso y corrección de textos. Provee un historial simple de edición similar al de su producto hermano Microsoft Word. Ayuda a crear, personalizar y compartir con facilidad una amplia variedad de publicaciones y material de *marketing*. Incluye una variedad de plantillas, instaladas y descargables desde su sitio *web*, para facilitar el proceso de diseño y edición. (Microsoft 2015)

Su principal desventaja es que no cuenta con un sistema de recopilación de información. Cuenta con un sistema de plantillas gratuitas, pero estándares, los paquetes personalizados y con más funcionalidades poseen un costo que oscila entre los 25 y 40 USD. Su licencia de uso posee un costo de 1.45 USD al mes. (Microsoft 2015)

1.2.5 Adobe InDesign

Adobe InDesign es una aplicación de *software* de edición electrónica producida por Adobe Systems. Se puede utilizar para crear obras como carteles, folletos, revistas, periódicos y libros. Adobe InDesign también puede publicar contenido adecuado para dispositivos *tablet* junto con Adobe Digital Publishing Suite. Los diseñadores gráficos y los artistas de producción son los principales usuarios, creando y distribuyendo publicaciones periódicas, carteles, medios impresos y boletines informativos. (Adobe Suite 2016)

Esta herramienta no cuenta con un sistema de recopilación y organización de la información propios, pero puede adquirir esta capacidad haciendo uso del *software* Adobe InCopy. Las plantillas, estilos, fuentes, paletas de colores, y diseños

vectoriales son de uso gratuito, con atribución al autor, para usuarios registrados en los servicios de Adobe Creative Suite o Adobe Creative Cloud, el uso de estos servicios no están disponibles desde Cuba por reglas administrativas de Adobe Systems. El costo de uso de este *software* asciende a los 49.90 USD al año. (Adobe Suite 2016)

1.2.6 Adobe Acrobat

Adobe Acrobat es una familia de *software* de aplicación y servicios *web* desarrollada por Adobe Systems para ver, crear, manipular, imprimir y administrar archivos en formato de documento portátil (PDF). La familia está formada por Acrobat Reader (anteriormente Adobe Reader), Acrobat (anteriormente Acrobat Exchange) y Acrobat.com. El Acrobat Reader básico, disponible para varias plataformas de escritorio y móviles; soporta ver, imprimir y anotar archivos PDF. El Acrobat comercial propietario, disponible sólo para Microsoft Windows y OS X, también puede crear, editar, convertir, firmar digitalmente, cifrar, exportar y publicar archivos PDF. (Adobe 2016)

1.3 ESTADO DEL ARTE. RESULTADOS DEL ESTUDIO

El análisis detallado del estado del arte asociado a los maquetadores *web* permitió identificar las principales características y deficiencias de las herramientas estudiadas. También se logró estudiar los diferentes patrones, estándares y tendencias acerca del diseño de este tipo de aplicación, así como sus principales servicios y técnicas de recopilación y manejo de la información. Los resultados de la evaluación de los sistemas en las variables definidas se detallan en la tabla 1.1.

Sistemas \ Variables	Gestión de la estructura	Gestión del contenido	Gestión de listas de correo	Exporta PDF	Sistema de Plantillas	Personalización y adición de plantillas	Uso de recursos propios	Licencia propietaria	Costo (USD)
Bee	x		X		x			x	250
WIX.com	x		X		x			x	59.90
Peurple DS	x	x		x			x	x	650

Microsoft Publisher	x			x	x		x	x	1.45
Adobe InDesing	x			x			x	x	49.90
Adobe Acrobat	x			x			x		0

Tabla 1.1: Tabla de comparación de sistemas.

Deficiencias:

- Los principales sistemas que permiten el diseño y composición de un boletín son de pago y propietarios, así como sus recursos y herramientas.
- Las principales herramientas y funcionalidades que permiten el diseño y composición de un boletín son comunes, pero difieren en cuanto a características de manejo.
- Cinco de los sistemas estudiados no brindan opciones que permitan gestionar de forma organizada la información proveniente de diferentes autores.
- Dos de estos sistemas usan información del proyecto, como por ejemplo las listas de clientes para realizar campañas de *marketing* de sus propios productos.

También de este estudio se adquirieron buenas prácticas, técnicas y estilos de desarrollo que son posibles implementar en la solución para así agregarle valor de usabilidad y hacerla más competitiva en el mercado. Las mismas son:

- Uso de la filosofía *Drag and Drop* para la gestión y posicionamiento de los componentes del boletín.
- Uso de paneles de información y herramientas laterales.
- Gestión de la información de manera dinámica, centralizándola y clasificándola para su uso.
- Sistema de grupos de usuarios asociados a un proyecto, donde solo una persona posee el rol de administrador del mismo.
- Plantillas personalizadas predefinidas para la gestión rápida de contenido dentro del boletín.
- Uso y posibilidad de creación de estilos tanto para texto como páginas.

- Salva del proyecto online y publicación en la nube.
- Importación de estilos a través de proyectos asociados a otras herramientas de diseño.
- Gestión del color y las formas a través de herramientas simples como selectores.
- Distribución vía correo electrónico del boletín a través de listas de correo.

1.4 METODOLOGÍA DE DESARROLLO DE SOFTWARE

Una metodología de desarrollo de *software* se refiere a un marco de trabajo que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información (Pressman 2005). A lo largo del tiempo, una gran cantidad de métodos han sido desarrollados diferenciándose por su fortaleza y debilidad. Actualmente existen diferentes metodologías de desarrollo de *software*, debido a que todos los proyectos de *software* no son iguales y por tanto necesitan un enfoque de desarrollo acorde a sus características particulares. Dos de los principales criterios que se usan para diferenciarlas son el tamaño del personal y la criticidad del sistema. Sobre la base de estas opciones, el equipo del proyecto selecciona la metodología ligera o pesada que desea para su proyecto. (Cockburn 2000)

Los proyectos difieren en cuanto a composición y prioridades. Las personas de un proyecto poseen además diferentes niveles de experiencia, principios y habilidades. (Cockburn 2000) Todos estos aspectos deben ser considerados para seleccionar correctamente una metodología de desarrollo de *software*.

1.4.1 Fundamentación de la metodología de desarrollo seleccionada

XP⁸ fue la metodología seleccionada debido a que se adapta en gran medida al tipo de proyecto a desarrollar, las condiciones de trabajo y las prácticas utilizadas. Se enuncian varias de las razones que motivaron la selección de esta metodología.

- El proyecto posee poca envergadura. XP está concebida para ser utilizada dentro de proyectos pequeños.

⁸ XP: eXtreme Programming.

- El cliente forma parte del equipo de desarrollo. Mediante la aplicación de XP se puede lograr una retroalimentación mayor y lograr un producto acorde a las necesidades definidas.
- El riesgo de desarrollo es elevado debido al corto tiempo de entrega planteado y a los continuos cambios de requerimientos. XP está diseñada para mitigar los riesgos en proyectos con estas características.
- Las funcionalidades del sistema pueden variar. El sistema debe cambiar y ampliar sus funcionalidades de forma que sea capaz de ajustarse a cada nuevo requerimiento. Uno de los principios básicos de XP es que el cambio frecuente de las funcionalidades es algo normal en el proceso de desarrollo.
- Algunas prácticas de XP como las entregas pequeñas, el diseño simple y el Desarrollo Guiado por Pruebas (TDD⁹) son parte de la filosofía de desarrollo del equipo.(Cockburn 2000)

1.5 TECNOLOGÍAS Y HERRAMIENTAS DE DESARROLLO

El desarrollo y crecimiento de la *web* ha sido vertiginoso y sin precedentes durante los últimos años, en cuanto a número de usuarios conectados, cantidad de sitios o portales *web* y tipo de herramientas para desarrollarlas. Existen numerosos lenguajes y tecnologías relacionadas con la programación de aplicaciones que permiten generar páginas *web*, no sólo del lado del servidor sino también del lado del cliente.(Guerrero 2013)

Esta enorme cantidad de recursos, y esta característica dual de las aplicaciones *web*, brinda un amplio abanico de posibilidades a la hora de concebir el diseño de este tipo de aplicaciones. A continuación, se realiza una breve descripción de las tecnologías y herramientas seleccionadas (tanto del lado del servidor como del cliente) para ser usadas en la fase de implementación de la presente investigación.

1.5.1 Tecnologías del lado del servidor

Python 2.7.11

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90. Es un lenguaje con una sintaxis muy limpia lo que favorece un código

⁹ TDD: del inglés Test Driven Development.

legible. Se trata además de un lenguaje interpretado o de script, con tipado dinámico, multiplataforma y orientado a objetos.

Es gratis, portable, potente, y relativamente fácil de usar. Python es empleado tanto en proyectos grandes como pequeños, mejorando la productividad de los desarrolladores y la calidad del *software*.

Python ha sido seleccionado como lenguaje de programación por las características enunciadas anteriormente, así como por la gran cantidad de bibliotecas disponibles que pueden ser útiles en el presente trabajo.(Python 2017)

Django 1.10

Django es un *framework* para desarrollo *web* escrito en Python, que permite construir y mantener aplicaciones *web* de alta calidad con el mínimo esfuerzo posible. Django permite a los desarrolladores concentrarse en el código de la aplicación propiamente dicha, manejando tareas repetitivas y complejas propias de la programación *web*. De esta forma, provee un alto nivel de abstracción para patrones comunes en el desarrollo *web*, incrementando la calidad de las soluciones, aumentando la productividad y disminuyendo los errores en el código.

Django provee una arquitectura similar a la muy conocida Modelo-Vista-Controlador (MVC), con los mismos beneficios, pero con un marcado énfasis en la productividad, llamada Modelo-Plantilla-Vista (MTV). Consecuentemente, provee un conjunto de herramientas asociadas a la arquitectura, como un sistema de Mapeo Relacional de Objetos (ORM), un motor de plantillas, entre otras utilidades.(Django 2017)

Django Rest Framework 3.1

REST (*Representational State Transfer*), es un tipo de arquitectura de desarrollo *web* que se apoya totalmente en el estándar HTTP¹⁰. REST permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que

¹⁰ HTTP: El protocolo de transferencia de hipertexto es un protocolo genérico, que puede utilizarse para muchas tareas más allá de su uso para el hipertexto, como servidores de sistemas de gestión de objetos distribuidos. (Leach et al. 1999)

entienda HTTP, por lo que es mucho más simple y convencional que otras alternativas que se han usado en los últimos diez años como SOAP ¹¹y XML-RPC¹². Debido a las características mencionadas anteriormente, se decide emplear una herramienta que facilite el desarrollo de una API ¹³REST ¹⁴que permita exponer los datos almacenados en el servidor. Se escoge Django Rest Framework como solución, debido a diversas razones que serán expuestas a continuación.

Django Rest Framework es una aplicación Django que permite construir proyectos *software* bajo la arquitectura REST, incluye gran cantidad de código para reutilizar (Views, Resources, Serializers, etc.), una API navegable y una interfaz administrativa desde la cual es posible realizar pruebas sobre las operaciones HTTP, como, por ejemplo: POST y GET. Cabe resaltar además que los mentores de este proyecto hacen un uso intensivo de las vistas genéricas, con el objetivo de aprovechar las ventajas de la programación orientada a objetos.(DjangoREST 2017)

ReportLab 3.4.0

Es una librería de código abierto que permite crear de manera simple y rápida documentos en formato PDF usando el lenguaje de programación Python. Permite la elaboración de diagramas y gráficos de datos con un alto nivel de complejidad en varios formatos de mapa de bits o vectores para luego insertarlos en el documento. ReportLab crea documentos PDF basándose en las diferentes combinaciones de comandos gráficos seleccionados por el usuario. Permite una amplia variedad de resultados, además de formas de entrada de los datos a plasmar en el documento y su posterior transformación acorde a las necesidades del usuario. Posee un motor

¹¹ SOAP: *Simple Object Access Protocol*, es un protocolo ligero para el intercambio de información en un entorno distribuido y descentralizado. (Don Box et al. 2000)

¹² XML-RPC: Es un protocolo de llamada a procedimiento remoto que funciona a través de Internet. (UserLand 2012)

¹³ API: Es un conjunto de definiciones de subrutina, protocolos y herramientas para crear software de aplicación. En términos generales, es un conjunto de métodos de comunicación claramente definidos entre varios componentes de software. («IBM - Connectivity and Integration» 2016)

¹⁴ REST: Es una forma de proporcionar interoperabilidad entre sistemas informáticos en Internet. Los servicios *web* compatibles con REST permiten a los sistemas solicitantes acceder y manipular representaciones textuales de recursos *web* utilizando un conjunto uniforme y predefinido de operaciones sin estado. («IBM - Connectivity and Integration» 2016)

embebido que le permite dibujar por capas de salida, creando así un resultado más complejo y profesional.(ReportLab 2017)

PostgreSQL 9.5.4

PostgreSQL es uno de los Sistemas de Gestión de Bases de Datos (SGBD) avanzados a nivel mundial. Sus características técnicas lo hacen uno de los más potentes y robustos del mercado. Su desarrollo comenzó hace 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que se han tenido en cuenta durante su desarrollo.

Además, se integra muy bien con el lenguaje de programación seleccionado, y es una de las bases de datos mejor soportadas por el ORM¹⁵ de Django. (PostgreSQL 2017)

Apache 2.4.7

Es actualmente uno de los servidores *web* utilizado para el desarrollo de aplicaciones. Ampliamente extensible a través de módulos y configurable. Entre sus principales características destaca la configuración de servidores virtuales que le permiten ejecutar, en la misma máquina, diferentes servidores para diferentes direcciones IP, diferentes nombres de máquina o diferentes puertos.

