

REPOSITORIO ACADÉMICO DIGITAL INSTITUCIONAL

Desarrollo de aplicación web para la publicación de boletines corporativos de Cinepolis.

Autor: Adrián de Jesús Piedra Goicochea

**Tesis presentada para obtener el título de:
Ingeniero en Sistemas Computacionales**

**Nombre del asesor:
Aldo Israel Sandoval Monroy**

Este documento está disponible para su consulta en el Repositorio Académico Digital Institucional de la Universidad Vasco de Quiroga, cuyo objetivo es integrar, organizar, almacenar, preservar y difundir en formato digital la producción intelectual resultante de la actividad académica, científica e investigadora de los diferentes campus de la universidad, para beneficio de la comunidad universitaria.

Esta iniciativa está a cargo del Centro de Información y Documentación "Dr. Silvio Zavala" que lleva adelante las tareas de gestión y coordinación para la concreción de los objetivos planteados.

Esta Tesis se publica bajo licencia Creative Commons de tipo "Reconocimiento-NoComercial-SinObraDerivada", se permite su consulta siempre y cuando se mantenga el reconocimiento de sus autores, no se haga uso comercial de las obras derivadas.





**UNIVERSIDAD
VASCO DE QUIROGA**

FACULTAD DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES

“Desarrollo de Aplicación Web para la Publicación de
Boletines Corporativos de Cinepolis”

Tesina Propuesta para la Obtención del Título de Carrera:
INGENIERO EN SISTEMAS COMPUTACIONALES

Presenta:

Adrian de Jesus Piedra Goicochea

Asesor:

Ing. Aldo Israel Sandoval Monroy

CLAVE: 16PSU0049F

ACUERDO: LIC100846

Septiembre 2014

INDICE GENERAL

Abstract	4
Antecedentes	5
Planteamiento del Problema	7
Objetivo	8
Justificación	9
CAPITULO 1 Introducción	10
CAPITULO 2 Marco Teórico	12
2.1 Cloud Computing	12
Definición.....	12
Beneficios.....	12
Software como Servicio.....	14
2.2 MVC Modelo Vista Controlador	16
Definición.....	16
Ventajas.....	18
2.3 Ajax	18
Definición.....	18
Desventajas	19
2.4 Frameworks	20
Definición.....	20
Tipos de Frameworks Web.....	21
Características.....	21
2.5 Metodología de Desarrollo	22
¿Qué es una metodología ágil?	24
¿Por qué surgen las Metodologías Ágiles?	24
Costos de los Cambios en la Construcción de Software.....	24
Comparativa de la Metodología Ágil.....	25
Metodología Ágil.....	25
Metodología No Ágil (Tradicional)	25
Scrum.....	26
Mockup	31
2.6 Bases de Datos	31
Definición.....	31
¿Por qué utilizar una base de datos?	32
Administración de la Base de Datos.....	32
Base de datos Relacional.....	33
Diseño de las Bases de Datos Relacionales.....	34
CAPITULO 3 Desarrollo	36
3.1 Estructura de la aplicación	36
3.2 Historias de Usuario	36
3.3 Mockup's del desarrollo de la aplicación	43
3.4 Calendarización de los módulos para el desarrollo de la aplicación	45
3.5 Desarrollo de la Base de Datos	46
3.6 Desarrollo del Software	51
3.6.1 Login	51
3.6.2 Edición.....	52
Conclusión	72

Bibliografía.....	73
Índice Figuras	74
Índice Tablas.....	76
Glosario de Términos	77

Abstract

The production of newsletters within companies is a form to keep the communication with your employees and customers of the activities and decisions that have been taken within the corporate.

Many of these newsletters are broadcast nationally and even internationally; this depends on the scope and expansion of each company. The Cinepolis corporate, one of the biggest companies film projection in Mexico, asked to Fan Creative Solutions the design and the publishing of newsletters.

That is why graphic Design Company turned to Kaizen Innova, those involved in the development of software, this with the purpose that they develop a web application which can replace a process which takes a long time to take to out, which is the online publication of newsletters designed by Fan Solutions.

The Web application was developed in .NET technology and publishes newsletters on the Internet given the designs at a maximum of 15 minutes depending of the number of the images and links that contain the newsletter.

Antecedentes

La empresa Fan Soluciones Creativas se dedica al diseño gráfico, la cual tiene como cliente potencial a la empresa Cinépolis. Fan Soluciones Creativas, es una agencia de publicidad fundada en junio del 2006. Nace con la finalidad de prestar un servicio integral en mercadotecnia y publicidad a empresas interesadas en dar a conocer sus productos y servicios, a nivel local o nacional. Fan Soluciones Creativas es, actualmente, de las pocas empresas de la ciudad que presta este servicio de manera integral, ya que ofrece a sus clientes un servicio completo, involucrando todo aquello que requiera la publicidad: materiales, carteles, radio, televisión e Internet.

De esta forma optimizan la inversión en publicidad, reduciendo los costos operativos y creando una experiencia emocional entre la marca y el cliente. Dentro de los desarrollos que la empresa Fan Soluciones Creativas ha creado para este cliente, se encuentra el diseño y creación de ilustraciones publicitarias, las cuales son colocadas en medios impresos como posters y espectaculares, además se han creado personajes animados, empleando la multimedia como herramienta para la generación de contenidos en las campañas publicitarias del propio cliente.

Cinépolis solicitó a Fan Soluciones Creativas realizar la generación de un boletín electrónico empresarial, el cual se transmitiera a nivel Corporativo, Nacional, India, Brasil y EU.

Anteriormente Fan Soluciones Creativas no contaba con alguna interfaz que le ayudara en el trabajo electrónico, hablando principalmente de Internet. La empresa realizaba los diseños por cada boletín, así como las subdivisiones, una vez creados los diseños y las secciones era necesario crear el código HTML con las coordenadas de botones para mostrar las secciones y subir las imágenes con los diseños al servidor.

Estas tareas consumían una gran cantidad de tiempo al realizar los boletines a prueba y error. El tiempo promedio de realización de un solo boletín era de aproximadamente de 60 a 90 minutos, se debía probar cada una de las secciones y subir las imágenes al servidor, lo que hacía que el procedimiento fuera complejo, además de que tenían que realizar el envío masivo de correo electrónico adjuntando el contenido al mismo, el diseño del boletín con los accesos a las diferentes secciones tenían que ser probados.

Las imágenes resultantes de los diseños no estaban organizadas; éstas las tenían guardadas en carpetas, sin embargo no las tenían relacionadas con los boletines a los cuales pertenecían, esto creaba muchos archivos y diseños que al comunicarse entre diseñadores generaban conflicto, ya que cambios en los diseños pueden haber sido solo por color, ciertas letras y detalles relativamente pequeños con los cuales es difícil identificar que versión se manejaba.

Planteamiento del Problema

De acuerdo a los antecedentes descritos, se realiza el siguiente planteamiento del problema.

- ¿Se puede reducir el tiempo de publicación de un boletín? De acuerdo al promedio de publicación del boletín se habla de aproximadamente de 60 a 90 minutos ya teniendo las ilustraciones desarrolladas.
- ¿Se puede generar una herramienta la cual pueda reducir el tiempo de generación de código? Aun teniendo las ilustraciones lo que ocasiona mucha pérdida de tiempo es el hecho de insertar coordenadas a prueba y error, teniendo en cuenta que los botones pueden tener una localización diferente con cada sección.
- ¿Cómo se mantienen organizadas las imágenes de los boletines? Las ilustraciones desarrolladas para cada boletín son numeradas y nombradas por cada encargado de la sección, es decir, no se tiene un control sobre los archivos en el ftp tanto en nombre y numeración.
- ¿Se podría simplificar el proceso de generación de boletines para los usuarios, a través del uso de la tecnología Web?

Objetivo

Se busca simplificar las operaciones que se tienen en la actualidad para la publicación rápida, con diseño y de manera electrónica un boletín corporativo mediante un dominio público o privado, elaborando tales boletines sobre un sistema web que permita la elaboración de proyectos en línea.

Como objetivos más específicos de acuerdo a los requerimientos se tienen los siguientes:

- Simplificar el proceso de generación de boletines.
- Desarrollar una aplicación la cual reduzca el tiempo de generación de código para la publicación de boletines.
- Desarrollar la aplicación con la funcionalidad de enviar correo con el contenido del boletín.
- Permitir que la aplicación organice las imágenes de acuerdo al número y sección de boletín.

Justificación

El proyecto beneficiará directamente a los diseñadores gráficos ya que no se involucrarán con código HTML para publicar el boletín, esto es uno de los objetivos principales ya que es uno de los procesos en la publicación que les lleva más tiempo desempeñar, además se necesitará enviar un correo electrónico con el contenido del mismo boletín.

El sistema estará diseñado para que lo puedan usar los administradores de proyectos y dar pronta respuesta al cliente de una manera más rápida y así tener una mejor calidad de servicio a sus clientes. Esto conlleva a que los diseñadores puedan emplear más tiempo en el diseño y no en códigos de programación, con ello los diseñadores ya no tendrán la necesidad de tener conocimiento en el área de programación ya que el sistema web realizará esa tarea de forma transparente para los diseñadores.

El tiempo de publicación se reducirá en más de la mitad del tiempo que se lleva actualmente y esto beneficia de manera directa también al cliente que solicite los boletines informativos. La plataforma de desarrollo será web usando tecnología .NET en conjunto con AJAX y JavaScript, esto permitirá que el trabajo se desarrolle en un entorno WEB y que las limitantes en cuanto a recursos de hardware en equipo de computo sean pocas ya que con solo el acceso a un explorador web el sistema podrá funcionar y el usuario podrá tener acceso a todas las funciones de la aplicación.

CAPITULO 1 Introducción

Un boletín informativo es una publicación distribuida de forma regular, generalmente centrada en un tema principal que es del interés de sus suscriptores. Muchos boletines son publicados por clubes, sociedades, asociaciones y negocios, particularmente compañías, para proporcionar información de interés a sus miembros o empleados. Algunos boletines informativos son creados con ánimo de lucro y se venden directamente a sus suscriptores, otros no.