Este trae consigo una serie de ventajas entre las que se encuentran:

- Es un servidor altamente configurable de diseño modular.
- Es una tecnología gratuita de código abierto.

Apache 2 presenta diversas características, entre ellas: un elaborado y manejable índice de directorios, un directorio de alias, negociación de contenidos, informe de errores HTTP configurable, soporte de SSL¹⁶, gestión de recursos para procesos

¹⁵ ORM: en informática es una técnica de programación para convertir datos entre sistemas de tipos incompatibles en lenguajes de programación orientados a objetos. Esto crea, de hecho, una "base de datos de objetos virtuales" que se puede utilizar desde dentro del lenguaje de programación.

¹⁶ SSL: Secure Sockets Layer es una tecnología de seguridad estándar para establecer un enlace cifrado entre un servidor y un cliente, normalmente un servidor web (sitio web) y un navegador, o un servidor de correo y un cliente de correo.

hijos, reescritura de los Localizadores Uniforme de Recursos, comprobación de la ortografía de las URL¹⁷ y manuales en línea. (Apache Software Foundation 2017)

1.5.2 Tecnologías del lado del Cliente

HTML5

HTML, es el lenguaje de marcado predominante usado para estructurar el contenido *web*. En esta última revisión del lenguaje (HTML5) se incluyen varias características verdaderamente útiles, como son nuevos elementos semánticos, mejoras en los formularios, soporte nativo para video y audio, un lienzo para el dibujo a través de JavaScript, soporte para almacenamiento local y geolocalización. (López Trujillo y Marulanda Echeverry 2012)

CSS3

Cascading Style Sheets (CSS) no es más que un lenguaje de estilo que describe cómo el contenido HTML es presentado al usuario. CSS3 es la última versión de la especificación CSS, incluye nuevas características para ayudar a los desarrolladores a resolver problemas sin la necesidad de un marcado no semántico, complejos scripts o imágenes extra. Estas nuevas características incluyen soporte para selectores adicionales, sombras, esquinas redondeadas, múltiples fondos, animaciones, transparencias, entre otras. (López Trujillo y Marulanda Echeverry 2012)

Java Script 1.8.5

JavaScript es un potente lenguaje de *scripting* basado en objetos. El código JavaScript puede ser embebido directamente en un documento HTML, e interpretado por un navegador *web* en el lado del cliente (su uso más extendido). Cuando se combina con el *Document Object Model* (DOM) definido por un navegador *web*, JavaScript permite crear o manipular dinámicamente el contenido HTML. (López Trujillo y Marulanda Echeverry 2012)

jQuery 1.9

¹⁷ URL: Localizador Uniforme de Recursos, denominado coloquialmente una dirección web, es una referencia a un recurso web que especifica su ubicación en una red informática y un mecanismo para recuperarla.

jQuery es una de las bibliotecas JavaScript. Su objetivo fundamental es simplificar algunas tareas comunes, como buscar elementos en el DOM, manipular esos elementos, editar atributos HTML y propiedades CSS, definir manejadores de eventos, entre otras. Cuenta además con utilidades Ajax para realizar peticiones HTML dinámicamente, así como funciones de propósito general para trabajar con objetos y colecciones.(jquery.org 2017)

1.5.3 Herramientas de desarrollo

Durante la etapa de implementación del sistema propuesto se utilizaron herramientas como Git sistema de control de versiones, sin el cual es difícil llevar a cabo tareas como la refactorización del código o prácticas como TDD.(Git 2017)

Como Entorno de Desarrollo Integrado (IDE) se utilizó PyCharm 2016.1, ya que el mismo tiene soporte para Python y Django, además de facilitar los procesos de búsqueda de errores, ejecución de tareas a través de consola, inicio y detención de servidores de desarrollo, utilidades para el sistema de control de versiones, entre muchas otras.(JetBrains 2017)

1.6 ANÁLISIS DE LA POLÍTICA DE COMUNICACIÓN INSTITUCIONAL Y EL MANUAL DE IDENTIDAD DE LA UCI.

Para la mayoría de las personas la comunicación es un fenómeno tan usual y natural que no requiere de mayores complejidades en cuanto a su desarrollo. En el mejor de los casos se llega a considerar la necesidad de un cierto nivel de atención a las acciones comunicativas hacia el exterior de una organización, pero sin llegar a plantearse la necesidad de un enfoque integrador de la comunicación tanto hacia el público externo como hacia el interno, a partir de un enfoque sinérgico y coherente, con objetivos claros y definidos.

La búsqueda de coordinación e integración forman la base de las políticas de comunicación que toda entidad debe asumir para alcanzar una eficacia óptima. La gestión, en tal sentido, tiene como objeto el control de la imagen a través de una práctica comunicativa integral y programada. Integral en cuanto al enfoque sinérgico de todos los recursos e instrumentos de imagen de la organización y programada como medio para eliminar la aleatoriedad e incoherencia en mensajes y discursos comunicativos(DIRECCIÓN DE COMUNICACIÓN INSTITUCIONAL 2014).

La Universidad de las Ciencias Informáticas, establece la comunicación organizacional a partir de una Dirección de Comunicación Institucional que coordina los procesos, para consolidar la identidad e imagen de la UCI. La Dirección de Comunicación Institucional, establece la adopción de políticas de comunicación que expliciten a todos los públicos la transformación que tendrá lugar en la institución y pauten las acciones comunicativas a su interior y en su relación con el entorno, tanto a nivel interinstitucional como con los usuarios de sus servicios. Corresponde a esta Dirección articularse coherentemente con los Objetivos Estratégicos de la Organización de manera que consiga brindar soporte efectivo a su implementación (DIRECCIÓN DE COMUNICACIÓN INSTITUCIONAL 2014).

Teniendo en cuenta la realidad de la Universidad de las Ciencias Informáticas y la necesidad de perfeccionar la imagen del boletín del Centro de Telemática de la Facultad dos se hace un estudio de dichas políticas y del Manual de Diseño de la institución con la finalidad de adecuar la propuesta lo mejor posible a lo establecido por la Dirección de Comunicación Institucional. De este estudio se desprendieron las pautas que rigen los principales canales de flujo de la información dentro del sistema.

Niveles de aprobación:

- Para su publicación o transmisión los contenidos informativos deberán ser aprobados al máximo nivel de la estructura institucional bajo la cual descansa la tutela de la publicación.

Objetivos de la comunicación externa a la que tributará el boletín:

- Posicionar la imagen de la Universidad en sus públicos a partir de la materialización de su misión.
- Crear vínculos con otras instituciones y con la sociedad en general.

El boletín debe ajustarse a las siguientes líneas de comunicación:

- Proceso de desarrollo de *software* en la Universidad y sus resultados más relevantes.
- Divulgación de eventos de carácter nacional e internacional desarrollados en la UCI.

- Papel de avanzada de la UCI en la migración hacia plataformas libres.
- Influencia de la Universidad en el desarrollo e informatización de la sociedad.
- Proyectos relevantes de investigación y desarrollo.
- Sustitución de importaciones y la integración de las producciones nacionales.
- Asimilación de nuevas tecnologías en la educación y la informática.

Los conceptos de identidad e imagen son interdependientes; no hay imagen sin identidad, pues lo que se comunica no puede ser puro diseño, sino que ha de estar anclado necesariamente en la realidad; y al mismo tiempo, no hay representación posible de la identidad si no es a través de la imagen, que constituye su mejor expresión. La imagen es una tarea que afecta a la alta dirección y su gestión exige el compromiso de las distintas áreas funcionales de la organización (DIRECCIÓN DE COMUNICACIÓN INSTITUCIONAL 2014).

El manual de identidad visual es un documento que reúne todas las normas para el uso correcto y reproducción de todos los elementos gráficos que representan una institución. En este manual se especifica con todo detalle el uso de la marca en diferentes medios, digitales o impresos. También suele incluirse información esencial de la empresa como la visión y misión. (Karen Salas 2017)

Del estudio del Manual de Identidad Visual de la UCI se desprendieron las siguientes directrices que deben regir el diseño de las propuestas de plantillas a implementar dentro del boletín.

Competencias claves

- Dominio de la tecnología informática de avanzada para lograr productos y servicios rápidos y de calidad.
- Mejora continua, basada en una organización que aprende.

Atributos

Los atributos de una institución son aquellas características que sus directivos han elegido y aprobado para comunicarlas a sus públicos con vista a cimentar determinada imagen en ellos. Estarán conectados con la realidad, tanto porque la misma los corrobora como porque se visualice en una meta alcanzable por la organización. (DIRECCIÓN DE COMUNICACIÓN INSTITUCIONAL 2014)

- Organización moderna
- Sistema articulado
- Fuerte capacidad tecnológica
- Servicios de excelencia
- Equidad en el acceso y la prestación de los servicios
- Alto compromiso social

Lenguaje formal (Relacionado con la estética y funcionalidad de los mensajes)

Se recomienda el uso de un sistema de formatos para evitar incoherencias en el carácter sistémico de los soportes de comunicación. Se considerarán los signos identificadores como un capital importante de la organización y darles un papel protagónico en los soportes de comunicación emitidos, de manera que se posicione con rapidez en la mente de los públicos. Para lograr este objetivo es de vital importancia asumir como base los atributos y rasgos de estilo definidos en el discurso de identidad de la UCI.(DIRECCIÓN DE COMUNICACIÓN INSTITUCIONAL 2014)

Es importante tomar en cuenta que los materiales que se emplean son elementos portadores de significado en los soportes. Para su selección se contemplarán variables como: acabado, capacidad de deformación, durabilidad, coherencia con el discurso de identidad definido. La adecuada selección contribuirá tanto a la prolongación de la vida útil de los mensajes como a la calidad de los mismos.(Lledó 2016)

1.7 CONCLUSIONES DEL CAPÍTULO

Con el desarrollo del marco teórico se logró organizar y guiar el trabajo, profundizando en el estudio de algunos conceptos fundamentales para el desarrollo de una aplicación *web* que permita informatizar el proceso de creación de boletines informativos para el Centro de Telemática de la Facultad dos de la Universidad de las Ciencias Informáticas. Se realizó un estudio de las aplicaciones similares existentes a nivel internacional, así como el documento que rige las políticas de comunicación de la universidad. Además, se seleccionaron las tecnologías, herramientas, lenguajes y técnicas de desarrollo propuestas para la solución del problema actual.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

En este capítulo se describirán las características del sistema a desarrollar. Se hace un análisis de los procesos del negocio que se desean informatizar, lo cual posibilita que se tenga un mayor conocimiento del mismo, en vista de lograr un sistema que cumpla con las necesidades del cliente. Además, se describen las principales características del sistema que fundamenta esta investigación.

2.1 ANÁLISIS DE EL PROCESO DE CREACIÓN Y MONTAJE DEL BOLETÍN DEL CENTRO DE TELEMÁTICA

En la actualidad el procedimiento de creación y montaje del boletín del centro TLM de la Facultad dos de la Universidad de las Ciencias Informáticas se realiza con la herramienta Adobe InDesing. Los principales procedimientos que ocurren durante este proceso son los siguientes:

1. Análisis de los intereses informativos del centro.
2. Recogida de la información a publicar.
3. Elaboración de los bocetos y plantillas acordes a la identidad del centro.
4. Montaje de la información promocional y científica sobre las plantillas.
5. Generar el boletín.
6. Revisión del boletín.
7. Publicación del boletín en los medios digitales.
8. Envío del boletín a través de las listas de usuarios definidas.

Las tareas 4, 5 y 6 representan un ciclo pues el boletín debe ser revisado por la dirección del centro hasta que pase por la aprobación de todos los miembros del consejo de dirección. Este proceso puede ser largo, tedioso y conlleva a que el boletín se genere un número considerable de veces, esto repercute en que la información no llegue de la manera más inmediata a los suscriptores lo cual lo hará totalmente ineficiente.

En la presente investigación se plantea la automatización de los procesos:

1. Recogida de la información a publicar.
2. Montaje de la información promocional y científica sobre las plantillas.
3. Generación del boletín.

4. Revisión del boletín.
5. Envío del boletín a través de las listas de usuarios definidas.

2.2 CARACTERÍSTICAS DE LA PROPUESTA DEL SISTEMA

En la presente investigación se propone el desarrollo de una plataforma *web* con las principales funcionalidades que permitan la gestión de las estructuras de un documento, también que faciliten la gestión del contenido informativo a plasmar o publicar, que viabilicen la gestión de listas de correo para el posterior envío del boletín, y posibiliten la obtención del resultado final del montaje en formato PDF. La misma estará programada fundamentalmente en Python y Django, posibilitando el acceso a los usuarios a través de un sistema de autenticación basado en *tokens*¹⁸. A la aplicación se podrá acceder mediante un sistema de autenticación que comprobará las credenciales del usuario a través de un servicio *web*. Luego de la verificación de las credenciales se les otorgará acceso a los diferentes módulos del sistema según su rol. Además, debe permitir la gestión de roles.

La recogida de información se realizará a partir de un módulo donde se podrán establecer los diferentes tipos de información (noticia, evento, información, nota, promoción). Cada información será relacionada con un tipo de plantilla automáticamente y podrá ser revisada por los usuarios con permiso de acceder a ella. Al módulo tendrán acceso los miembros del consejo de dirección del centro y el grupo de mercadotecnia.

El montaje se realizará en el módulo de maquetación donde se podrán seleccionar las diferentes plantillas e informaciones que compondrán la edición del boletín. El módulo permitirá guardar la edición como proyecto para su posterior revisión y modificación. El sistema podrá generar el boletín en formato PDF haciendo uso de la librería ReportLab. Además de gestionar el envío del mismo a través de listas de correo.

En la figura 2.1 se representa el funcionamiento de la propuesta del sistema.

¹⁸ Tokens: En seguridad informática, es un mecanismo que se le da a un usuario autorizado de un servicio computarizado para facilitar el proceso de autenticación.

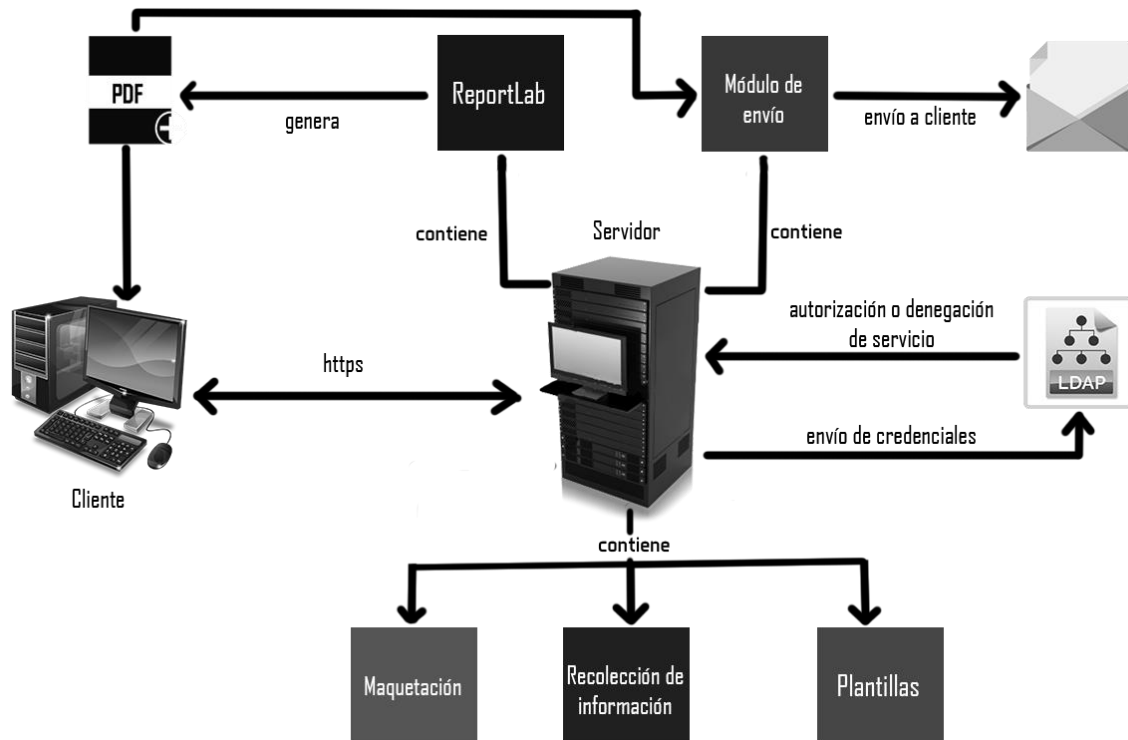


Figura 2.1: Propuesta de sistema.

2.3 CARACTERÍSTICAS NO FUNCIONALES DEL SISTEMA

Las características no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable(Cockburn 2000).

Usabilidad

- Los elementos que conforman una interfaz de usuario, entiéndase imágenes, botones, campos de entrada de datos, deben tener un nombre sugerente, que logre que el usuario perciba la acción, además de mostrar una breve descripción de su función en la propia interfaz o con solo pasarle el cursor por encima.
- En el caso de los formularios deben especificar las reglas de sus campos, como pueden ser los obligatorios, aquellos que solo aceptan tipos de datos específicos o longitudes de cadenas limitadas.
- Se debe garantizar que los contenidos tengan validez gramatical y correcta ortografía.

- El sistema debe contar con un estilo de presentación coherente para cada página.
- El sistema debe presentar una interfaz amigable que permita una fácil interacción. Debe, además, ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación.

Software

En el cliente, aunque solo precisa de un navegador *web*, es necesario tener en consideración cuáles son adecuados y recomendables para hacer uso del sistema:

- Mozilla Firefox (versión 35 o superior).
- Chrome (versión 30 o superior).

Hardware

Para el correcto funcionamiento del módulo se requiere para el servidor:

- PC Dual Core o superior.
- CPU 1.9 GHz o superior.
- 4 GB de RAM o superior.
- 250 GB de disco duro o superior.

Seguridad

Confidencialidad: La información manejada por el sistema está protegida de acceso y divulgación no autorizada. Esto se deberá lograr mediante el uso e implementación de un control de usuarios.

Integridad: La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes. Esto deberá garantizarse mediante el uso de la integridad referencial característica básica que brinda PostgreSQL. La integridad referencial es una funcionalidad disponible en las bases de datos relacionales que garantiza la coherencia de datos entre relaciones aparejadas.

Disponibilidad: A los usuarios autorizados les será garantizado el acceso a la información y los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado. El servidor deberá contar con un sistema de respaldo capaz de

realizar una salva de la base de datos cada cierto tiempo para en caso de malfuncionamiento o daños poder restaurarla.

2.4 CONCLUSIONES DEL CAPÍTULO

En el presente capítulo se realizó un análisis sobre el proceso de montaje del boletín informativo del centro TLM, el cual permitió definir los procesos que deben ser automatizados para dar solución al problema planteado en esta investigación. Además se analizó de forma detallada el funcionamiento del sistema propuesto como solución, haciendo énfasis en las características de la propuesta del sistema y las características no funcionales que debe poseer el mismo para lograr el objetivo de la presente investigación.

CAPÍTULO 3: EXPLORACIÓN, PLANIFICACIÓN Y DISEÑO

En el presente capítulo se presentan los artefactos obtenidos en las fases de Exploración, Planificación y Diseño correspondientes a la metodología de desarrollo seleccionada (XP) para la implementación del sistema que se propone.

3.1 EXPLORACIÓN

Esta fase es la primera del proceso de desarrollo de *software* que define XP. En ella los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología. (Kent Beck 2000)

3.1.1 Historias de usuario

Las HU¹⁹ son la forma en que se especifican los requisitos del sistema con el uso de la metodología XP. Estas se escriben desde la perspectiva del cliente, aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas. (Kent Beck 2000) Son escritas en lenguaje natural, sin un formato predeterminado, no excediendo su tamaño de unas pocas líneas de texto. Además, guían la construcción de las pruebas de aceptación y son utilizadas para estimar tiempos de desarrollo.

Durante la fase de exploración se identificaron las siguientes historias de usuario:

- Autenticación de usuario.
- Gestionar contenido informativo.
- Gestionar usuario.
- Gestionar lista de correo.
- Maquetación de boletín.
- Exportar archivo a formato PDF.

¹⁹ HU: Historia de Usuario

Las HU definidas se clasificaron atendiendo a diferentes criterios, facilitando así su organización y la correcta planificación del proyecto. Los parámetros definidos en cada una de las HU se detallan a continuación.

- **Número:** Para mantener el orden de las historias.
- **Nombre:** Para identificar la HU.
- **Prioridad:**
 1. **Alta:** Constituyen funcionalidades principales del sistema, o forman parte esencial de la arquitectura del mismo.
 2. **Media:** Son funcionalidades importantes y de gran valor para el usuario pero que no impiden poner el proyecto en marcha si no se tienen.
 3. **Baja:** Son funcionalidades que sería deseable tener y podrían incluirse en caso de que hubiese recursos para ello.
- **Riesgo en desarrollo:**
 1. **Alto:** En caso de tener algún error de implementación, pueden afectar la disponibilidad del sistema.
 2. **Medio:** En caso de presentar errores retrasan la entrega de la versión.
 3. **Bajo:** En caso de presentar errores, estos pueden ser tratados con facilidad y no afectan el desarrollo del proyecto.
- **Iteración asignada:** Número de la iteración a la que ha sido asignada.
- **Puntos estimados:** Tiempo estimado de desarrollo para completar la HU. Un punto equivale a una semana ideal de programación (40 horas).
- **Descripción:** Es donde se define la funcionalidad que se quiere desarrollar.
- **Observaciones:** Detalles a tener en cuenta para desarrollar la HU correctamente.
- **Prototipo de interfaz:** Es un boceto representativo de la vista de usuario. (Kent Beck 2000)

Siguiendo las pautas planteadas por XP se definen las historias de usuarios correspondientes a la aplicación.

Número: 1	Nombre: Autenticación de usuario
Prioridad: Alta	Iteración Asignada: 1

Riesgo en desarrollo: Alto	Puntos estimados : 0.5
Descripción: Interfaz mediante la cual un usuario puede acceder al sistema con su nombre de usuario y contraseña del dominio UCI.	
Observaciones: Un usuario puede acceder al sistema, siempre y cuando el servidor LDAP de la UCI esté disponible, o ya se haya autenticado anteriormente.	
Prototipo:	

Tabla 3.1: Historia de Usuario - Autenticación de usuario

Número: 2	Nombre: Maquetación de Boletín.
Prioridad: Alta	Iteración Asignada: 1
Riesgo en desarrollo: Alto	Puntos estimados: 2
Descripción: Permite al usuario montar y ubicar la información dentro del boletín, así como determinar el número de páginas y el formato de las mismas.	
Observaciones:	

Los boletines no deben llamarse igual, y los artículos que lo compongan deben haber sido aprobados.

Prototipo:

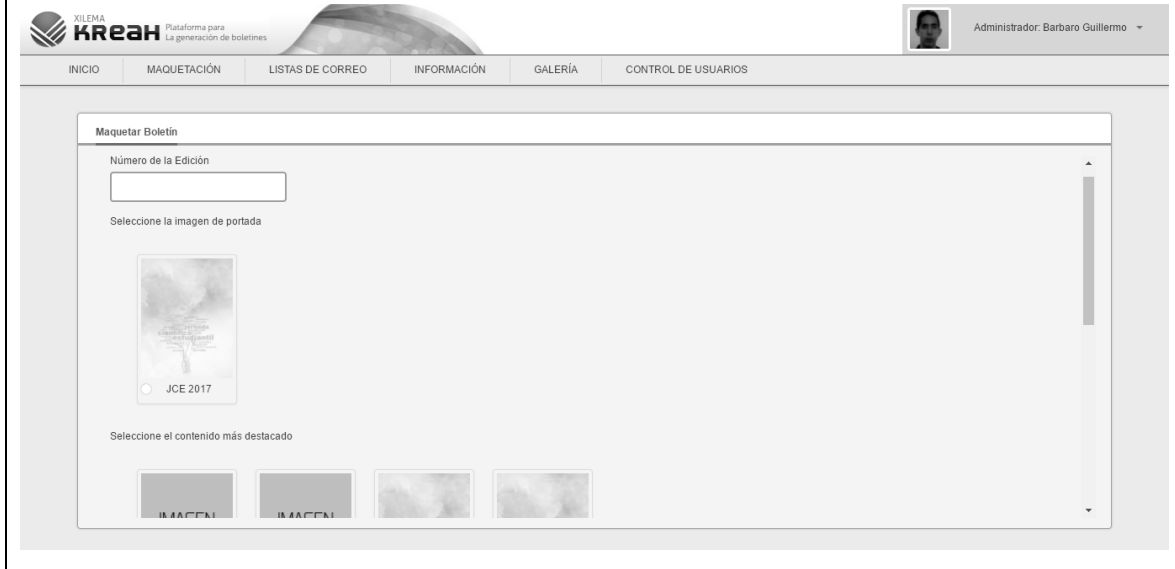


Tabla 3.2: Historia de Usuario - Maquetación de Boletín

Número: 3	Nombre: Exportar archivo a formato PDF.
Prioridad: Alta	Iteración Asignada: 1
Riesgo en desarrollo: Alto	Puntos estimados: 0.5
Descripción: Permite obtener el resultado de la maquetación en un archivo PDF en una ubicación designada por el usuario, además del envío inmediato vía correo electrónico.	
Observaciones: El archivo debe contener los metadatos del autor, en este caso los datos de contacto y referencia del centro.	
Prototipo:	

The image shows a user interface for exporting a PDF. At the top, there is a dark header with the text 'EXPORTAR PDF'. Below this, there are two input fields: 'Nombre de Archivo' and 'Ruta de Salida'. The 'Ruta de Salida' field has a small menu icon (three dots) on its right side. Below the input fields is a checkbox labeled 'Envío Inmediato'. Underneath is a section titled 'Listas de Envío' which contains two cards, each with an '@' icon and the text 'LISTA 1 30 MIEMBROS' and 'LISTA 2 30 MIEMBROS' respectively. To the right of these cards is a dashed box containing a plus sign '+'. At the bottom right of the form are two buttons: 'CANCELAR' and 'EXPORTAR'.