Muchas empresas corporativas realizan diferentes actividades, eventos, asociaciones, inclusive quieren publicar datos estadísticos para darlos a conocer a sus clientes y a sus propios empleados. Hacerlos parte de la empresa y publicar los eventos que se realizan dentro del corporativo en boletines informativos es en cierta forma una manera de dar gratificación a sus empleados y a miembros de la empresa.

Algunas compañías distribuyen de manera masiva boletines informativos electrónicos, que suelen ser enviados a los suscriptores por medio de correo electrónico, a veces es común que solo se envíe dentro del correo electrónico únicamente el link para acceder al boletín; otras maneras es insertar directamente el boletín informativo en el mail, pero a veces eso hace que el correo sea demasiado pesado si es que no se tiene una buena administración del espacio y del peso de las imágenes que lleva el contenido del correo electrónico.

Las empresas que diseñan boletines tiene muchas maneras de distribuirlos, una de ellas es por medio de correos masivos como ya se ha comentado y otra de ellas es por medio de alguna página web, inclusive los boletines pueden ser leídos dentro de la página de la empresa que distribuye los boletines. El tiempo que invierten los diseñadores en la publicación de un boletín debe ser 100% en diseño, dejando de lado la parte técnica, la cual contempla la publicación en la web por medio de la programación en HTML, Flash, creación de correo electrónico con código HTML integrado, etc.

Ante este inconveniente y a la falta de un programador o de alguien con conocimiento técnico en programación HTML, se realizó la creación de una aplicación Web. En esta tesina se encontrará la manera en la que se resolvió esta problemática, ya que la empresa Fan Soluciones tenía la necesidad de publicar boletines electrónicos de la empresa Cinepolis dentro de su propio portal. Se realizaron medidas dentro de la empresa de diseño gráfico, sobre todo en tiempo, esto para ver la optimización que tuvo el proceso que se seguía antes y la que tiene ahora, mostrando la mejoría en el tiempo de publicación de boletines electrónicos informativos.

CAPITULO 2 Marco Teórico

2.1 Cloud Computing

Definición

Se habla mucho estos días del término: Que nuestros datos y relaciones no estén en nuestros equipos ni dependan del sistema operativo de nuestros equipos, sino en la red [1].

Usted puede haber oído el término cloud computing o "la Nube", pero ¿podría describir qué es? Hay tantas definiciones. La computación en la nube es simplemente un conjunto de recursos informáticos agrupados y servicios prestados a través de la web. El llamado cloud computing o computación de nube (en estricta traducción) es el término que se le da a la tendencia de basar las aplicaciones en servicios alojados de forma externa, en la propia web.

Dicho de otra forma, los datos y relaciones en este esquema no están en nuestros equipos ni dependen del sistema operativo de los mismos, sino en la red [1].

El Cloud computing permite al usuario acceder a un catálogo de servicios estandarizados y responder a las necesidades de su negocio, de forma flexible y adaptativa, en caso de demandas no previsibles o de picos de trabajo, pagando únicamente por el consumo efectuado.

Este concepto incorpora el software como servicio, como en la Web 2.0 y otras tendencias tecnológicas recientes, que tienen en común el que confían en Internet para satisfacer las necesidades de cómputo de los usuarios. [1]

Beneficios

- Integración probada de servicios Web. Por su naturaleza, la tecnología de Cloud Computing se puede integrar con mucha mayor facilidad y rapidez con el resto de sus aplicaciones empresariales (tanto software tradicional como Cloud Computing basado en infraestructuras), ya sean desarrolladas de manera interna o externa.

- Prestación de servicios a nivel mundial. Las infraestructuras de Cloud Computing proporcionan mayor capacidad de adaptación, recuperación de desastres completa y reducción al mínimo de los tiempos de inactividad.
- Una infraestructura 100% de Cloud Computing no necesita instalar ningún tipo de hardware. La belleza de la tecnología de Cloud Computing es su simplicidad y el hecho de que requiera mucha menor inversión para empezar a trabajar.
- Implementación más rápida y con menos riesgos. Podrá empezar a trabajar muy rápidamente gracias a una infraestructura de Cloud Computing. No tendrá que volver a esperar meses o años e invertir grandes cantidades de dinero antes de que un usuario inicie sesión en su nueva solución. Sus aplicaciones en tecnología de Cloud Computing estarán disponibles en cuestión de semanas o meses, incluso con un nivel considerable de personalización o integración.
- Actualizaciones automáticas que no afectan negativamente a los recursos de TI. Si se actualiza a la última versión de la aplicación, nos veremos obligados a dedicar tiempo y recursos (que no se tienen) a volver a crear las personalizaciones e integraciones. La tecnología de Cloud Computing no le obliga a decidir entre actualizar y conservar su trabajo, porque esas personalizaciones e integraciones se conservan automáticamente durante la actualización.
- Contribuye al uso eficiente de la energía. En este caso, a la energía requerida para el funcionamiento de la infraestructura. En los datacenters tradicionales, los servidores consumen mucha más energía de la requerida realmente. En cambio, en las nubes, la energía consumida es sólo la necesaria, reduciendo notablemente el desperdicio. [2]

Software como Servicio

El término “Software + Services” fue concebido por Microsoft con el objetivo de ponerle un nombre a la estrategia relacionada con Software como Servicio. La categoría software + services aporta otras características que no son parte del modelo base, pero que agregan valor al modelo en general. La siguiente definición, extraída del sitio de Microsoft, resume el concepto:

“Software-plus-Services is an industry shift - combining Internet services with client and server software to deliver more compelling opportunities and solutions... Microsoft helps its customers and partners capitalize on the opportunities presented by the evolution to Software-plus-Services. Microsoft’s platform uniquely supports all classes of the services manifestation — services delivery (SaaS), services composition (SOA) or the services experience (Web 2.0) — across a range of devices in addition to the PC, the various types of browser clients, servers in data-centers and Internet services.” [Microsoft Corp., 2008]

La combinación de servicios de internet con software cliente y de servidor como indica la definición y el acceso a través de diferentes dispositivos, no solamente la PC es la visión de Microsoft. En resumen, no solamente la experiencia web es importante, sino también la experiencia con dispositivos, la experiencia del escritorio y la experiencia sin conexión.

El siguiente diagrama mostrado en la Figura 2.1 sintetiza el análisis realizado de las características relacionadas con software brindado como servicio.

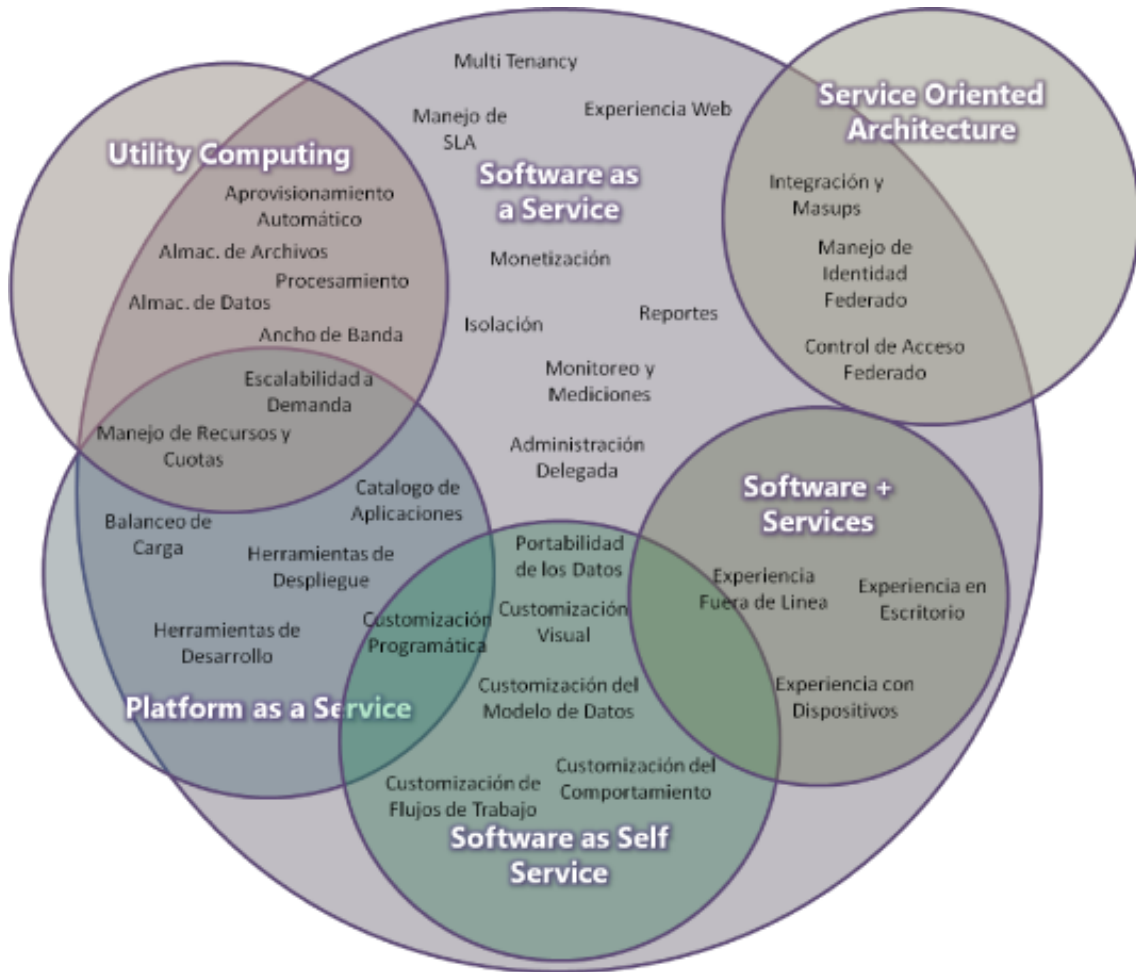


Figura 2.1: ANÁLISIS DEL DOMINIO DE SOFTWARE COMO SERVICIO.