Tabla 3.3: Historia de Usuario - Exportar archivo a formato PDF

Número: 4	Nombre: Gestionar contenido informativo.
Prioridad: Media	Iteración Asignada: 2
Riesgo en desarrollo: Alto	Puntos estimados: 1
Descripción: Permite al usuario gestionar (crear, editar, visualizar, eliminar) la información a insertar en el boletín.	
Observaciones: Las informaciones no deben repetirse, y deben pasar por los estados de revisión establecidos(Revisión, Aprobado).	
Prototipo:	

The screenshot shows the XILEMA KReah platform interface. At the top, there is a navigation menu with items: INICIO, MAQUETACIÓN, LISTAS DE CORREO, INFORMACIÓN, GALERÍA, and CONTROL DE USUARIOS. The user is logged in as 'Administrador: Barbaro Guillermo'. Below the menu is a search bar with the text 'Buscar:' and a dropdown menu for 'Ver' set to '10' 'entradas'. The main content area displays a table of articles with columns for 'Titulo', 'Autor', 'Estado', and 'Opciones'. The table contains four rows of article data, each with 'Aprobado' status and three action buttons: 'VISUALIZAR', 'MODIFICAR', and 'ELIMINAR'. At the bottom of the table, it says 'Mostrando 4 de 4 entradas'.

Tabla 3.4: Historia de Usuario - Gestionar contenido informativo

Número: 5	Nombre: Gestión de Usuario.
Prioridad: Media	Iteración Asignada: 2
Riesgo en desarrollo: Alto	Puntos estimados: 1.5
Descripción: Permite al usuario con rol de administración gestionar(eliminar, modificar, visualizar) en el sistema los usuarios del mismo.	
Observaciones: Las usuarios son únicos y poseen un solo rol dentro del sistema.	
Prototipo:	

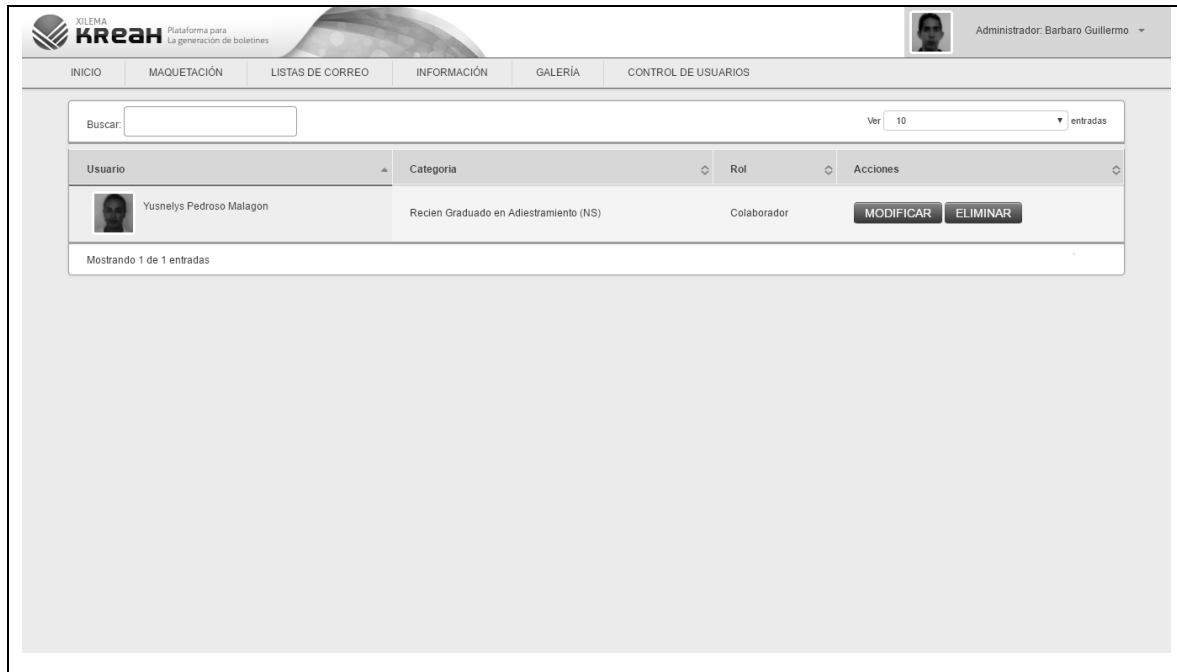


Tabla 3.5: Historia de Usuario - Gestión de Usuario.

Número: 6	Nombre: Gestionar lista de correo		
Prioridad: Baja		Iteración Asignada: 3	
Riesgo en desarrollo: Alto		Puntos estimados: 1.5	
Descripción: Un usuario puede gestionar (crear, editar, visualizar, eliminar) las listas de correo del sistema.			
Observaciones: Los contactos no deben repetirse, así como poseer direcciones de correo inválidas.			
Prototipo:			

ADICIONAR CONTACTO

Nombre de Contacto

Dirección de Correo Electrónico

Compañía

Lista de Correo Asociada

Descripción

CANCELAR ADICIONAR

Tabla 3.6: Historia de Usuario - Gestionar lista de correo

3.2 PLANIFICACIÓN

En esta fase el cliente establece la prioridad de cada historia de usuario, y en correspondencia, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Las estimaciones de esfuerzo asociado a la implementación de las HU la establecen los programadores utilizando como medida el punto. Un punto equivale a una semana ideal de programación o 40 horas. (Kent Beck 2000)

3.2.1 Estimación del esfuerzo por historias de usuario

Para el desarrollo del sistema propuesto se realizó una estimación del esfuerzo para cada una de las HU identificadas, llegándose a los resultados que se muestran en la siguiente tabla.

HU	Puntos Estimados
Autenticación de usuario	0.5
Maquetación de Boletín	2

Exportar archivo a formato PDF	0.5
Gestionar contenido informativo	1
Gestionar Usuario	1.5
Gestionar lista de correo	1.5

Tabla 3.7: Estimación del esfuerzo por HU.

3.2.2 Plan de iteraciones

En el plan de iteraciones se establece cuántas iteraciones serán necesarias realizar sobre el sistema para su desarrollo. Sin embargo, se precisa establecer el contenido de trabajo para todas y cada una de ellas, es aquí donde se establece el plan de iteraciones, regulando la cantidad de historias de usuario a implementar dentro del rango establecido por la estimación efectuada. Tomando como referencia los aspectos antes tratados, la aplicación que se pretende construir se desarrollará en 3 iteraciones, explicada más detalladamente a continuación:

Iteración 1

Esta iteración tiene como objetivo la implementación de las historias de usuario Autenticación de usuario, Maquetación de Boletín, Exportar archivo a formato PDF que poseen mayor prioridad. Durante el transcurso de la misma se creará la base de la arquitectura del sistema con una funcionalidad mínima. Al final de esta se contará con una primera versión de prueba, la cual será mostrada al cliente con el objetivo de obtener una retroalimentación para el equipo de desarrollo.

Iteración 2

El objetivo de esta iteración es la implementación de las historias de usuario Gestionar contenido informativo y Gestionar usuario, que poseen prioridad media. Al finalizar la misma se contará con una versión funcional del sistema, donde se podrán probar las funcionalidades relacionadas con la selección y refinamiento del contenido informativo de interés. Esta versión igualmente se mostrará al cliente con el objetivo de realizar cambios necesarios en base a la opinión del mismo.

Iteración 3

Durante el transcurso de esta iteración se implementan la historia de usuario Gestionar lista de correo que posee baja prioridad. Al finalizar la misma se contará

con la versión 1.0 del producto final, adicionando a las funcionalidades anteriores todo lo concerniente al sistema de distribución de la información. Como resultado de esta, el sistema estará listo para su despliegue.

Duración de las iteraciones

Como parte del ciclo de vida de un proyecto que utiliza XP se crea el plan de duración de cada una de las iteraciones, en este caso se hace para el único equipo de desarrollo con que se cuenta. Este plan se encarga de mostrar las historias de usuario que serán abordadas en cada una de las iteraciones, así como la duración estimada de estas últimas y el orden en que se implementarán las HU.

Iteración	HU a implementar	Duración total de iteración
Iteración 1	Autenticación de usuario	3 semanas
	Maquetación de Boletín	
	Exportar archivo a formato PDF	
Iteración 2	Gestionar contenido informativo	2.5 semanas
	Gestionar Usuario	
Iteración 3	Gestionar lista de correo	1.5 semanas

Tabla 3.8: Plan de duración de las iteraciones.

3.2.3 Plan de entregas

Se presenta el plan de entregas elaborado para la fase de implementación, que comienza el lunes 6 de febrero de 2017. Como producto del mismo se harán entregas incrementales del sistema al finalizar cada iteración, en la fecha aproximada que se indica en la siguiente tabla.

Iteración	Fecha de Entrega
Iteración 1	27/02/2017
Iteración 2	15/03/2017
Iteración 3	27/03/2017

Tabla 3.9: Plan de entrega de las iteraciones.

3.3 DISEÑO

Durante el diseño de la solución, la máxima simplicidad posible es la clave para el éxito de XP. Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra deben ser evitados en todo momento. (Cockburn 2000)

El diseño adecuado para el *software* es aquel que supera con éxito todas las pruebas, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos. Se debe tener en cuenta que un diseño complejo siempre tarda más en desarrollarse que uno simple, y que siempre es más fácil añadir complejidad a un diseño simple que quitarla de uno complejo. (Pressman 2005)

3.3.1 Patrón arquitectónico

Los patrones arquitectónicos se utilizan para expresar una estructura de organización base o esquema para un *software*. Proporcionando un conjunto de sub-sistemas predefinidos, especificando sus responsabilidades, reglas, directrices que determinan la organización, comunicación, interacción y relaciones entre ellos. Los patrones arquitectónicos heredan mucha de la terminología y conceptos de patrones de diseño, pero se centran en proporcionar modelos y métodos re-utilizables específicamente para la arquitectura general de los sistemas de información. En otras palabras, quiere decir que a diferencia de los patrones de diseño estas son plantillas incompletas y no se pueden aplicar directamente al código con modificaciones meramente contextuales. (ingeniods 2013)

Modelo Plantilla Vista (MTV)

Django, el *framework* de desarrollo seleccionado del lado del servidor, respeta el paradigma conocido como Modelo Plantilla Vista (MTV), el cual está fuertemente inspirado en la filosofía de desarrollo Modelo Vista Controlador (MVC), con ciertas diferencias. Como resultado, lo que se llamaría controlador en un verdadero *framework* MVC se llama en Django vista, y lo que se llamaría vista se llama plantilla. Para un mejor entendimiento se presenta el siguiente diagrama:

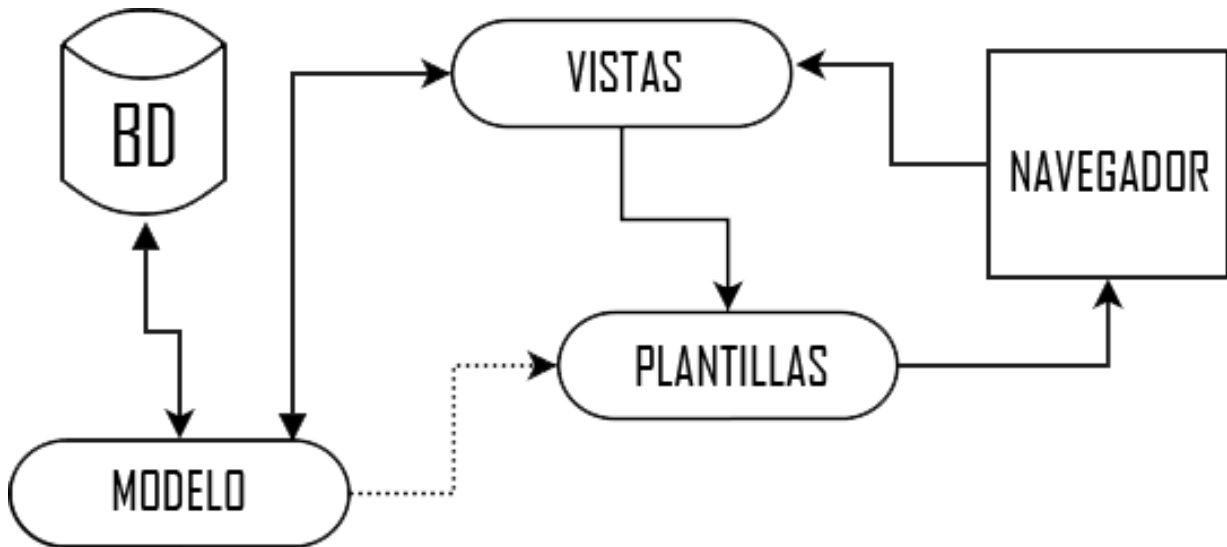


Figura 3.1: Diagrama del patrón MTV.

El patrón MTV al igual que el MVC permite separar los datos, la lógica de la aplicación y la presentación gráfica con la finalidad de conseguir un código limpio y un fácil mantenimiento ya que las modificaciones de una parte de la aplicación no afectan a las demás.(Django 2017)

3.3.2 Patrones de diseño

El término de patrón fue dado por primera vez en el año 1977 por el arquitecto Christopher Alexander, quien dio en su libro *A Pattern Language* la siguiente definición: “Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo ni siquiera dos veces de la misma forma”. En la ingeniería del *software*, un patrón constituye el apoyo para la solución a los problemas más comunes que se presentan durante las diferentes etapas del ciclo de vida del *software*.(Guerrero, Suárez y Gutiérrez 2013)

Para el diseño de la aplicación se hizo uso de los Patrones Generales de *Software* para Asignar Responsabilidades (GRASP), así como de los patrones *Gang of Four* o Pandilla de los Cuatro (GoF).