2.2 MVC Modelo Vista Controlador

Definición

El modelo de arquitectura MVC¹ separa una aplicación en tres componentes principales: el modelo, la vista y el controlador. El marco de ASP.NET MVC proporciona una alternativa al modelo de formularios Web Forms de ASP.NET para crear aplicaciones web. El marco de ASP.NET MVC es un marco de presentación de poca complejidad y fácil de comprobar que (como las aplicaciones basadas en formularios Web Forms) se integra con las características de ASP.NET existentes, como páginas maestras y la autenticación basada en pertenencia. El marco de MVC se define en el ensamblado System.Web.Mvc.

MVC es un modelo de diseño estándar con el que están familiarizados muchos desarrolladores. Algunos tipos de aplicaciones web salen beneficiadas con el marco de MVC. Otras seguirán utilizando el modelo de la aplicación ASP.NET tradicional que está basado en formularios Web Forms y postbacks. Otros tipos de aplicaciones web combinarán las dos estrategias; una no excluye a la otra.

[3]

El marco de MVC incluye los componentes siguientes:

- **Modelos.** Los objetos de modelo son las partes de la aplicación que implementan la lógica del dominio de datos de la aplicación. A menudo, los objetos de modelo recuperan y almacenan el estado del modelo en una base de datos. Por ejemplo, un objeto Product podría recuperar información de una base de datos, trabajar con ella y, a continuación, escribir la información actualizada en una tabla Productos de una base de datos de SQL Server. En las aplicaciones pequeñas, el modelo es a menudo una separación conceptual en lugar de física. Por ejemplo, si la aplicación solo lee un conjunto de datos y lo envía a la vista, la

¹ Model-View-Controller

aplicación no tiene un nivel de modelo físico y las clases asociadas. En ese caso, el conjunto de datos asume el rol de un objeto de modelo.

- **Vistas.** Las vistas son los componentes que muestra la interfaz de usuario de la aplicación. Normalmente, esta interfaz de usuario se crea a partir de los datos de modelo. Un ejemplo sería una vista de edición de una tabla Productos que muestra cuadros de texto, listas desplegables y casillas basándose en el estado actual de un objeto Product.
- **Controladores.** Los controladores son los componentes que controlan la interacción del usuario, trabajan con el modelo y por último seleccionan una vista para representar la interfaz de usuario. En una aplicación MVC, la vista solo muestra información; el controlador administra y responde a los datos proporcionados por el usuario y su interacción.

Por ejemplo, el controlador administra los valores de la cadena de consulta y pasa estos valores al modelo, que a su vez podría utilizarlos para consultar la base de datos.

El modelo de MVC le ayuda a crear aplicaciones que separan los aspectos diferentes de la aplicación (lógica de entrada, lógica comercial y lógica de la interfaz de usuario), proporcionando un vago acoplamiento entre estos elementos. El modelo especifica dónde se debería encontrar cada tipo de lógica en la aplicación. La lógica de la interfaz de usuario pertenece a la vista. La lógica de entrada pertenece al controlador. La lógica comercial pertenece al modelo. Esta separación le ayuda a administrar la complejidad al compilar una aplicación, ya que le permite centrarse en cada momento en un único aspecto de la implementación. Por ejemplo, se puede centrar en la vista sin estar condicionado por la lógica comercial.

El acoplamiento vago entre los tres componentes principales de una aplicación MVC también favorece el desarrollo paralelo. Por ejemplo, un desarrollador de software puede trabajar en la vista, un segundo desarrollador puede ocuparse

de la lógica del controlador y un tercero se puede centrar en la lógica comercial del modelo. [3]

Ventajas

El marco de ASP.NET MVC ofrece las ventajas siguientes:

- Facilita la administración de la complejidad, al dividir una aplicación en el modelo, la vista y el controlador.
- No utiliza el estado de vista ni formularios basados en servidor. Esto hace el marco de MVC sea ideal para los desarrolladores que deseen un control completo sobre el comportamiento de una aplicación.
- Utiliza un modelo Controlador frontal que procesa las solicitudes de la aplicación web a través de un controlador único. Esto permite diseñar una aplicación que admite una infraestructura de enrutamiento avanzada. Para obtener más información, vea Front Controller.
- Proporciona una mayor compatibilidad con el desarrollo basado en pruebas (TDD).
- Funciona bien para las aplicaciones web en las que trabajan equipos grandes de desarrolladores y para los diseñadores web que necesitan un alto grado de control sobre el comportamiento de la aplicación. [3]

2.3 Ajax

Definición

AJAX son las siglas de Asynchronous JavaScript And XML. No es un lenguaje de programación sino un conjunto de tecnologías que nos permiten hacer páginas de internet más interactivas. La característica fundamental de AJAX es permitir actualizar parte de una página con información que se encuentra en el servidor sin tener que refrescar completamente la página. De modo similar podemos enviar información al servidor. [4]

Ventajas

- Utiliza tecnologías ya existentes.
- Soportada por la mayoría de los navegadores modernos.
- Interactividad. El usuario no tiene que esperar hasta que lleguen los datos del servidor.
- Portabilidad (no requiere plug-in como Flash y Applet de Java)
- Mayor velocidad, esto debido que no hay que retornar toda la página nuevamente.
- La página se asemeja a una aplicación de escritorio.

Desventajas

- Se pierde el concepto de volver a la página anterior.
- Si se guarda en favoritos no necesariamente al visitar nuevamente el sitio se ubique donde nos encontrábamos al grabarla.
- La existencia de páginas con AJAX y otras sin esta tecnología hace que el usuario se desoriente.
- Problemas con navegadores antiguos que no implementan esta tecnología.
- No funciona si el usuario tiene desactivado el JavaScript en su navegador.
- Requiere programadores que conozcan todas las tecnologías que intervienen en AJAX.
- Dependiendo de la carga del servidor podemos experimentar tiempos tardíos de respuesta que desconciertan al visitante.

Por medio de la Figura 2.2 podemos observar el modo en el que trabaja la tecnología Ajax, en la parte izquierda se tiene la petición de la parte cliente, la cual puede ser representada por una petición de algún listado de registros a una base de datos que se encuentra en la parte del servidor la cual está del lado derecho.

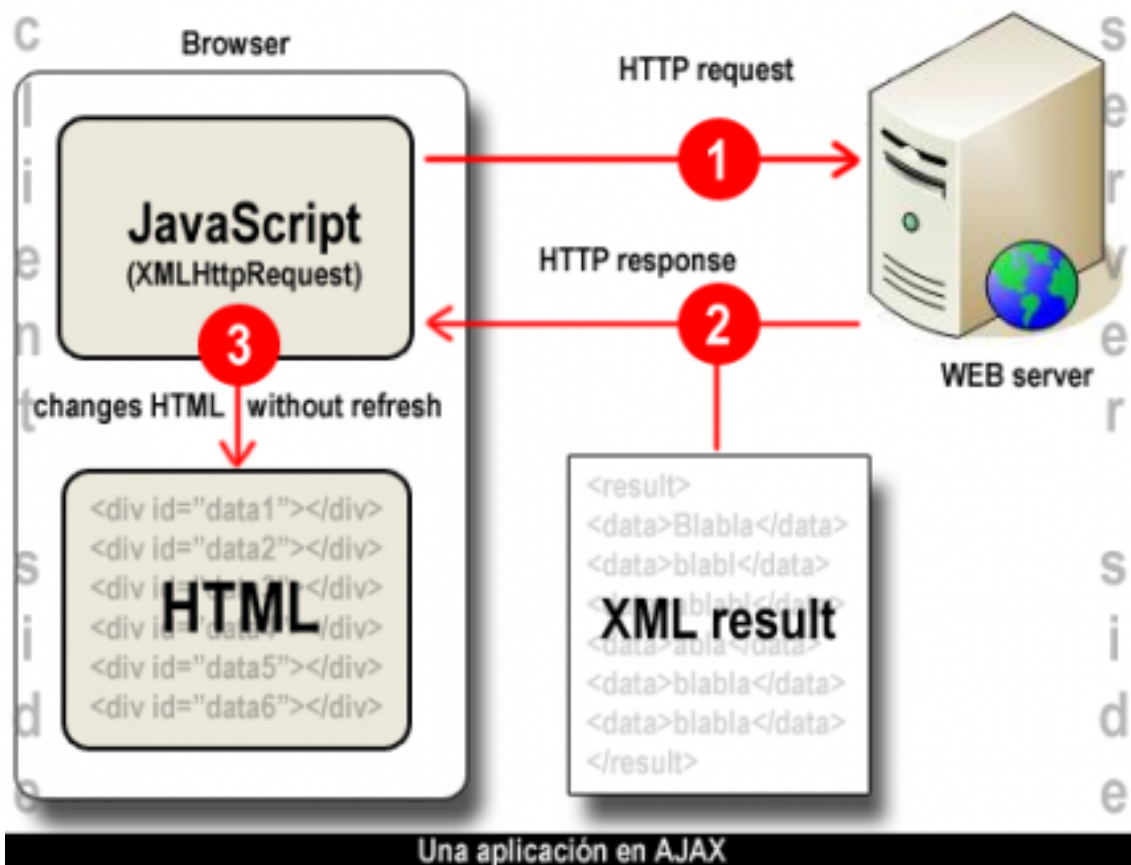


Figura 2.2: DIAGRAMA PETICIÓN AJAX

2.4 Frameworks

Definición

El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito que pueda ser empleado.

En general, el término framework, se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Un framework Web, por tanto, se puede definir como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

Tipos de Frameworks Web

Existen varios tipos de frameworks Web: orientados a la interfaz de usuario, como Java Server Faces, orientados a aplicaciones de publicación de documentos, como Coocon, orientados a la parte de control de eventos, como Struts y algunos que incluyen varios elementos como Tapestry.

La mayoría de frameworks Web se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC o con el modelo 2 de Servlets y JSP, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación.

Características

A continuación, en la Tabla 2.1 enunciamos una serie de características que podemos encontrar en prácticamente todos los frameworks existentes.

Tabla 2.1 Características En Los Frameworks

Abstracción de URLs y sesiones.	No es necesario manipular directamente las URLs ni las sesiones, el framework ya se encarga de hacerlo.
Acceso a datos.	Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en BBDD, XML, etc.
Controladores.	La mayoría de frameworks implementa una

	serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.
Autenticación y control de acceso	Incluyen mecanismos para la identificación de usuarios mediante login y password y permiten restringir el acceso a determinadas páginas a determinados usuarios.
Internacionalización.	
Separación entre diseño y contenido.	

2.5 Metodología de Desarrollo

Con mucho, el desarrollo de software es una actividad caótica, frecuentemente caracterizada por la frase "codifica y corrige". El software se escribe con un plan subyacente mínimo, y el diseño del sistema se adoquina con muchas decisiones a corto plazo. Esto realmente funciona muy bien si el sistema es pequeño, pero conforme el sistema crece llega a ser cada vez más difícil agregar nuevos aspectos al mismo. Además los bugs llegan a ser cada vez más frecuentes y más difíciles de corregir. La seña típica de tal sistema es una larga fase de pruebas después de que el sistema ha sido "completado". Tal fase larga de pruebas hace estragos con los planes de pruebas y depurado llegando a ser imposible de incluir en el programa de trabajo.

Se ha trabajado con este estilo de desarrollo por mucho tiempo, pero también hemos tenido una alternativa desde hace mucho: Metodología. Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería.

Las metodologías ingenieriles han estado presentes durante mucho tiempo. No se han distinguido precisamente por ser muy exitosas. Aún menos por su popularidad. La crítica más frecuente a estas metodologías es que son burocráticas. Hay tanto que hacer para seguir la metodología que el ritmo entero del desarrollo se retarda.

Como una reacción a estas metodologías, un nuevo grupo de metodologías ha surgido en los últimos años. Durante algún tiempo se conocían como metodologías ligeras, pero el término aceptado ahora es metodologías ágiles. Para mucha gente el encanto de estas metodologías ágiles es su reacción ante la burocracia de las metodologías monumentales. Estos nuevos métodos buscan un justo medio entre ningún proceso y demasiado proceso, proporcionando simplemente suficiente proceso para que el esfuerzo valga la pena.

El resultado de todo esto es que los métodos ágiles cambian significativamente algunos de los énfasis de los métodos ingenieriles. La diferencia inmediata es que son menos orientados al documento, exigiendo una cantidad más pequeña de documentación para una tarea dada. De muchas maneras son más bien orientados al código: siguiendo un camino que dice que la parte importante de la documentación es el código fuente.

Sin embargo, éste no es el punto importante sobre los métodos ágiles. La falta de documentación es un síntoma de diferencias mucho más profundas:

- Los métodos ágiles son adaptables en lugar de predictivos. Los métodos ingenieriles tienden a intentar planear una parte grande del proceso del software en gran detalle para un plazo largo de tiempo, esto funciona bien hasta que las cosas cambian. Así que su naturaleza es resistirse al cambio. Para los métodos ágiles, no obstante, el cambio es bienvenido. Intentan ser procesos que se adaptan y crecen en el cambio, incluso al punto de cambiarse ellos mismos.
- Los métodos ágiles son orientados a la gente y no orientados al proceso. La meta de los métodos ingenieriles es definir un proceso que funcionará bien con cualquiera que lo use. Los métodos ágiles afirman

que ningún proceso podrá nunca maquillar las habilidades del equipo de desarrollo, de modo que el papel del proceso es apoyar al equipo de desarrollo en su trabajo. Explícitamente puntualizan el trabajar a favor de la naturaleza humana en lugar de en su contra y enfatizan que el desarrollo de software debe ser una actividad agradable.

¿Qué es una metodología ágil?

Las Metodologías Ágiles o “ligeras” constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de e-projects que las metodologías convencionales (ISO-9000, CMM, etc.) debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

¿Por qué surgen las Metodologías Ágiles?

Se destacan los siguientes puntos como los principales motivos de surgimiento de las Metodologías Ágiles:

1. Dificultad para implantar metodologías tradicionales. Sofisticadas herramientas CASE y notaciones (UML)
2. Una solución a medida para un segmento importante de proyectos de desarrollo de software
3. Pugna entre comunidades / gurúes
4. Aceptar el cambio

Costos de los Cambios en la Construcción de Software

En la Figura 2.3 se muestra el costo de producir cambios en el software que se desarrolla mediante una metodología tradicional versus el costo de producir cambios en el software que se desarrolla mediante alguna de las metodologías ágiles.

Como se puede apreciar, a medida que avanza el tiempo, el costo es exponencial en el caso de la construcción mediante una metodología tradicional.

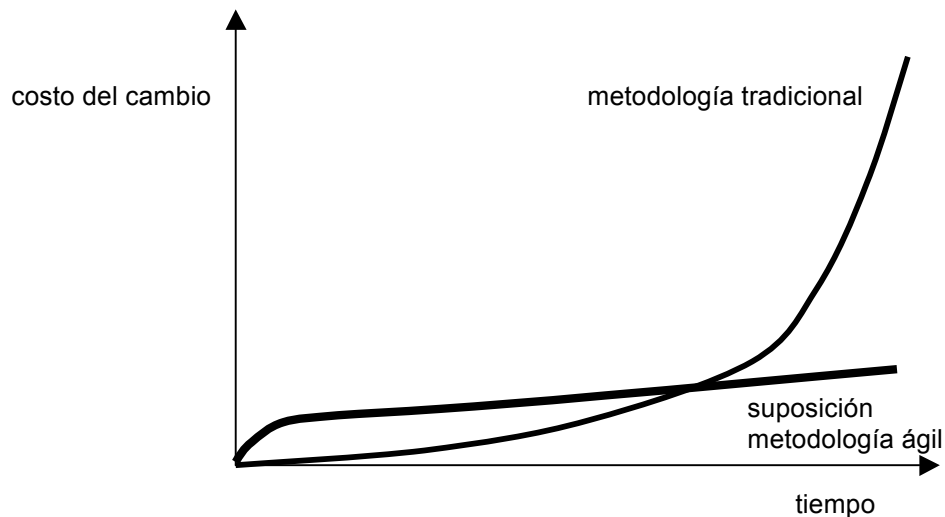


Figura 2.1 Características En Los Frameworks

Comparativa de la Metodología Ágil

La Tabla 2.2 muestra la comparativa entre los recursos y situaciones en las que se utiliza un Método Ágil contra las mismas características cuando no se ocupa un Método Ágil.

Tabla 2.2 Tabla Comparativa de Metodología Ágil

Metodología Ágil	Metodología No Ágil (Tradicional)
Pocos artefactos	Más artefactos
Pocos roles	Más roles
No existe un contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo (además in-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (< 10 integrantes) y trabajando en el mismo sitio	Grupos grandes
Menos énfasis en la arquitectura	La arquitectura es esencial

Scrum

Scrum se enfoca en el hecho de que procesos definidos y repetibles sólo funcionan para atacar problemas definidos y repetibles con gente definida y repetible en ambientes definidos y repetibles.

Scrum divide un proyecto en iteraciones (que ellos llaman carreras cortas) de 30 días. Antes de que comience una carrera se define la funcionalidad requerida para esa carrera y entonces se deja al equipo para que la entregue. El punto es estabilizar los requisitos durante la carrera.

Sin embargo la gerencia no se desentiende durante la carrera corta. Todos los días el equipo sostiene una reunión corta (quince minutos), llamada scrum, dónde el equipo discurre lo que hará al día siguiente. En particular muestran a los bloques de la gerencia: los impedimentos para progresar que se atraviesan y que la gerencia debe resolver. También informan lo que se ha hecho para que la gerencia tenga una actualización diaria de dónde va el proyecto.

La literatura de Scrum se enfoca principalmente en la planeación iterativa y el seguimiento del proceso. Es muy cercana a las otras metodologías ágiles en muchos aspectos y debe funcionar bien con las prácticas de código de XP.

Sprint

Un Sprint en Scrum es el término que denomina a una iteración que está acotada en el tiempo, usualmente entre 2 y 4 semanas, durante la cual el Equipo trabaja para convertir items del Backlog Del Producto en un Incremento Del Producto potencialmente productivo.

La duración del Sprint debería ser lo suficientemente larga para crear algo de valor y con la suficiente calidad como para ser demostrado frente al dueño del producto y los stakeholders. Con un plazo superior a 4 semanas el Equipo pierde agilidad, ya que tendrá más artefactos y documentación por la cual preocuparse.

Un Sprint de 2 semanas es la duración ideal, y se convirtió en un estándar defecto por los siguientes motivos:

- Fuerza al equipo a operar de una manera ágil, en vez de hacer "mini cascadas"
- Brinda un feedback más frecuente en lo que se está construyendo
- Reduce el riesgo de "construir la cosa equivocada"
- Incrementa la respuesta del equipo ante cambios en el negocio
- Le brinda al equipo más oportunidades de analizar y adaptarse a la forma que trabajan

Los objetivos y actividades del Sprint se planifican al comienzo de cada Sprint en la reunión de Planificación Del Sprint. Al finalizar el Sprint el equipo demuestra lo que construyeron en la Revisión Del Sprint, y analizan su propia performance y deciden qué mejorar en la Retrospectiva Del Sprint.

La duración de las reuniones del Sprint (Planificación, Revisión y Retrospectiva) se escalan de acuerdo a la duración del Sprint.

El progreso del Sprint se sigue con el Backlog Del Sprint y un gráfico Burndown.

Reglas del Sprint

- El Equipo puede buscar ayuda externa (consejos, recomendaciones, información y soporte) durante el Sprint.
- Nadie del exterior debe proveer al Equipo con instrucciones, comentarios o direcciones durante el Sprint. El Equipo es el único responsable de sí mismo: se auto-administra completamente, y cada miembro del equipo es responsable por la gestión del mismo.
- El Equipo se compromete a resolver items del Backlog Del Producto durante la reunión de Planificación Del Sprint. Nadie fuera del Equipo debe cambiar los items del Backlog Del Producto que el equipo ya comprometió.
- Si el Sprint se torna no viable, el Scrum Master puede terminar anormalmente el Sprint y organizar una nueva reunión de Planificación

Del Sprint para iniciar un Sprint nuevo. El Scrum Master puede realizar este cambio a su criterio, o a pedido del Dueño Del Producto o del Equipo. El Sprint podría ser no viable si la tecnología no funciona, las condiciones del negocio cambian, o si alguien exterior interfiere con el Equipo.