Los patrones GRASP utilizados fueron:

- **Alta cohesión:** Se aplica en la mayoría las clases del diseño, ya que en cada una solo se implementan las funcionalidades que le corresponden.
- **Bajo acoplamiento:** Se evidencia en todas las clases ya que cada una se comunica con un número relativamente pequeño de clases.
- **Creador:** Las clases que tienen la responsabilidad de crear objetos contienen toda la información necesaria para construir los mismos. Se evidencia en la clase ReportLabController.
- **Experto:** Este patrón se evidencia en todas las clases. Se mantiene el encapsulamiento, los objetos utilizan su propia información para llevar a cabo sus tareas. Se distribuye el comportamiento entre las clases que contienen la información requerida. Son más fáciles de entender y mantener.

Ahora bien, los patrones de diseño GoF “describen soluciones simples y elegantes a problemas específicos en el diseño de *software* orientado a objetos”.(Larman Craig 2001) Los utilizados en el diseño de la solución fueron:

- **Front Controller (Controlador Frontal):** Django posee una implementación de Controlador Frontal que despacha las peticiones hacia métodos o clases, que en la práctica son páginas controladoras. Antes del despacho, la petición es procesada por varios filtros (*middlewares*).
- **Singleton (Patrón Solitario):** es un patrón de tipo creacional con el objetivo de garantizar la existencia de una única instancia para una clase y posibilitar el acceso global a dicha instancia. Su empleo se evidencia cuando se realiza una conexión LDAP al servidor, pues con una única instancia del objeto conexión se realizan las operaciones sobre este. Tal es el caso de la clase UCILDAP, donde con una única instancia se realizan todas las operaciones LDAP en el servidor.
- **Remote agent (Agente remoto):** Crea una clase de servicio local la cual representará el servicio externo y tendrá la responsabilidad de contactarlo además de administrarlo. Su empleo se evidencia en la clase MailingService la cual es una extensión del servicio de correo a usar y sus principales funcionalidades.

3.3.3 Tarjetas CRC

En la fase de Diseño, la metodología XP no requiere la representación del sistema mediante diagramas de clases utilizando notación Lenguaje Unificado de Modelado (UML). En su lugar se usan otras técnicas como las tarjetas Clase Responsabilidad Colaboración (CRC) como una extensión informal a UML. La técnica de las tarjetas CRC se puede usar para guiar el sistema a través de análisis guiados por la responsabilidad. Las clases se examinan, se filtran y se refinan en base a sus responsabilidades con respecto al sistema, y las clases con las que necesitan colaborar para completar sus responsabilidades. (Wallace, Raggett y Aufgang 2002) Se muestran las tarjetas CRC obtenidas durante la etapa de diseño de la presente investigación.

Tarjeta CRC #1	
Clase: Models	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Guardar y recuperar desde la Base de Datos la información asociada a cada uno de los modelos del sistema. • Definir el modelo de datos a través del ORM de Django. 	<ul style="list-style-type: none"> • django.db.models

Tabla 3.10: Tarjeta CRC #1.

Tarjeta CRC #2	
Clase: Views	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Proveer las APIs correspondientes al sistema de autenticación y los modelos del sistema. 	<ul style="list-style-type: none"> • Models • UCILDAP • MailingService • ReportLabController

<ul style="list-style-type: none"> • Gestionar adecuadamente las peticiones <i>GET</i>, <i>POST</i>, <i>PUT</i>, <i>DELETE</i> y <i>OPTIONS</i> para cada una de las APIs generadas. 	
---	--

Tabla 3.11: Tarjeta CRC #2.

Tarjeta CRC #3	
Clase: UCILDAP	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Realizar la autenticación a través del servicio LDAP de la UCI y gestionar la información correspondiente del usuario. 	<ul style="list-style-type: none"> • Models

Tabla 3.12: Tarjeta CRC #3.

Tarjeta CRC #4	
Clase: ReportLabController	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Gestionar adecuadamente las peticiones y llamadas a la librería de dibujo ReportLab.py. 	<ul style="list-style-type: none"> • Models

Tabla 3.13: Tarjeta CRC #4.

Tarjeta CRC #5	
Clase: MailingService	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Gestionar adecuadamente el envío de información vía correo usando la librería SMTPlib de Python. 	<ul style="list-style-type: none"> • Models

Tabla 3.14: Tarjeta CRC #5.

3.3 CONCLUSIONES DEL CAPÍTULO

Se identificaron seis HU; de ellas tres fueron catalogadas de prioridad alta, dos de prioridad media y una de prioridad baja. Se elaboraron los artefactos correspondientes a las etapas de Exploración, Planificación y Diseño, definiéndose además las funcionalidades que tendrá el sistema, así como la planificación del esfuerzo dedicado a la realización de cada una de estas en el orden en que se les dará cumplimiento según las necesidades del cliente. Además, permitió definir una fecha estimada en la que se le entregará una primera versión del producto al cliente. Se definieron los patrones arquitectónicos y de diseño usados con el objetivo de lograr una mayor organización en los elementos que conforman la aplicación, así como las tarjetas CRC para una mejor comprensión de la misma.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

La metodología XP plantea que el proceso de desarrollo del *software* debe realizarse de forma iterativa, obteniendo al culminar cada iteración un producto funcional que debe ser probado y mostrado al cliente para retroalimentar a los desarrolladores con la opinión de este. En el presente capítulo se describen los estándares de codificación utilizados para obtener un código limpio y legible. Se detallan además las tres iteraciones llevadas a cabo durante la etapa de construcción del sistema, exponiendo las tareas generadas por cada HU, así como las pruebas unitarias y de aceptación efectuadas sobre la aplicación.

4.1 ESTÁNDARES DE CODIFICACIÓN

Según Guido van Rossum (científico de la computación, conocido por ser el autor del lenguaje de programación Python), el código es leído muchas más veces de lo que es escrito. XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios. (Patricio Letelier y M^a Carmen Penadés Patricio 2014)

Por tanto, se hace necesario definir pautas para lograr una mejor legibilidad del código y hacerlo consistente. Para lograr este objetivo se utilizó la Guía de estilo para el código Python (PEP 257). («PEP 257 -- Docstring Conventions» 2017) Esta guía posee una gran cantidad de convenciones para escribir código legible, dentro de las cuales se destacan:

- Usar cuatro espacios por indentación.
- Limitar todas las líneas a un máximo de caracteres (120 en este proyecto).
- Separar funciones de alto nivel y definiciones de clase con dos líneas en blanco, mientras que las definiciones de métodos dentro de una clase son separadas por una línea en blanco.
- Las importaciones deben estar en líneas separadas.
- Evitar usar espacios en blanco innecesarios.

- Utilizar el estilo CamelCase para nombrar clases, y el lower_case_with_underscores para funciones y métodos.

4.2 ITERACIONES

Durante el transcurso de las iteraciones se realiza la implementación de las historias de usuario seleccionadas para cada una de estas y se llevan a cabo las pruebas unitarias y de aceptación asociadas a cada una de las mismas. Al iniciar cada iteración, se lleva a cabo una revisión del plan de iteraciones y se modifica de ser necesario. Como parte de este plan, se descomponen las HU en tareas de ingeniería. Estas tareas son para el uso de los programadores, pueden escribirse utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente. (Kent Beck 2000)

Siguiendo la planificación realizada, se llevaron a cabo tres iteraciones de desarrollo sobre el sistema, obteniéndose al finalizar un producto listo para su despliegue. A continuación, se detallan cada una de las iteraciones.

4.2.1 ITERACIÓN 1

Durante esta iteración se abordaron las historias de usuario de mayor prioridad y se construyó la base de la arquitectura del sistema con el fin de obtener un producto con las funcionalidades críticas para ser mostrado al cliente y obtener una rápida retroalimentación de este.

HU - Autenticación de usuario

Tarea de Ingeniería	
Nombre: API de autenticación	
HU: Autenticación de usuario	Puntos Estimados: 0.2
Fecha Inicio: 06/02/2017	Fecha Fin: 06/02/2017
Descripción: Se debe proveer una API de autenticación basada en <i>tokens</i> , donde las credenciales a recibir son el nombre de usuario y contraseña del dominio UCI.	

Tabla 4.1: Tarea de Ingeniería - API de autenticación

Tarea de Ingeniería

Nombre: Autenticar usuario	
HU: Autenticación de usuario	Puntos Estimados: 0.2
Fecha Inicio: 07/02/2017	Fecha Fin: 07/02/2017
Descripción: Se debe mostrar un formulario de autenticación para que el usuario entre sus credenciales (nombre de usuario y contraseña). Los datos deberán ser válidos para autenticarse, mostrando un mensaje de error en caso contrario. No se permitirá el acceso a las funcionalidades del sistema hasta que el usuario no se haya autenticado.	

Tabla 4.2: Tarea de Ingeniería - Autenticar usuario

Tarea de Ingeniería	
Nombre: Cerrar sesión	
HU: Autenticación de usuario	Puntos Estimados: 0.1
Fecha Inicio: 08/02/2017	Fecha Fin: 08/02/2017
Descripción: Se debe permitir al usuario cerrar la sesión desde cualquier página, rediriéndolo inmediatamente hacia la página de acceso.	

Tabla 4.3: Tarea de Ingeniería - Cerrar sesión

HU - Maquetación de Boletín.

Tarea de Ingeniería	
Nombre: API para los boletines	
HU: Maquetación de Boletín.	Puntos Estimados: 0.5
Fecha Inicio: 08/02/2017	Fecha Fin: 10/02/2017
Descripción: Se debe proveer una API para gestionar los boletines creados por los usuarios, garantizando un correcto funcionamiento de los procesos de adición de páginas y monitoreo de la información.	

Tabla 4.4: Tarea de Ingeniería - API para los boletines.

Tarea de Ingeniería	
Nombre: Listar Boletines	
HU: Maquetación de Boletín.	Puntos Estimados: 0.1
Fecha Inicio: 13/02/2017	Fecha Fin: 13/02/2017
Descripción: Un usuario puede ver una tabla con los boletines que tiene guardados el sistema y un resumen de cada uno (nombre y estado), permitiendo gestionar (editar, mostrar, eliminar) cada uno de ellos.	

Tabla 4.5: Tarea de Ingeniería - Listar Boletines.

Tarea de Ingeniería	
Nombre: Añadir página	
HU: Maquetación de Boletín.	Puntos Estimados: 0.4
Fecha Inicio: 13/02/2017	Fecha Fin: 15/02/2017
Descripción: Un usuario puede adicionar una página a un boletín junto a su contenido.	

Tabla 4.6: Tarea de Ingeniería - Añadir página.

Tarea de Ingeniería	
Nombre: Mostrar página	
HU: Maquetación de Boletín.	Puntos Estimados: 0.2
Fecha Inicio: 15/02/2017	Fecha Fin: 16/02/2017
Descripción: Un usuario puede ver el contenido asociado a una página de su elección. Se deben cargar los archivos externos para una correcta visualización del contenido.	

Tabla 4.7: Tarea de Ingeniería - Mostrar página.

Tarea de Ingeniería	
Nombre: Eliminar página	
HU: Maquetación de Boletín.	Puntos Estimados: 0.4

Fecha Inicio: 16/02/2017	Fecha Fin: 20/02/2017
Descripción: Un usuario puede eliminar una página de su boletín. Los contenidos asociados a la página no serán eliminados. Esta acción debe confirmarse debido a que es irreversible.	

Tabla 4.8: Tarea de Ingeniería - Eliminar página

Tarea de Ingeniería	
Nombre: Crear Boletín	
HU: Maquetación de Boletín.	Puntos Estimados: 0.2
Fecha Inicio: 20/02/2017	Fecha Fin: 21/02/2017
Descripción: Un usuario puede crear su boletín. Debe insertar el nombre del proyecto y la edición del mismo.	

Tabla 4.9: Tarea de Ingeniería - Crear Boletín

Tarea de Ingeniería	
Nombre: Modificar Boletín	
HU: Maquetación de Boletín.	Puntos Estimados: 0.1
Fecha Inicio: 21/02/2017	Fecha Fin: 21/02/2017
Descripción: Un usuario puede modificar su boletín.	

Tabla 4.10: Tarea de Ingeniería - Modificar Boletín

Tarea de Ingeniería	
Nombre: Eliminar Boletín	
HU: Maquetación de Boletín.	Puntos Estimados: 0.1
Fecha Inicio: 22/02/2017	Fecha Fin: 22/02/2017
Descripción: Un usuario puede eliminar su boletín. Los contenidos asociados al mismo no serán eliminados. Esta acción debe confirmarse debido a que es irreversible.	

Tabla 4.11: Tarea de Ingeniería - Eliminar Boletín

HU - Exportar archivo a formato PDF

Tarea de Ingeniería	
Nombre: API Exportación PDF	
HU: Exportar archivo a formato PDF.	Puntos Estimados: 0.3
Fecha Inicio: 23/02/2017	Fecha Fin: 24/02/2017
Descripción: Se debe proveer una API para gestionar la exportación de los boletines creados por los usuarios, garantizando una correcta obtención del contenido plasmado. Se debe garantizar la comunicación con la librería ReportLab.	

Tabla 4.12: Tarea de Ingeniería - API Exportación PDF.

Tarea de Ingeniería	
Nombre: Exportación PDF	
HU: Exportar archivo a formato PDF.	Puntos Estimados: 0.2
Fecha Inicio: 24/02/2017	Fecha Fin: 27/02/2017
Descripción: Se mostrará un formulario donde el usuario podrá escoger la ruta de salida del boletín , el nombre del mismo, y si desea enviarlo de inmediato y a cuál lista de distribución.	