- Si el Equipo siente que no podrá completar todos los items del Backlog del Producto comprometidos en el Sprint, puede consultar al Dueño Del Producto sobre qué items quitar. Si son demasiados los items a quitar, y el Sprint pierde así sentido, el Scrum Master puede terminar anormalmente el Sprint.
- Si el Sprint requiere un 20% más de trabajo que el planificado luego de dos días después de la Planificación Del Sprint, se necesita planificar mejor. Esto es algo para tratar en la Retrospectiva Del Sprint.
- Si el Equipo determina que puede resolver más items del Backlog del Producto de los que seleccionó durante la Planificación del Sprint, el Equipo le puede consultar al Dueño Del Producto sobre otros items para agregar al mismo **Sprint**.
- Los Miembros Del Equipo De Scrum tienen dos responsabilidades administrativas durante el Sprint: tienen que asistir a la Reunión Diaria De Scrum, y tienen que mantener actualizado el estado de los items del Backlog Del Sprint (por ejemplo, estimado en horas restantes). Se pueden agregar nuevas tareas al Backlog Del Sprint a medida que vayan surgiendo.
- Si los miembros del Equipo informan el mismo item el mismo día, tienen que planificar mejor y descomponer las tareas con un mayor nivel de granularidad.
- El Equipo tiene que adherir a los estándares y convenciones existentes para el desarrollo, prácticas y arquitectura.

Meta del Sprint

Así como la Visión del Proyecto comunica el objetivo global y el espíritu del proyecto, es también útil contar con una visión para cada Sprint que encapsule todas las tareas en una "Meta". Por ejemplo, en vez de que el "Sprint 6" sea un

Sprint más donde se construyen cosas, se podría convertir en el "Sprint del sistema de pagos". Una meta del Sprint puede ser algo como:

- "En este Sprint se logrará que los usuarios puedan ingresar al sitio, recuperar un password olvidado, y administrar su perfil"

La Meta del Sprint se acuerda durante la reunión de Planificación Del Sprint al comienzo del Sprint.

Cancelación del Sprint

Un sprint puede ser cancelado antes de tiempo cuando:

- El equipo cancela un sprint si sienten que no pueden completar la Meta del Sprint
- La gerencia puede cancelar el Sprint si circunstancias externas anulan el valor de la Meta del Sprint

Cuando un Sprint se termina anormalmente, el próximo paso es realizar una nueva reunión de Planificación Del Sprint, en donde se revisa la razón de terminación.

Decidiendo las Historias para el Sprint

Una de las principales actividades durante la planificación de Sprint es decidir qué historias se incluyen en el Sprint. Más específicamente, qué historias de la Pila de Producto copiar en la Pila de Sprint.

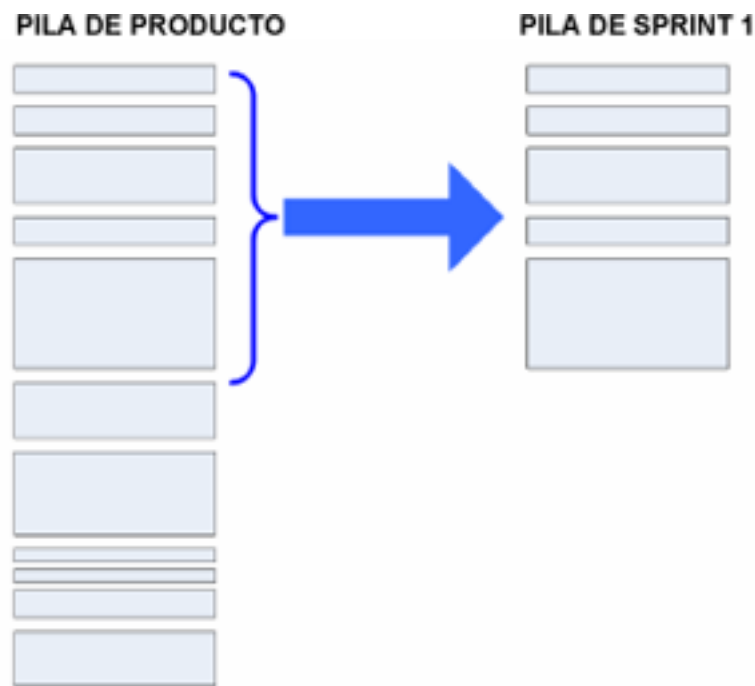


Figura 2.4: Seccionamiento de Historias de Usuario para el Sprint.

Observando la Figura 2.4, Cada rectángulo representa una historia, ordenadas por importancia. La historia más importante está al principio de la lista. El tamaño de cada rectángulo representa el tamaño de la historia (es decir, el tiempo estimado en puntos de historia). La altura del corchete azul representa la velocidad estimada del equipo, es decir, cuántos puntos de historia cree el equipo que puede completar durante el próximo Sprint.

La Pila de Sprint de la derecha es una instantánea de las historias la Pila de Producto. Representa las historias a las que el equipo se compromete durante este Sprint. El equipo puede oír al dueño del Sprint o al Dueño de Producto ni nadie más.

Esto plantea dos cuestiones:

1. ¿Cómo decide el equipo qué historias incluir en el Sprint?
2. ¿Cómo puede el Dueño de Producto alterar la decisión del equipo?

Mockup

Desde la llegada de la web 2.0, las aplicaciones móviles, las redes sociales y la interconexión entre todas estas cada vez se hace más complejo el acceso y utilización de las nuevas tecnologías. Por eso mismo es necesario, casi tanto como buenas funcionalidades, hacerlas usables y fáciles de utilizar. Aparece entonces el concepto de Mockup, o lo que sería lo mismo una maqueta.

Mediante estos Mockups o maquetas lo que se pretende es hacer un boceto preliminar de cómo sería la interfaz de nuestra aplicación y la navegación de está entre las diferentes secciones y vistas de nuestra aplicación, para así tener de forma más visual una imagen de todo el proceso, y poder dividir mejor tareas, detectar errores en el diseño y poder incorporar cambios de forma sencilla, así como ver el impacto en el resto de la aplicación y su forma de trabajar con ella.

Muchas veces el éxito de una aplicación puede residir en la forma en que se presenta al usuario la información y cómo puede trabajar con ella. No es lo mismo tener que hacer varios clicks para acceder a un listado de datos que, por ejemplo, se sabe que son datos frecuentados disponer de accesos directos. Esto cobra mucho más sentido con el boom de las aplicaciones móviles ya que no es posible mostrar toda la información en una misma vista y hay que optimizar la navegación.

Se dispone de herramientas tanto de escritorio como online, de pago y gratuitas para poder realizar estos bocetos.

2.6 Bases de Datos

Definición

Las bases de datos son el método preferido para el almacenamiento estructurado de datos. Desde las grandes aplicaciones multiusuario, hasta los teléfonos móviles y las agendas electrónicas utilizan tecnología de bases de datos para asegurar la integridad de los datos y facilitar la labor tanto de usuarios como de los programadores que las desarrollaron.

Una base de datos (cuya abreviatura es BD) es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible. Diferentes programas y diferentes usuarios deben poder utilizar estos datos. Por lo tanto, el concepto de base de datos generalmente está relacionado con el de red ya que se debe poder compartir esta información. De allí el término base. "Sistema de información" es el término general utilizado para la estructura global que incluye todos los mecanismos para compartir datos que se han instalado.

¿Por qué utilizar una base de datos?

Una base de datos proporciona a los usuarios el acceso a datos, que pueden visualizar, ingresar o actualizar, en concordancia con los derechos de acceso que se les hayan otorgado. Se convierte más útil a medida que la cantidad de datos almacenados crece.

Una base de datos puede ser local, es decir que puede utilizarla sólo un usuario en un equipo, o puede ser distribuida, es decir que la información se almacena en equipos remotos y se puede acceder a ella a través de una red. La principal ventaja de utilizar bases de datos es que múltiples usuarios pueden acceder a ellas al mismo tiempo.

Administración de la Base de Datos

Rápidamente surgió la necesidad de contar con un sistema de administración para controlar tanto los datos como los usuarios. La administración de bases de datos se realiza con un sistema llamado DBMS².

El DBMS es un conjunto de servicios (aplicaciones de software) para administrar bases de datos, que permite:

- Un fácil acceso a los datos.
- El acceso a la información por parte de múltiples usuarios.
- La manipulación de los datos encontrados en la base de datos (insertar, eliminar, editar).

² (Database management system [Sistema de administración de bases de datos])

El DBMS puede dividirse en tres subsistemas:

- El sistema de administración de archivos:
 - Para almacenar información en un medio físico.
- El DBMS interno:
 - Para ubicar la información en orden.
- El DBMS externo:
 - Representa la interfaz del usuario.

Base de datos Relacional

Una base de datos relacional es una base de datos que cumple con el modelo relacional, el cual es el modelo más utilizado en la actualidad para implementar bases de datos ya planificadas. Permiten establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), y a través de dichas conexiones relacionar los datos de ambas tablas, de ahí proviene su nombre: "Modelo Relacional".

Claves Primarias

Una clave primaria es una clave única elegida entre todas las candidatas que define unívocamente a todos los demás atributos de la tabla, para especificar los datos que serán relacionados con las demás tablas. La forma de hacer esto es por medio de claves foráneas.

Sólo puede existir una clave primaria por tabla y ningún campo de dicha clave puede contener valores NULL.

Características

- Una base de datos relacional se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y ajenas (o foráneas).
- Las claves primarias son la clave principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de datos.

- Las claves ajenas se colocan en la tabla hija, contienen el mismo valor que la clave primaria del registro padre; por medio de éstas se hacen las relaciones.

Ventajas

- Provee herramientas que garantizan evitar la duplicidad de registros.
- Garantiza la integridad referencial, así, al eliminar un registro elimina todos los registros relacionados dependientes.
- Favorece la normalización por ser más comprensible y aplicable.

Desventajas

- Presentan deficiencias con datos gráficos, multimedia, CAD y sistemas de información geográfica.
- No se manipulan de forma manejable los bloques de texto como tipo de dato.
- Las BDOO³ se propusieron con el objetivo de satisfacer las necesidades de las aplicaciones anteriores y así, complementar pero no sustituir a las bases de datos relacionales.