Tabla 4.13: Tarea de Ingeniería - Exportación PDF.

Pruebas unitarias

En esta iteración se realizaron 18 pruebas unitarias en el lado del servidor. Los resultados de su ejecución con el entorno integrado de desarrollo pueden observarse en la figura 4.1.

```

Creating test database for alias 'default'
.....
-----
Ran 18 tests in 3.023s

OK
Destroying test database for alias 'default'

```

Figura 4.1: Resultados de las pruebas unitarias de la primera iteración

Pruebas de Aceptación

Para realizar las pruebas de aceptación correctamente se elaboraron casos de prueba para cada una de ellas. Seguidamente se describe el caso de prueba correspondiente a la HU Autenticación de usuario. Los demás deben consultarse en el [Anexo 1](#).

Caso de Prueba de Aceptación	
Nombre:	Autenticación - Datos válidos
Historia de Usuario:	Autenticación de usuario
Descripción:	Se prueba el acceso al sistema con credenciales correctas. Comprobando los elementos del escenario correspondiente.
Precondiciones:	Se deben proveer credenciales correctas.
Pasos de ejecución:	<ul style="list-style-type: none"> • Acceder al sistema con credenciales correctas. • Cerrar sesión.

Tabla 4.14: Caso de Prueba de Aceptación - Autenticación - Datos válidos

Caso de Prueba de Aceptación	
Nombre:	Autenticación - Datos Inválidos
Historia de Usuario:	Autenticación de usuario
Descripción:	Se prueba el acceso al sistema con credenciales incorrectas. Comprobando los elementos del escenario correspondiente.
Precondiciones:	--
Pasos de ejecución:	<ul style="list-style-type: none"> • Intentar acceder a una página restringida sin autenticarse. • Intentar acceder con credenciales incorrectas.

Tabla 4.15: Caso de Prueba de Aceptación - Autenticación - Datos Inválidos

En esta iteración se realizaron 6 pruebas de aceptación, dos por cada HU a implementar. A continuación, se muestran los resultados arrojados durante la ejecución de las pruebas en conjunto.



Gráfico 4.1: Resultados de las pruebas de aceptación

4.2.2 ITERACIÓN 2

Durante esta iteración se abordaron las historias de usuario de mediana prioridad y se implementaron las funcionalidades asociadas al manejo de la información y los usuarios.

HU - Gestionar contenido informativo.

Tarea de Ingeniería	
Nombre: API de Información	
HU: Gestionar contenido informativo.	Puntos Estimados: 0.4
Fecha Inicio: 27/02/2017	Fecha Fin: 01/03/2017
Descripción: Se debe proveer una API que gestione todo el contenido informativo, así como el trabajo sobre la base de datos asegurando el correcto almacenamiento de los contenidos.	

Tabla 4.16: Tarea de Ingeniería - API de Información

Tarea de Ingeniería	
Nombre: Listar Contenido	
HU: Gestionar contenido informativo.	Puntos Estimados: 0.1
Fecha Inicio: 01/03/2017	Fecha Fin: 01/03/2017
Descripción: Un usuario puede ver una tabla con los contenidos que tiene guardados el sistema y un resumen de cada uno (nombre y estado), permitiendo gestionar (editar, mostrar, eliminar) cada uno de ellos.	

Tabla 4.16: Tarea de Ingeniería - Listar Contenido

Tarea de Ingeniería	
Nombre: Añadir Contenido	
HU: Gestionar contenido informativo.	Puntos Estimados: 0.2
Fecha Inicio: 02/03/2017	Fecha Fin: 02/03/2017
Descripción: Un usuario puede adicionar una contenido a una página.	

Tabla 4.17: Tarea de Ingeniería - Añadir Contenido

Tarea de Ingeniería	
Nombre: Mostrar Contenido	
HU: Gestionar contenido informativo.	Puntos Estimados: 0.2
Fecha Inicio: 03/03/2017	Fecha Fin: 03/03/2017
Descripción: Un usuario puede ver el contenido de su elección. Se deben cargar los archivos externos para una correcta visualización del contenido.	

Tabla 4.18: Tarea de Ingeniería - Mostrar Contenido

Tarea de Ingeniería	
Nombre: Eliminar Contenido	
HU: Gestionar contenido informativo..	Puntos Estimados: 0.1

Fecha Inicio: 06/03/2017	Fecha Fin: 06/03/2017
Descripción: Un usuario puede eliminar un contenido de su elección. Las páginas asociadas a los contenidos no serán eliminadas. Esta acción debe confirmarse debido a que es irreversible.	

Tabla 4.19: Tarea de Ingeniería - Eliminar Contenido

HU – Gestión de Usuario.

Tarea de Ingeniería	
Nombre: API Control de Usuario	
HU: Gestión de Usuario	Puntos Estimados: 0.5
Fecha Inicio: 06/03/2017	Fecha Fin: 08/03/2017
Descripción: Se debe proveer una API para gestionar los usuarios dentro del sistema y asignarle sus roles dentro del mismo.	

Tabla 4.20: Tarea de Ingeniería - API Control de Usuario

Tarea de Ingeniería	
Nombre: Crear Usuario	
HU: Gestión de Usuario.	Puntos Estimados: 0.4
Fecha Inicio: 09/03/2017	Fecha Fin: 10/03/2017
Descripción: El sistema debe crear el usuario automáticamente justo cuando este se autentique dentro de la aplicación si sus credenciales son correctas. No se le asigna ningún rol inicialmente.	

Tabla 4.21: Tarea de Ingeniería - Crear Usuario

Tarea de Ingeniería	
Nombre: Modificar Usuario	
HU: Gestión de Usuario.	Puntos Estimados: 0.4
Fecha Inicio: 13/03/2017	Fecha Fin: 14/03/2017

Descripción: Un usuario de tipo administrador puede gestionar los permisos de accesibilidad al sistema de los demás usuarios del mismo.

Tabla 4.22: Tarea de Ingeniería - Modificar Usuario

Tarea de Ingeniería	
Nombre: Eliminar Usuario	
HU: Gestión de Usuario.	Puntos Estimados: 0.2
Fecha Inicio: 15/03/2017	Fecha Fin: 15/03/2017
Descripción: Un usuario de tipo administrador puede eliminar un usuario de la lista de usuarios del sistema.	

Tabla 4.23: Tarea de Ingeniería - Eliminar Usuario

Pruebas unitarias

En esta iteración se realizaron 22 pruebas unitarias en el lado del servidor. Los resultados de su ejecución con el entorno integrado de desarrollo pueden observarse en la figura 4.2.

```

Creating test database for alias 'default'
.....
-----
Ran 22 tests in 5.845s

OK
Destroying test database for alias 'default'
    
```

Figura 4.2: Resultados de las pruebas unitarias de la segunda iteración

Pruebas de Aceptación

Para realizar las pruebas de aceptación correctamente se elaboraron dos casos de pruebas para cada una de las HU los cuales pueden ser consultados en el [Anexo 1](#). A continuación, se muestran los resultados arrojados durante la ejecución de las pruebas en conjunto.

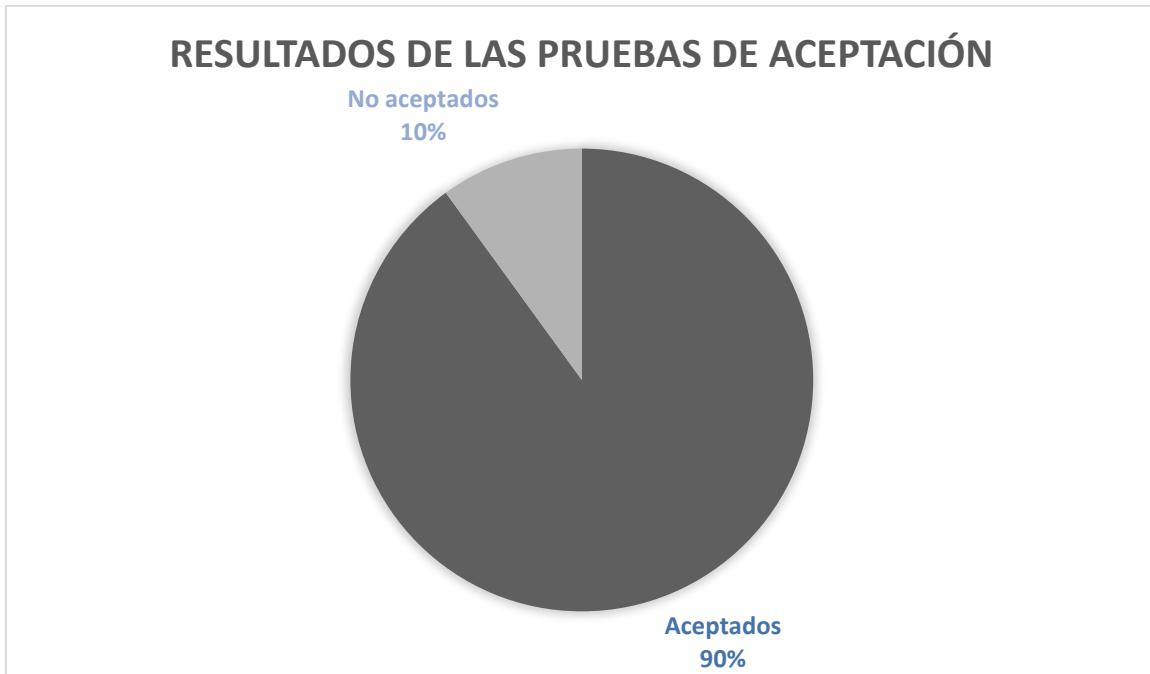


Gráfico 4.2: Resultados de las pruebas de aceptación.

4.2.3 ITERACIÓN 3

Durante esta iteración se abordó la historia de usuario de menor prioridad y se implementaron las funcionalidades asociadas al manejo de las listas de correo y envío de las mismas.

HU - Gestionar lista de correo

Tarea de Ingeniería	
Nombre: API de Correo	
HU: Gestionar lista de correo.	Puntos Estimados: 0.5
Fecha Inicio: 15/03/2017	Fecha Fin: 17/03/2017
Descripción: Se debe proveer una API que gestione todo lo relacionado a la conexión con el servicio de correo establecido para la aplicación, así como el trabajo con las listas de contactos y los contactos.	

Tabla 4.24: Tarea de Ingeniería - API de Correo.

Tarea de Ingeniería
Nombre: Listar Contactos

HU: Gestionar lista de correo	Puntos Estimados: 0.1
Fecha Inicio: 20/03/2017	Fecha Fin: 20/03/2017
Descripción: Un usuario puede ver una tabla con los contactos que tiene guardados el sistema y un resumen de cada uno (nombre y organismo), permitiendo gestionar (editar, mostrar, eliminar) cada uno de ellos.	

Tabla 4.25: Tarea de Ingeniería - Listar Contactos.

Tarea de Ingeniería	
Nombre: Añadir Contacto	
HU: Gestionar contenido informativo.	Puntos Estimados: 0.2
Fecha Inicio: 20/03/2017	Fecha Fin: 21/03/2017
Descripción: Se muestra un formulario con los campos requeridos para adicionar un contacto al sistema. Los contactos no deben repetirse	

Tabla 4.26: Tarea de Ingeniería - Añadir Contacto.

Tarea de Ingeniería	
Nombre: Modificar Contacto	
HU: Gestionar lista de correo	Puntos Estimados: 0.2
Fecha Inicio: 21/03/2017	Fecha Fin: 22/03/2017
Descripción: Un usuario puede modificar el contacto de su elección.	

Tabla 4.27: Tarea de Ingeniería - Modificar Contacto.

Tarea de Ingeniería	
Nombre: Eliminar Contacto	
HU: Gestionar lista de correo.	Puntos Estimados: 0.1
Fecha Inicio: 22/03/2017	Fecha Fin: 22/03/2017
Descripción: Un usuario puede eliminar un contacto de su elección. Esta acción debe confirmarse debido a que es irreversible.	

Tabla 4.28: Tarea de Ingeniería - Eliminar Contacto.

Tarea de Ingeniería	
Nombre: Crear Lista de Correo	
HU: Gestionar lista de correo.	Puntos Estimados: 0.1
Fecha Inicio: 23/03/2017	Fecha Fin: 23/03/2017
Descripción: Un usuario puede crear lista de correo. Debe insertar el nombre de la lista.	

Tabla 4.30: Tarea de Ingeniería - Crear Lista de Correo.

Tarea de Ingeniería	
Nombre: Modificar Lista de Correo	
HU: Gestionar lista de correo.	Puntos Estimados: 0.2
Fecha Inicio: 24/03/2017	Fecha Fin: 24/03/2017
Descripción: Un usuario puede modificar su lista de correo añadiendo o quitando contactos de la misma.	

Tabla 4.31: Tarea de Ingeniería - Modificar Lista de Correo.

Tarea de Ingeniería	
Nombre: Eliminar Lista de Correo	
HU: Gestionar lista de correo.	Puntos Estimados: 0.1
Fecha Inicio: 27/03/2017	Fecha Fin: 27/03/2017
Descripción: Un usuario puede eliminar una lista de correo de su elección. Esta acción debe confirmarse debido a que es irreversible.	

Tabla 4.32: Tarea de Ingeniería - Eliminar Lista de Correo.

Pruebas unitarias

En esta iteración se realizaron 30 pruebas unitarias en el lado del servidor. Los resultados de su ejecución con el entorno integrado de desarrollo pueden observarse en la figura 4.3.

```
Creating test database for alias 'default'
.....
-----
Ran 30 tests in 8.845s

OK
Destroying test database for alias 'default'
```

Figura 4.3: Resultados de las pruebas unitarias de la tercera iteración

Pruebas de Aceptación

Para realizar las pruebas de aceptación correctamente se elaboraron dos casos de pruebas para cada una de las HU los cuales pueden ser consultados en el Anexo 1. A continuación, se muestran los resultados arrojados durante la ejecución de las pruebas en conjunto.



Gráfico 4.3: Resultados de las pruebas de aceptación

4.3 CONCLUSIONES DEL CAPÍTULO

En este capítulo se llevaron a cabo las fases de implementación y pruebas planteadas por la metodología XP. Se elaboraron e implementaron las tareas de ingeniería que dieron solución a todas las HU logrando una mayor organización y rapidez en el desarrollo del sistema. Además se realizaron las pruebas unitarias y

de aceptación con resultados satisfactorios, cuyo objetivo fue contribuir a elevar la calidad final del producto, validando cada una de las funcionalidades para evitar que *software* con defectos llegue al cliente.

CONCLUSIONES

El desarrollo de este trabajo permitió definir las principales tecnologías y herramientas que facilitan a los usuarios el diseño y composición de un boletín informativo, así como los procesos asociados a su conformación, lo cual garantizó una base metodológica sobre los elementos que deben caracterizar el proceso de generación de un boletín. Se obtuvo un modelo guía para la implementación del sistema mediante la generación de los artefactos correspondientes a los flujos de trabajo propuestos por la metodología XP y se ha obtenido una herramienta que permite gestionar el proceso de generación del boletín en el Centro de Telemática de la Facultad dos de la Universidad de las Ciencias Informáticas, posibilitando que se cumpla satisfactoriamente el objetivo trazado para este trabajo de diploma, dando solución a la problemática planteada inicialmente por el cliente.

RECOMENDACIONES

Se recomienda continuar con el desarrollo de la aplicación propuesta, agregando nuevas funcionalidades, como, por ejemplo:

- Gestión automática de contenido de interés o relevancia a través de canales RSS.
- Módulo de gestión de plantillas para que permita la conformación de las mismas usando la tecnología *Drag and Drop*.

Se propone además desplegar la aplicación en la Universidad de las Ciencias Informáticas para comprobar su utilidad en un entorno real, así como validar su posible implementación para las demás organizaciones del país.

REFERENCIAS BIBLIOGRÁFICAS

- ADOBE, 2016. Adobe Acrobat. En: Page Version ID: 747533136, *Adobe* [en línea]. [Consulta: 10 noviembre 2016]. Disponible en: https://adobe.com/index.php?title=Adobe_Acrobat&oldid=747533136.
- ADOBE SUITE, 2016. Adobe InDesign. En: Page Version ID: 748550541, *Adobe* [en línea]. [Consulta: 10 noviembre 2016]. Disponible en: https://adobe.com/index.php?title=Adobe_InDesign&oldid=748550541.
- APACHE SOFTWARE FOUNDATION, 2017. Welcome to The Apache Software Foundation! [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <http://www.apache.org/>.
- COCKBURN, A., 2000. Selecting a project's methodology. *IEEE Software*, vol. 17, no. 4, pp. 64-71. ISSN 0740-7459. DOI 10.1109/52.854070.
- CREXENTIA, 2015. La importancia del Boletín Informativo en la web corporativa | Crexentia. [en línea]. [Consulta: 25 octubre 2016]. Disponible en: <http://www.crexentia.com/la-importancia-del-boletin-informativo-en-la-web-corporativa/>.
- DIRECCIÓN DE COMUNICACIÓN INSTITUCIONAL, 2014. *POLÍTICA DE COMUNICACIÓN INSTITUCIONAL*. mayo 2014. S.l.: s.n.
- DJANGO, 2017. The Web framework for perfectionists with deadlines | Django. [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <https://www.djangoproject.com/>.
- DJANGOREST, 2017. Django REST framework. [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <http://www.django-rest-framework.org/>.
- DON BOX, GOPAL KAKIVAYA, MICROSOFT, ANDREW LAYMAN, MICROSOFT, NOAH MENDELSON, LOTUS DEVELOPMENT CORP., HENRIK FRYSTYK NIELSEN, MICROSOFT, SATISH THATTE, MICROSOFT y DAVE WINER, USERLAND SOFTWARE, INC., 2000. Simple Object Access Protocol (SOAP) 1.1. [en línea]. [Consulta: 11 abril 2017]. Disponible en: <https://www.w3.org/TR/2000/NOTE-SOAP-20000508/>.
- GIT, 2017. Git. [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <https://git-scm.com/>.
- GUERRERO, C.A., SUÁREZ, J.M. y GUTIÉRREZ, L.E., 2013. Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. *Información tecnológica* [en línea], vol. 24, no. 3, pp. 103-114. [Consulta: 9 marzo 2017]. ISSN 0718-0764. DOI 10.4067/S0718-07642013000300012. Disponible en: http://www.scielo.cl/scielo.php?script=sci_abstract&pid=S0718-07642013000300012&lng=es&nrm=iso&tlng=es.
- GUERRERO, L.A., 2013. Modelando Interfaces para Aplicaciones Web. *Ingeniería del Software en la Década del* [en línea], pp. 227-236. [Consulta: 9 diciembre 2016]. Disponible en: <http://dcc.uchile.cl/~luguerre/papers/IS-2003.pdf>.
- IBM - Connectivity and Integration. [en línea], 2016. [Consulta: 11 abril 2017]. Disponible en: <http://www.ibm.com/middleware/us-en/solutions/connectivity-and-integration/>.
- INGENIODS, 2013. Patrones Arquitectónicos. *Ingenio DS* [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <https://ingeniods.wordpress.com/2013/09/16/patrones-arquitectonicos/>.
- JETBRAINS, 2017. PyCharm. *JetBrains* [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <https://www.jetbrains.com/pycharm/>.
- JQUERY.ORG, jQuery F.-, 2017. jQuery. [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <http://jquery.com/>.
- KAREN SALAS, 2017. Generando contenidos atractivos para su cliente. *Karen Salas Blog* [en línea]. [Consulta: 29 abril 2017]. Disponible en: <http://karensalas.com/>.
- KENT BECK, 2000. Extreme Programming Explained. *Embrace Change*,
- LARMAN CRAIG, 2001. *UML y Patrones*. 2. S.l.: s.n. ISBN 970-17-0261-1.

- LEACH, P.J., BERNERS-LEE, T., MOGUL, J.C., MASINTER, L., FIELDING, R.T. y GETTYS, J., 1999. Hypertext Transfer Protocol -- HTTP/1.1. [en línea]. [Consulta: 11 abril 2017]. Disponible en: <https://tools.ietf.org/html/rfc2616>.
- LLEDÓ, N., 2016. Recuperando información. *Nacho Lledó* [en línea]. [Consulta: 29 abril 2017]. Disponible en: <http://www.nacholledo.com/>.
- LÓPEZ TRUJILLO, M. y MARULANDA ECHEVERRY, C.E., 2012. Tecnologías Web 2.0 para la divulgación del conocimiento local. *Revista Virtual Universidad Católica del Norte* [en línea], vol. 1, no. 31, pp. 60–83. [Consulta: 26 octubre 2016]. Disponible en: <http://revistavirtual.ucn.edu.co/index.php/RevistaUCN/article/view/37>.
- MARTINEZ, B.H.O.J.J., 2014. Influencia de las TIC en la gestion de la informacion empresarial. , pp. 15.
- MICROSOFT, 2015. Crear un boletín informativo para mantener el contacto con los clientes - Publisher. [en línea]. [Consulta: 26 octubre 2016]. Disponible en: <https://support.office.com/es-es/article/Crear-un-bolet%25C3%25ADn-informativo-para-mantener-el-contacto-con-los-clientes-3431DD22-5FAB-404A-8E1F-41F5F4796E9A?ui=es-ES&rs=es-ES&ad=ES&s=1&auth=1&nf=1&fromAR=1>.
- PATRICIO LETELIER y M^a CARMEN PENADÉS PATRICIO, 2014. *Metodologías Agiles para el desarrollo de software XP.pdf*. 2014. S.l.: s.n.
- PCWORLD, 2012. Create a Website Easily With Wix (Even the Free Version). *PCWorld* [en línea]. [Consulta: 10 noviembre 2016]. Disponible en: <http://www.pcworld.com/article/253379/wix.html>.
- PEP 257 -- Docstring Conventions. *Python.org* [en línea], 2017. [Consulta: 29 abril 2017]. Disponible en: <https://www.python.org/dev/peps/pep-0257/>.
- POSTGRESQL, 2017. PostgreSQL: The world's most advanced open source database. [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <https://www.postgresql.org/>.
- PRESSMAN, R.S., 2005. *Software Engineering: A Practitioner's Approach*. S.l.: Palgrave Macmillan. ISBN 978-0-07-301933-8.
- PYTHON, 2017. Welcome to Python.org. *Python.org* [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <https://www.python.org/>.
- REPORTLAB, 2017. ReportLab - Content to PDF Solutions. [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <http://www.reportlab.com/>.
- RIVAS, M.C., 2012. Aprendizaje de las tecnologías de la información y comunicación en entornos Web 2.0. *Multiciencias* [en línea], vol. 11, no. 4. [Consulta: 26 octubre 2016]. Disponible en: <http://www.produccioncientificaluz.org/index.php/multiciencias/article/view/16876>.
- SPA, M., 2016. BeeFree the free online email editor. [en línea]. [Consulta: 10 noviembre 2016]. Disponible en: [//beefree.io](http://beefree.io).
- SPRYLAB, 2016. Puerple DS Suite. [en línea]. [Consulta: 21 febrero 2017]. Disponible en: <https://sprylab.com/en/ds/purple-products.html>.
- TLM, 2010. TLM - Centro de Telemática. [en línea]. [Consulta: 9 marzo 2017]. Disponible en: https://gespro.tlm.prod.uci.cu/login?back_url=https%3A%2F%2Fgespro.tlm.prod.uci.cu%2F.
- UCI, 2002. Misión | Portal de la Universidad de las Ciencias Informáticas. [en línea]. [Consulta: 25 octubre 2016]. Disponible en: <http://www.uci.cu/?q=mision>.
- USERLAND, 2012. XML-RPC Specification. [en línea]. [Consulta: 11 abril 2017]. Disponible en: <http://xmlrpc.scripting.com/spec.html>.
- WALLACE, D., RAGGETT, I. y AUFANG, J., 2002. *Extreme Programming for Web Projects*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. ISBN 978-0-201-79427-4.

BIBLIOGRAFÍA

- ADOBE, 2016. Adobe Acrobat. En: Page Version ID: 747533136, Adobe [en línea]. [Consulta: 10 noviembre 2016]. Disponible en: https://adobe.com/index.php?title=Adobe_Acrobat&oldid=747533136.
- ADOBE SUITE, 2016. Adobe InDesign. En: Page Version ID: 748550541, Adobe [en línea]. [Consulta: 10 noviembre 2016]. Disponible en: https://adobe.com/index.php?title=Adobe_InDesign&oldid=748550541.
- APACHE SOFTWARE FOUNDATION, 2017. Welcome to The Apache Software Foundation! [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <http://www.apache.org/>.
- ARLINE, K., 26, B.N.D.C.F. y EST, 2015 02:16 pm, [sin fecha]. What Is E-Commerce? Business News Daily [en línea]. [Consulta: 10 noviembre 2016]. Disponible en: <http://www.businessnewsdaily.com/4872-what-is-e-commerce.html>.
- BARCOS, S.J., 2008. Reflexiones Acerca De Los Sistemas De Información Universitarios Ante Los Desafíos Y Cambios Generados Por Los Procesos De Evaluación Y Acreditación. Avaliação: Revista da Avaliação da Educação Superior [en línea]. [Consulta: 26 octubre 2016]. Disponible en: <http://4www.redalyc.org/articulo.oa?id=219114876012>.
- CLARO, R.L.H. y NAVARRO, D.G., 2010. Estándares de Diseño Web. Ciencias de la Información [en línea]. [Consulta: 9 noviembre 2016]. Disponible en: <http://www.redalyc.org/articulo.oa?id=181421569009>.
- COCKBURN, A., 2000. Selecting a project's methodology. IEEE Software, vol. 17, no. 4, pp. 64-71. ISSN 0740-7459. DOI 10.1109/52.854070.
- COMMUNITY TOOL BOX, [sin fecha]. Capítulo 6. Comunicación para promover interés | Sección 10. Crear boletines | Sección Principal | Community Tool Box. [en línea]. [Consulta: 29 abril 2017]. Disponible en: <http://ctb.ku.edu/es/tabla-de-contenidos/participacion/promover-interes-en-la-comunidad/boletines/principal>.
- CREXENTIA, 2015. La importancia del Boletín Informativo en la web corporativa | Crexentia. [en línea]. [Consulta: 25 octubre 2016]. Disponible en: <http://www.crexentia.com/la-importancia-del-boletín-informativo-en-la-web-corporativa/>.
- CSS3 Introduction. [en línea], [sin fecha]. [Consulta: 9 marzo 2017]. Disponible en: https://www.w3schools.com/css/css3_intro.asp.
- DIRECCIÓN DE COMUNICACIÓN INSTITUCIONAL, 2014. POLÍTICA DE COMUNICACIÓN INSTITUCIONAL. mayo 2014. S.l.: s.n.
- DJANGO, 2017. The Web framework for perfectionists with deadlines | Django. [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <https://www.djangoproject.com/>.
- DJANGOREST, 2017. Django REST framework. [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <http://www.django-rest-framework.org/>.
- DON BOX, GOPAL KAKIVAYA, MICROSOFT, ANDREW LAYMAN, MICROSOFT, NOAH MENDELSON, LOTUS DEVELOPMENT CORP., HENRIK FRYSTYK NIELSEN, MICROSOFT, SATISH THATTE, MICROSOFT y DAVE WINER, USERLAND SOFTWARE, INC., 2000. Simple Object Access Protocol (SOAP) 1.1. [en línea]. [Consulta: 11 abril 2017]. Disponible en: <https://www.w3.org/TR/2000/NOTE-SOAP-20000508/>.
- ENCUENTRO DE PROFESORES UNIVERSITARIOS DE *MARKETING* (ed.), 2005. XVII Encuentro de Profesores Universitarios de *Marketing*: [Madrid 22 y 23 de septiembre de 2005. Madrid: ESIC. ISBN 978-84-7356-424-3.
- GIT, 2017. Git. [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <https://git-scm.com/>.
- GÓMEZ, G., 2005. Significados de la Información Web. Razón y Palabra [en línea]. [Consulta: 26 octubre 2016]. Disponible en: <http://www.redalyc.org/articulo.oa?id=199520623024>.