Diseño de las Bases de Datos Relacionales

El primer paso para crear una base de datos, es planificar el tipo de información que se quiere almacenar en la misma, teniendo en cuenta dos aspectos: la información disponible y la información que necesitamos.

La planificación de la estructura de la base de datos, en particular de las tablas, es vital para la gestión efectiva de la misma. El diseño de la estructura de una tabla consiste en una descripción de cada uno de los campos que componen el registro y los valores o datos que contendrá cada uno de esos campos.

Los campos son los distintos tipos de datos que componen la tabla, por ejemplo: nombre, apellido, domicilio. La definición de un campo requiere: el nombre del campo, el tipo de campo, el ancho del campo, etc.

³ Bases de datos orientadas a objetos

Los registros constituyen la información que va contenida en los campos de la tabla, por ejemplo: el nombre del paciente, el apellido del paciente y la dirección de este. Generalmente los diferentes tipos de campos que se pueden almacenar son los siguientes: Texto (caracteres), Numérico (números), Fecha / Hora, Lógico (informaciones lógicas si/no, verdadero/falso, etc.), imágenes.

En resumen, el principal aspecto a tener en cuenta durante el diseño de una tabla es determinar claramente los campos necesarios, definirlos en forma adecuada con un nombre especificando su tipo y su longitud.

En la Figura 2.5 se muestra un ejemplo de cómo se construye una base de datos relacional.

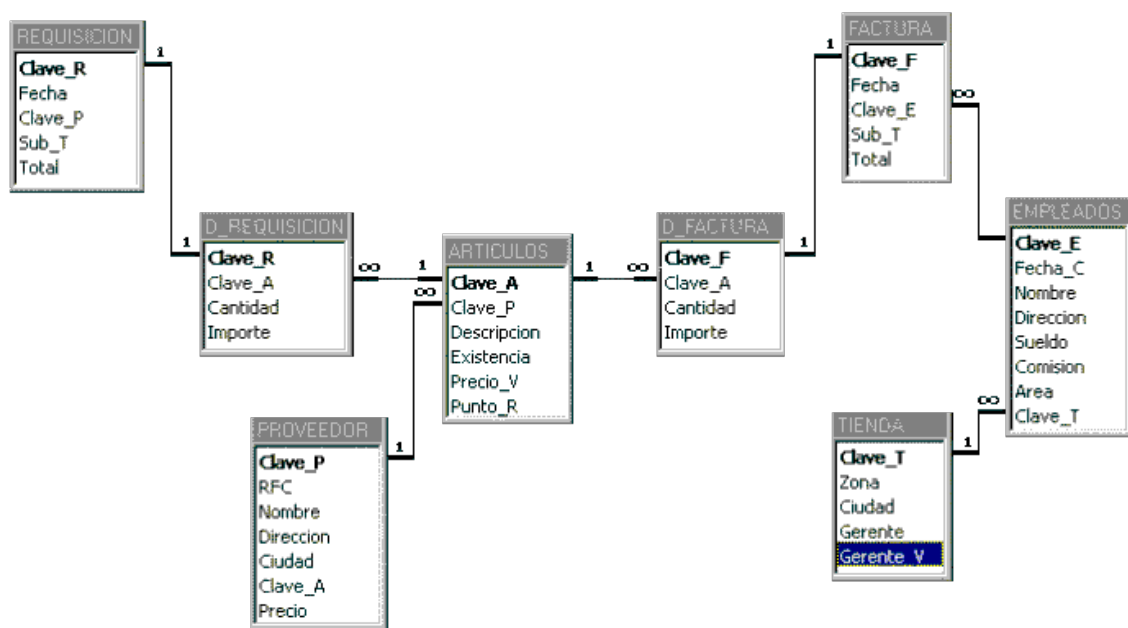


Figura 2.5: Diseño de las tablas en una base de datos relacional.

CAPITULO 3 Desarrollo

3.1 Estructura de la aplicación

El desarrollo se compone de los siguientes elementos

- Entorno administrativo
 - **Login:** Ingreso al sistema por medio de un Usuario y Contraseña.
 - **Entorno de desarrollo:** Parte en la cual el diseñador tiene la vista de los controles y herramientas las cuales servirán para la publicación del boletín.
 - **Galería Imágenes:** Modulo que servirá para almacenar todas las imágenes tanto principales como de secciones que se han publicado o guardado.
 - **Carga de imágenes:** Modulo el cual estará destinado a cargar las imágenes que se han guardado a lo largo de los desarrollos.
 - **Envío de Email:** Función que estará destinada al envío de correo electrónico a los encargados del diseño y publicación del boletín, así como al encargado por parte de Cinepolis.
 - **Preview:** Generará una vista previa del boletín.
 - **Detección de coordenadas:** Herramienta que servirá para identificar los puntos de acceso a las secciones del boletín.
- Entorno Cliente
 - **Boletín:** Vista la cual tendrán los visitantes del boletín.
 - **Secciones:** Vista de las secciones contenidas en el boletín.

Estos módulos son los que se construirán ya que son los requerimientos que se pidieron en el desarrollo. Durante la examinación de los módulos se crearon y diseñaron los Mockup's que se muestran en las siguientes figuras.

3.2 Historias de Usuario

Dentro del desarrollo se tuvieron que elaborar las historias de usuario, esto para corroborar los requerimientos que el cliente quiere que se realicen en el

sistema. Las siguientes tablas muestran las historias de usuario que se elaboraron después de la entrevista.

Numero:	US001
Nombre:	Login
Descripción:	Creación de un módulo que permita a los usuarios del sistema ingresar por medio de un Usuario y una Contraseña.
Criterios de Aceptación	<ul style="list-style-type: none"> - Que el usuario no se repita en los demás registros. - El Password debe ser de más de 6 caracteres. - Debe de mandar error si los datos al ingresar son incorrectos.
Prioridad	2
Hrs Estimadas:	5 Hrs
Observaciones	

Numero:	US002
Nombre:	Cerrar Sesión
Descripción:	Función que permita cerrar la sesión del usuario y salga del entorno de desarrollo.
Criterios de Aceptación	<ul style="list-style-type: none"> - Botón del lado superior derecho del entorno de desarrollo. - Al cerrar la sesión debe de salirse del sistema y llevarlo al lado cliente.
Prioridad	2
Hrs Estimadas:	2 Hrs

Observaciones	
----------------------	--

Numero:	US003
Nombre:	Botón Nuevo Boletín
Descripción:	Función que servirá para crear un nuevo boletín.
Criterios de Aceptación	<ul style="list-style-type: none"> - Limpia el entorno de desarrollo. - Resetea el control del número del boletín. - Limpia el panel para los botones de contenido.
Prioridad	4
Hrs Estimadas:	3 Hrs
Observaciones	

Numero:	US004
Nombre:	Botón Eliminar Boletín
Descripción:	Función para la eliminación de un boletín cuando ya no se utilice.
Criterios de Aceptación	<ul style="list-style-type: none"> - El boletín será eliminado de los registros. - El boletín se elimina, mientras que las imágenes insertadas quedan en los registros de la base de datos. - Las imágenes de link de contenido también deben permanecer.
Prioridad	4
Hrs Estimadas:	2 Hrs
Observaciones	

Numero:	US005
----------------	-------

Nombre:	Galería Imágenes
Descripción:	Ventana en la cual se mostrarán las imágenes que han sido almacenadas para los boletines creados. Las imágenes podrán ser seleccionadas ya sea para un nuevo boletín o para una imagen de link de contenido.
Criterios de Aceptación	<ul style="list-style-type: none"> - La ventana podrá ser llamada desde el panel de agregación de links de contenido o desde el panel de diseño del boletín principal. - El listado de las imágenes debe contener una vista previa de la imagen, fecha de agregación y nombre. - El listado debe de estar paginado.
Prioridad	4
Hrs Estimadas:	5 Hrs
Observaciones	

Numero:	US006
Nombre:	Vista Previa
Descripción:	Crear un botón que permita generar una vista previa del boletín principal que se está creando.
Criterios de Aceptación	<ul style="list-style-type: none"> - La vista previa debe de abrirse en una ventana separada del entorno de trabajo del boletín.

Prioridad	2
Hrs Estimadas:	2 Hrs
Observaciones	

Numero:	US007
Nombre:	Envío de Email
Descripción:	Esta función envía un correo electrónico a los encargados de la publicación del boletín así como a los administradores del diseño.
Criterios de Aceptación	<ul style="list-style-type: none"> - El correo debe contener el boletín principal y un enlace a la página del boletín como contenido. - El enlace debe contener "Si no puedes ver esta imagen da click aquí "
Prioridad	4
Hrs Estimadas:	4 Hrs
Observaciones	

Numero:	US008
Nombre:	Detección de coordenadas para links de contenido
Descripción:	Mientras se crea el boletín, el usuario puede dar click en la imagen principal y generarse una acción de captura de coordenadas las cuales deben corresponder al lugar de la imagen donde se generó la acción del click.
Criterios de Aceptación	<ul style="list-style-type: none"> - Las coordenadas deben de mostrarse en numeración.

Prioridad	5
Hrs Estimadas:	5 Hrs
Observaciones	

Numero:	US009
Nombre:	Generación de links de contenido de forma dinámica.
Descripción:	El usuario puede agregar los links de contenido que desee. Estos links se generan de forma dinámica.
Criterios de Aceptación	<ul style="list-style-type: none"> - Cada link de contenido debe contener sus coordenadas en la imagen. - Cada link de contenido debe contener la ruta de su imagen contenido.
Prioridad	5
Hrs Estimadas:	8 Hrs
Observaciones	

Numero:	US010
Nombre:	Eliminar Link de Contenido
Descripción:	El usuario puede eliminar el link de contenido.
Criterios de Aceptación	
Prioridad	4
Hrs Estimadas:	3 Hrs
Observaciones	

Numero:	US011
Nombre:	Boletines por Zona
Descripción:	El usuario podrá seccionar los

	boletines por zona.
Criterios de Aceptación	<ul style="list-style-type: none"> - Deberán existir las siguientes zonas: Brazil, USA, Corporativo, Mailing, LATAM, India y Conjuntos. - Cada zona tendrá la numeración de los boletines.
Prioridad	5
Hrs Estimadas:	4 Hrs
Observaciones	

Numero:	US011
Nombre:	Lado Cliente
Descripción:	El cliente podrá visitar la página usando la dirección hoy.cinepolis.com/ la cual mostrará el boletín informativo.
Criterios de Aceptación	<ul style="list-style-type: none"> - La página debe mostrar siempre el último boletín creado.
Prioridad	5
Hrs Estimadas:	8 Hrs
Observaciones	

3.3 Mockup's del desarrollo de la aplicación

De acuerdo a cada historia de usuario se diseñaron los siguientes Mockups para el desarrollo de la aplicación.