- GUERRA, Y.M., GONZÁLEZ, R.C. y FEBLES, J.P., 2016a. Diseño web adaptativo para la plataforma educativa ZERA. Revista Cubana de Ciencias Informáticas [en línea]. [Consulta: 26 octubre 2016]. Disponible en: <http://www.redalyc.org/articulo.oa?id=378345292008>.
- GUERRA, Y.M., GONZÁLEZ, R.C. y FEBLES, J.P., 2016b. Diseño web adaptativo para la plataforma educativa ZERA. Revista Cubana de Ciencias Informáticas [en línea], vol. 10, no. 2, pp. 100–115. [Consulta: 26 octubre 2016]. Disponible en: <http://rcci.uci.cu/index.php?journal=rcci&page=article&op=view&path%5B%5D=1048>.
- GUERRERO, C.A., SUÁREZ, J.M. y GUTIÉRREZ, L.E., 2013. Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. Información tecnológica [en línea], vol. 24, no. 3, pp. 103-114. [Consulta: 9 marzo 2017]. ISSN 0718-0764. DOI 10.4067/S0718-07642013000300012. Disponible en: http://www.scielo.cl/scielo.php?script=sci_abstract&pid=S0718-07642013000300012&lng=es&nrm=iso&tlng=es.
- GUERRERO, J., [sin fecha]. REPÚBLICA BOLIVARIANA DE VENEZUELA UNIVERSIDAD NACIONAL ABIERTA DIRECCIÓN DE INVESTIGACIÓN Y POSTGRADO. [en línea], [Consulta: 26 octubre 2016]. Disponible en: <http://biblo.una.edu.ve/docu.7/bases/marc/texto/t36148.pdf>.
- GUERRERO, L.A., 2013. Modelando Interfaces para Aplicaciones Web. Ingeniería del Software en la Década del [en línea], pp. 227–236. [Consulta: 9 diciembre 2016]. Disponible en: <http://dcc.uchile.cl/~luguerre/papers/IS-2003.pdf>.
- GUTIÉRREZ, C., 2008. Diseño web y arquitectura de información para sitios 2.0. Cuadernos de Información [en línea]. [Consulta: 9 noviembre 2016]. Disponible en: <http://www.redalyc.org/articulo.oa?id=97112294005>.
- HERNÁNDEZ-CLARO, R.L. y GREGUAS-NAVARRO, D., 2010. Estándares de Diseño Web. Ciencias de la Información, vol. 41, no. 2, pp. 69–71.
- HTML5 Introduction. [en línea], [sin fecha]. [Consulta: 9 marzo 2017]. Disponible en: https://www.w3schools.com/html/html5_intro.asp.
- IBM - Connectivity and Integration. [en línea], 2016. [Consulta: 11 abril 2017]. Disponible en: <http://www.ibm.com/middleware/us-en/solutions/connectivity-and-integration/>.
- INFORMACIÓN, C.M. sobre la S. de la, 2005. Construir sociedades de la información que atiendan a las necesidades humanas. Revista de Estudios Sociales [en línea]. [Consulta: 26 octubre 2016]. Disponible en: <http://www.redalyc.org/articulo.oa?id=81502214>.
- INGENIODS, 2013. Patrones Arquitectónicos. Ingenio DS [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <https://ingeniods.wordpress.com/2013/09/16/patrones-arquitectonicos/>.
- Introduction to HTML. [en línea], [sin fecha]. [Consulta: 13 junio 2017]. Disponible en: https://www.w3schools.com/html/html_intro.asp.
- JETBRAINS, 2017. PyCharm. JetBrains [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <https://www.jetbrains.com/pycharm/>.
- JQUERY.ORG, jQuery F.-, 2017. jQuery. [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <http://jquery.com/>.
- KAREN SALAS, 2017. Generando contenidos atractivos para su cliente. Karen Salas Blog [en línea]. [Consulta: 29 abril 2017]. Disponible en: <http://karensalas.com/>.
- KENT BECK, 2000. Extreme Programming Explained. Embrace Change,
- LARMAN CRAIG, 2001. UML y Patrones. 2. S.l.: s.n. ISBN 970-17-0261-1.
- LEACH, P.J., BERNERS-LEE, T., MOGUL, J.C., MASINTER, L., FIELDING, R.T. y GETTYS, J., 1999. Hypertext Transfer Protocol -- HTTP/1.1. [en línea]. [Consulta: 11 abril 2017]. Disponible en: <https://tools.ietf.org/html/rfc2616>.
- LLEDÓ, N., 2016. Recuperando información. Nacho Lledó [en línea]. [Consulta: 29 abril 2017]. Disponible en: <http://www.nacholledo.com/>.

- LÓPEZ TRUJILLO, M. y MARULANDA ECHEVERRY, C.E., 2012. Tecnologías Web 2.0 para la divulgación del conocimiento local. *Revista Virtual Universidad Católica del Norte* [en línea], vol. 1, no. 31, pp. 60–83. [Consulta: 26 octubre 2016]. Disponible en: <http://revistavirtual.ucn.edu.co/index.php/RevistaUCN/article/view/37>.
- MARÍA, E., 2010. Boletín electrónico. *Consumoteca - Información y consejo a los consumidores en España* [en línea]. [Consulta: 10 noviembre 2016]. Disponible en: <http://www.consumoteca.com/telecomunicaciones/internet/boletín-electronico/>.
- MARTINEZ, B.H.O.J.J., 2014. Influencia de las TIC en la gestión de la información empresarial., pp. 15.
- MICROSOFT, 2015. Crear un boletín informativo para mantener el contacto con los clientes - Publisher. [en línea]. [Consulta: 26 octubre 2016]. Disponible en: <https://support.office.com/es-es/article/Crear-un-bolet%25C3%25ADn-informativo-para-mantener-el-contacto-con-los-clientes-3431DD22-5FAB-404A-8E1F-41F5F4796E9A?ui=es-ES&rs=es-ES&ad=ES&s=1&auth=1&nf=1&fromAR=1>.
- Oxford Dictionaries - Dictionary, Thesaurus, & Grammar. [en línea], [sin fecha]. [Consulta: 13 junio 2017]. Disponible en: <https://en.oxforddictionaries.com/>.
- PABÓN, F.O.M., URIBE, G.A. y MOSQUERA, F.L., 2011. OneWeb: plataforma de adaptación de contenidos web basada en las recomendaciones del W3C Mobile Web Initiative. *Ingeniería e Investigación* [en línea], vol. 31, no. 1, pp. 117–126. [Consulta: 26 octubre 2016]. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=3666632>.
- PADILLA, A.M.G., MARTÍNEZ, R.B., APONCIO, Z.E.G. y AIZPURU, M.C., 2006. Potencial de las tecnologías data warehouse en el contexto de la empresa hotelera. El uso de las tecnologías de la información en la empresa en Canarias [en línea]. S.I.: Fundación FYDE-CajaCanarias, pp. 169–190. [Consulta: 26 octubre 2016]. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=2014381>.
- PATRICIO LETELIER y M^a CARMEN PENADÉS PATRICIO, 2014. *Metodologías Agiles para el desarrollo de software XP.pdf*. 2014. S.I.: s.n.
- PCWORLD, 2012. Create a Website Easily With Wix (Even the Free Version). *PCWorld* [en línea]. [Consulta: 10 noviembre 2016]. Disponible en: <http://www.pcworld.com/article/253379/wix.html>.
- PDF boletín - Boletines Electrónicos. [en línea], [sin fecha]. [Consulta: 10 noviembre 2016]. Disponible en: <https://sites.google.com/site/uelbosqueboletineselectronicos/pdf-boletín>.
- PEP 257 -- Docstring Conventions. *Python.org* [en línea], 2017. [Consulta: 29 abril 2017]. Disponible en: <https://www.python.org/dev/peps/pep-0257/>.
- PÉREZ SALAZAR, G., 2011. La Web 2.0 y la sociedad de la información. *Revista mexicana de ciencias políticas y sociales* [en línea], vol. 56, no. 212, pp. 57–68. [Consulta: 26 octubre 2016]. Disponible en: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0185-19182011000200004.
- P, F.O.M., G, G.A.U. y P, F.L.M., 2011. OneWeb: plataforma de adaptación de contenidos web basada en las recomendaciones del W3C Mobile Web Initiative. *Ingeniería e Investigación* [en línea]. [Consulta: 26 octubre 2016]. Disponible en: <http://www.redalyc.org/articulo.oa?id=64321170014>.
- POSTGRESQL, 2017. PostgreSQL: The world's most advanced open source database. [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <https://www.postgresql.org/>.
- PRESSMAN, R.S., 2005. *Software Engineering: A Practitioner's Approach*. S.I.: Palgrave Macmillan. ISBN 978-0-07-301933-8.
- PYTHON, 2017. Welcome to Python.org. *Python.org* [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <https://www.python.org/>.
- REPORTLAB, 2017. ReportLab - Content to PDF Solutions. [en línea]. [Consulta: 9 marzo 2017]. Disponible en: <http://www.reportlab.com/>.

- RIVAS, M.C., 2012. Aprendizaje de las tecnologías de la información y comunicación en entornos Web 2.0. *Multiciencias* [en línea], vol. 11, no. 4. [Consulta: 26 octubre 2016]. Disponible en: <http://www.produccioncientificaluz.org/index.php/multiciencias/article/view/16876>.
- ROLANDO ALFREDO HERNÁNDEZ LEÓN y SAYDA COELLO GONZÁLEZ, 2012. *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA*. 2. S.l.: s.n. ISBN 978-959- 16-1557-2.
- SALAZAR, G.P., 2011. La Web 2.0 y la sociedad de la información. *Revista Mexicana de Ciencias Políticas y Sociales* [en línea]. [Consulta: 26 octubre 2016]. Disponible en: <http://www.redalyc.org/articulo.oa?id=42119790004>.
- SÁNCHEZ-PINILLA, M.D., 2003. Las Tecnologías De La Información Y La Comunicación: Sus Opciones, Sus Limitaciones Y Sus Efectos En La Enseñanza. *Nómadas* [en línea]. [Consulta: 26 octubre 2016]. Disponible en: <http://4www.redalyc.org/articulo.oa?id=18100809>.
- SelectingDevelopmentApproach.pdf [en línea], [sin fecha]. S.l.: s.n. [Consulta: 13 junio 2017]. Disponible en: <https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>.
- SPA, M., 2016. BeeFree the free online email editor. [en línea]. [Consulta: 10 noviembre 2016]. Disponible en: [//beefree.io](http://beefree.io).
- SPRYLAB, 2016. Puerple DS Suite. [en línea]. [Consulta: 21 febrero 2017]. Disponible en: <https://sprylab.com/en/ds/purple-products.html>.
- TLM, 2010. TLM - Centro de Telemática. [en línea]. [Consulta: 9 marzo 2017]. Disponible en: https://gespro.tlm.prod.uci.cu/login?back_url=https%3A%2F%2Fgespro.tlm.prod.uci.cu%2F.
- UCI, 2002. Misión | Portal de la Universidad de las Ciencias Informáticas. [en línea]. [Consulta: 25 octubre 2016]. Disponible en: <http://www.uci.cu/?q=mision>.
- USERLAND, 2012. XML-RPC Specification. [en línea]. [Consulta: 11 abril 2017]. Disponible en: <http://xmlrpc.scripting.com/spec.html>.
- WALLACE, D., RAGGETT, I. y AUFANG, J., 2002. *Extreme Programming for Web Projects*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. ISBN 978-0-201-79427-4.
- What is e-commerce (electronic commerce or EC)? - Definition from WhatIs.com. SearchCIO [en línea], [sin fecha]. [Consulta: 10 noviembre 2016]. Disponible en: <http://searchcio.techtarget.com/definition/e-commerce>.
- WIX.COM, 2016. Free Website Builder | Create a Free Website. Wix.com [en línea]. [Consulta: 13 junio 2017]. Disponible en: <https://www.wix.com>.