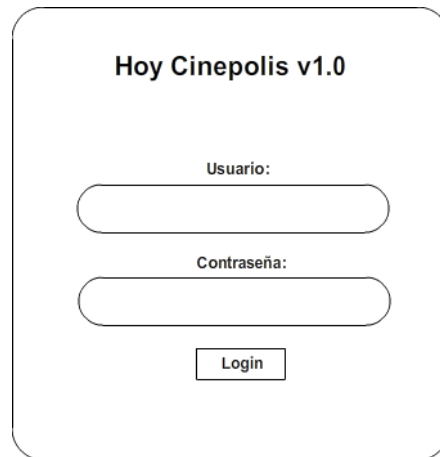


Figura 3.1 Diseño del dialogo para iniciar sesión en la aplicación.

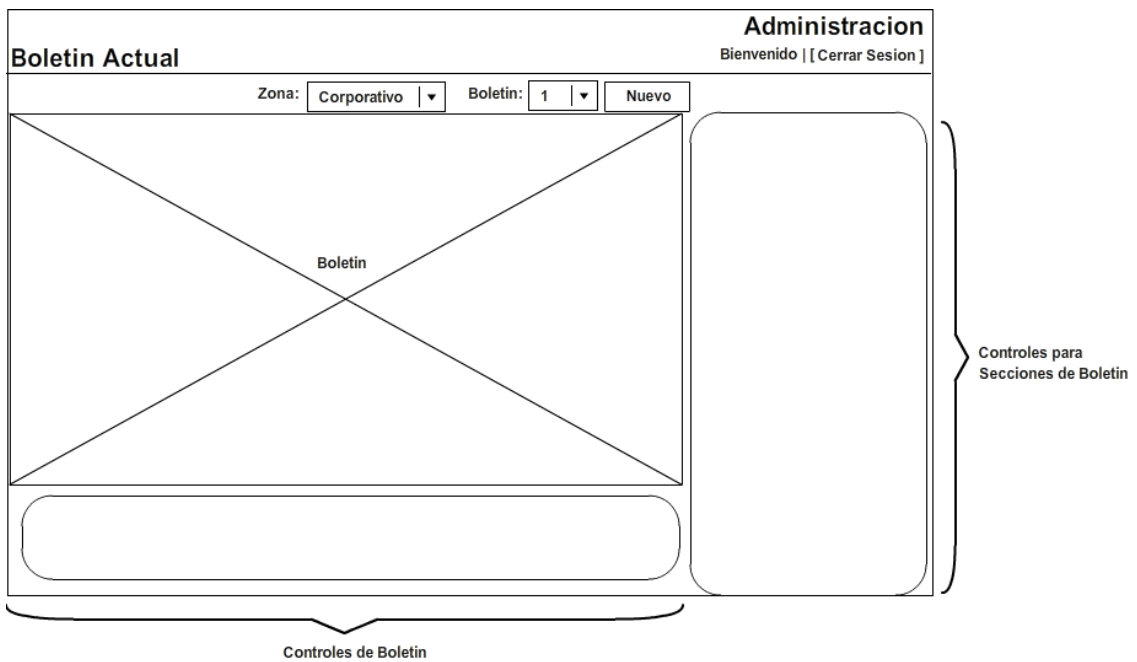


Figura 3.2 Diseño para el entorno de desarrollo de los diseñadores.

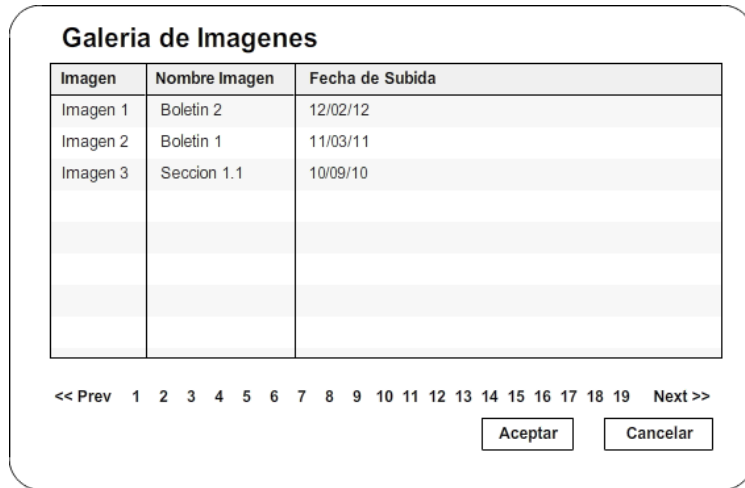


Figura 3.3 Mockup del entorno de la galería de imágenes.

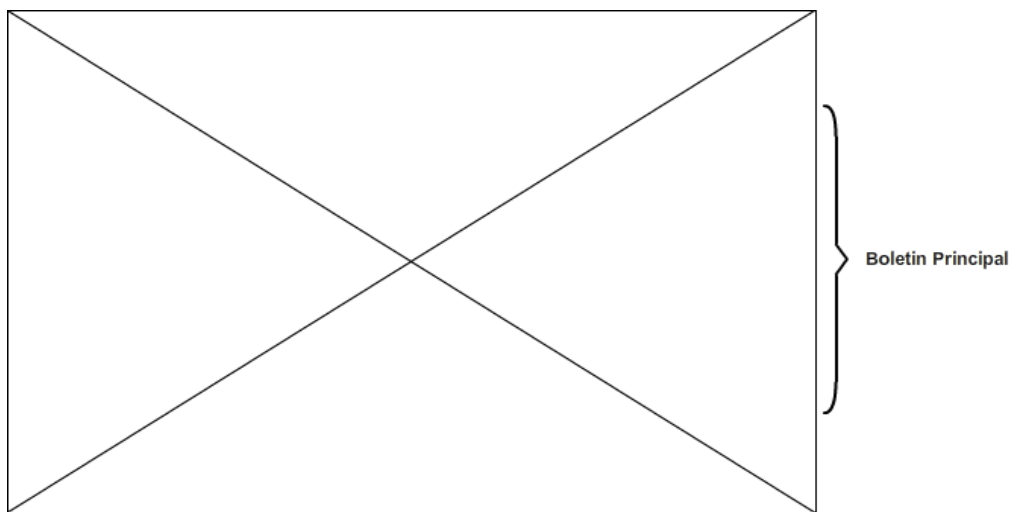


Figura 3.4 Mockup de la parte cliente donde se mostrará el boletín final.

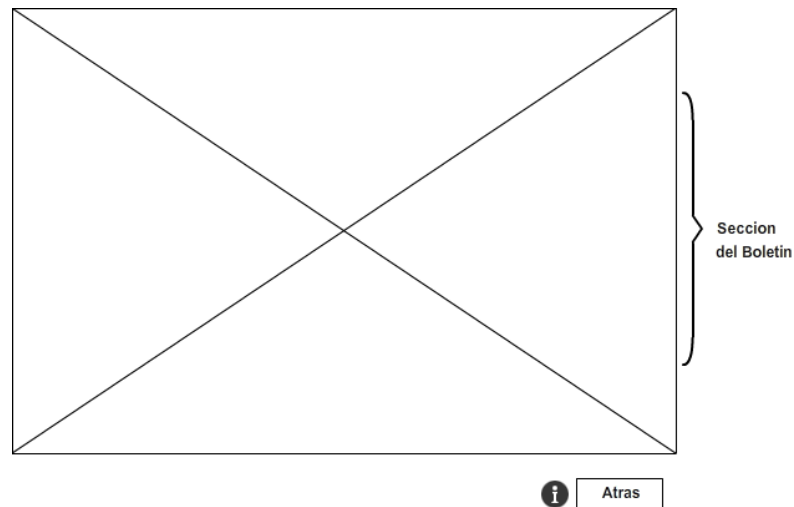


Figura 3.5 Mockup de la parte de la sección de un boletín.

3.4 Calendarización de los módulos para el desarrollo de la aplicación.

Los módulos se crearan de acuerdo a las historias de usuario y se dividen en los siguientes puntos:

- Desarrollo de la Base de Datos (2 Día)
- Accesos (2 Día)
- Publicación (5 Días)
- Edición (8 Días)
- Lado Cliente (2 Día)
- QA (2 Días)

En el desarrollo de la base de datos, únicamente se realizará la inserción de las tablas obtenidas en el análisis de los datos y que fueron mencionadas así como explicadas en el subtema anterior. En el caso del módulo Login, se realizará la sección en la que el administrador ingresará por medio de un usuario y una contraseña al editor de boletines. Este login será accesado por

medio de una dirección administrativa encriptada la cual será definida de la siguiente manera <http://hoy.cinepolis.com/A2592BE117BA8/>.

El módulo de Edición y Publicación será en el que el usuario pueda subir las imágenes, seleccionar las coordenadas de los que serán botones de las imágenes contenido, envío de correo electrónico a clientes para que puedan realizar una vista previa del boletín publicado.

El módulo Preview es donde los usuarios puedan realizar una vista previa de la imagen inicial, así como el acceso al contenido del boletín.

El modulo del Lado Cliente es simplemente el acceso a los visitantes del boletín, en este el visitante puede acceder al contenido de cada link ya sea una imagen contenido o un link externo.

3.5 Desarrollo de la Base de Datos

Realizando un análisis de las Historias de Usuario identificando las entidades y elaborando una lista de los datos que necesitan ser almacenados para que el proceso de elaboración y publicación de un boletín sea viable, el resultado llega a ser el listado que se muestra a continuación en la Figura 3.6.

Table	Owner	Date created
aspnet_Applications	dbo	02/09/2012 11:28:42 AM
aspnet_Membership	dbo	02/09/2012 11:28:42 AM
aspnet_Paths	dbo	02/09/2012 11:28:43 AM
aspnet_PersonalizationAllUsers	dbo	02/09/2012 11:28:43 AM
aspnet_PersonalizationPerUser	dbo	02/09/2012 11:28:43 AM
aspnet_Profile	dbo	02/09/2012 11:28:42 AM
aspnet_Roles	dbo	02/09/2012 11:28:43 AM
aspnet_SchemaVersions	dbo	02/09/2012 11:28:42 AM
aspnet_Users	dbo	02/09/2012 11:28:42 AM
aspnet_UsersInRoles	dbo	02/09/2012 11:28:43 AM
aspnet_WebEvent_Events	dbo	02/09/2012 11:28:43 AM
ContentGallery	dbo	02/22/2012 11:19:13 AM
ExternalLinks	dbo	02/22/2012 11:19:36 AM
IndexGallery	dbo	02/22/2012 11:16:30 AM
Paths	dbo	02/22/2012 11:14:46 AM
Zones	dbo	02/22/2012 11:15:47 AM

Figura 3.6 Muestra las tablas correspondientes a la aplicación de publicación de Boletín.

Primeramente se estudió la parte de la imagen inicial del boletín, la cual, para que los boletines o imágenes de inicio sean almacenadas se elaboró la siguiente tabla con sus campos correspondientes como se muestra en la Figura 3.7


Key	Column	Type	Length	Prec.	Scale	Nullable	Default	Rule	Id.	Id. Incr.	Id. seed	Row GUID
	IdIndexGallery	uniqueidentifier	16	0	0	False			False	0	0	False
	Name	nvarchar	250	0	0	False			False	0	0	False
	Path	nvarchar	250	0	0	False			False	0	0	False
	DateUpload	date	3	10	0	False			False	0	0	False
	IdZone	uniqueidentifier	16	0	0	False			False	0	0	False
	NoNewsletter	int	4	10	0	True			False	0	0	False

Figura 3.7 Muestra los campos de la tabla IndexGallery que servirán para el almacenado de las imágenes principales del boletín.

La tabla consta de los campos que se van a explicar a continuación:

- ❖ *IdIndexGallery*: ID o identificador de tipo GUID* el cual nos servirá como valor único de cada registro y así poder identificarlo en las búsquedas.
- ❖ *Name*: Nombre que se le asignará a la imagen al momento de guardar un registro en ella.
- ❖ *Path*: Ruta de la imagen principal del boletín, la cual es almacenada en forma encriptada.
- ❖ *DateUpload*: Fecha en que la imagen fue registrada.
- ❖ *IdZone*: Identificador y llave foránea que hará relación con la zona a la cual pertenece el boletín.
- ❖ *NoNewsletter*: Número del boletín al cual pertenece la imagen.

Dentro de la tabla la llave foránea se muestra en la Figura 3.8

IndexGallery Relationships			
Relationship	Primary key table	Foreign key table	Type
 FK_IndexGallery_Zones	IndexGallery	Zones	
	IdZone	IdZone	UniqueIdentifier

Figura 3.8 Muestra la llave foránea IdZone en la tabla IndexGallery

Esta llave foránea nos ayudará a relacionar la zona a la cual pertenece cada uno de los boletines.

En la Figura 3.9 se muestra la tabla ContentGallery que es consecuente de la anteriormente mostrada; en esta tabla se almacenarán las imágenes de los contenidos del boletín, así como la configuración de su acceso por el medio electrónico.


Key	Column	Type	Length	Prec.	Scale	Nullable	Default	Rule	Id.	Id. Incr.	Id. seed	Row GUID
	IdContentGallery	uniqueidentifier	16	0	0	False			False	0	0	False
	Name	nvarchar	250	0	0	False			False	0	0	False
	Path	nvarchar	250	0	0	False			False	0	0	False
	DateUpload	date	3	10	0	False			False	0	0	False
	IdIndexGallery	uniqueidentifier	16	0	0	False			False	0	0	False
	X1Button	int	4	10	0	False			False	0	0	False
	Y1Button	int	4	10	0	False			False	0	0	False
	X2Button	int	4	10	0	False			False	0	0	False
	Y2Button	int	4	10	0	False			False	0	0	False

Figura 3.9 Tabla ContentGallery la cual almacena las imágenes contenido del boletín.

A continuación se explicarán los campos que contiene esta tabla:

- ❖ *IdContentGallery*: ID o identificador único del registro.
- ❖ *Name*: Nombre de la imagen a ser almacenada.
- ❖ *Path*: Ruta de la imagen contenido.
- ❖ *DateUpload*: Fecha en la cual la imagen es cargada al servidor.
- ❖ *IdIndexGallery*: ID o identificador único el cual hace referencia hacia la imagen principal a la que corresponde la imagen contenido insertada.
- ❖ *X1Button*, *Y1Button*, *X2Button*, *Y2Button*: Corresponden a las coordenadas en la imagen principal la cual servirá como acceso a las imágenes contenido, creando con éstas coordenadas de los links en la imagen principal.

La Figura 3.10 muestra la llave foránea la cual hace el enlace hacia la tabla de IndexGallery ya que con ella relacionamos la imagen contenido con la imagen principal en el boletín.

Relationship	Primary key table	Foreign key table	Type
FK_IndexGallery_ContentGallery	ContentGallery	IndexGallery	
	IdIndexGallery	IdIndexGallery	UniqueIdentifier

Figura 3.10 Llave foránea hacia IndexGallery que contiene la imagen principal del boletín.

De acuerdo a que los boletines tienen que ser clasificados por zonas ya que los requerimientos así fueron, se decide crear una tabla de Zonas que únicamente almacene las Zonas disponibles para los boletines y así sean organizados por los administradores de la aplicación. La tabla se muestra en la Figura 3.11

Key	Column	Type	Length	Prec.	Scale	Nullable	Default	Rule	Id.	Id. Incr.	Id. seed	Row GUID
	IdZone	uniqueidentifier	16	0	0	False			False	0	0	False
	IdPath	uniqueidentifier	16	0	0	False			False	0	0	False
	Name	nvarchar	250	0	0	False			False	0	0	False

Zones Indexes										
Index	Columns	Primary	Unique	Clustered	Ignore dup. keys	Unique key	Stats no recomp.	Fill factor	File group	
PK_Zones	IdZone	True	True	True	False	False	False	0	PRIMARY	

Zones Relationships			
Relationship	Primary key table	Foreign key table	Type
FK_Zones_Paths	Zones	Paths	
	IdPath	IdPath	UniqueIdentifier

Figura 3.11 Muestra la tabla Zone y su Foreign key hacia Path.

La tabla Path corresponde a una serie de datos los cuales permiten tener diferentes directorios, organizando así las imágenes de acuerdo a su zona, esto quiere decir que las imágenes serán organizadas por medio de carpetas con el nombre encriptado y que cada una corresponderá a una zona específica que contendrá las imágenes de los boletines. La figura 3.12 muestra la tabla resultante del análisis de la información.

Key	Column	Type	Length	Prec.	Scale	Nullable	Default	Rule	Id.	Id. Incr.	Id. seed	Row GUID
	IdPath	uniqueidentifier	16	0	0	False			False	0	0	False
	DecryptPath	nvarchar	250	0	0	False			False	0	0	False
	EncryptPath	nvarchar	250	0	0	False			False	0	0	False

Figura 3.12 Tabla de Path que contendrá las rutas de las imágenes por zona.

Dentro de los requerimientos se pidió que en las imágenes de boletín tanto principales como de contenido, tendría que tener la posibilidad de ingresar links externos, es decir, que habría algunos botones que redireccionarán a otras páginas en una ventana por separado. Para ello se creó la tabla de ExternalLinks, la cual se muestra en la figura 3.13.

Key	Column	Type	Length	Prec.	Scale	Nullable	Default	Rule	Id.	Id. Incr.	Id. seed	Row GUID
	IdExternalLink	uniqueidentifier	16	0	0	False			False	0	0	False
	IdContentGallery	uniqueidentifier	16	0	0	False			False	0	0	False
	Link	nvarchar	350	0	0	False			False	0	0	False
	X1Button	int	4	10	0	False			False	0	0	False
	Y1Button	int	4	10	0	False			False	0	0	False
	X2Button	int	4	10	0	False			False	0	0	False
	Y2Button	int	4	10	0	False			False	0	0	False

ExternalLinks Indexes											
Index	Columns	Primary	Unique	Clustered	Ignore dup. keys	Unique key	Stats no recomp.	Fill factor	File group		
	IdExternalLink	True	True	True	False	False	False	0	PRIMARY		
PK_ExternalLinks											

ExternalLinks Relationships			
Relationship	Primary key table	Foreign key table	Type
	ExternalLinks	ContentGallery	
	IdContentGallery	IdContentGallery	UniqueIdentifier

Figura 3.13 Tabla ExternalLink mostrando la llave foránea hacia ContentGallery ya que está relacionada con la imagen contenido.

Con estas tablas y datos la parte de base de datos es cubierta y con ello la edición así como la publicación puede ser almacenada. El siguiente paso sería la calendarización de los módulos a programar la cual se explica en el siguiente capítulo.

3.6 Desarrollo del Software

Una vez terminada la elaboración de la base de datos se continúa con lo que será la codificación del proyecto, en base a las historias de usuario y a la prioridad que se les dio al momento de elaborarlas es como se iniciará la construcción de los módulos.

3.6.1 Login

Como se había comentado el login será el acceso al sistema de edición, en la base de datos se insertó solo un registro como usuario administrativo el cual sería:

- Usuario: *HCAAdmin*
- Contraseña: *SuperAHC*

Las restricciones que se tiene al acceder al sistemas son las mismas que se tienen en cualquier sistema administrativo, en la siguiente lista tenemos los criterios los cuales se valoran al momento de ingresar al sistema.

- *Coincidencia en usuario y contraseña*
- *Usuario no debe de ser vacío*
- *Contraseña no debe ser vacía*

Tras tener estos criterios el UI quedó como se muestra en la figura 3.14

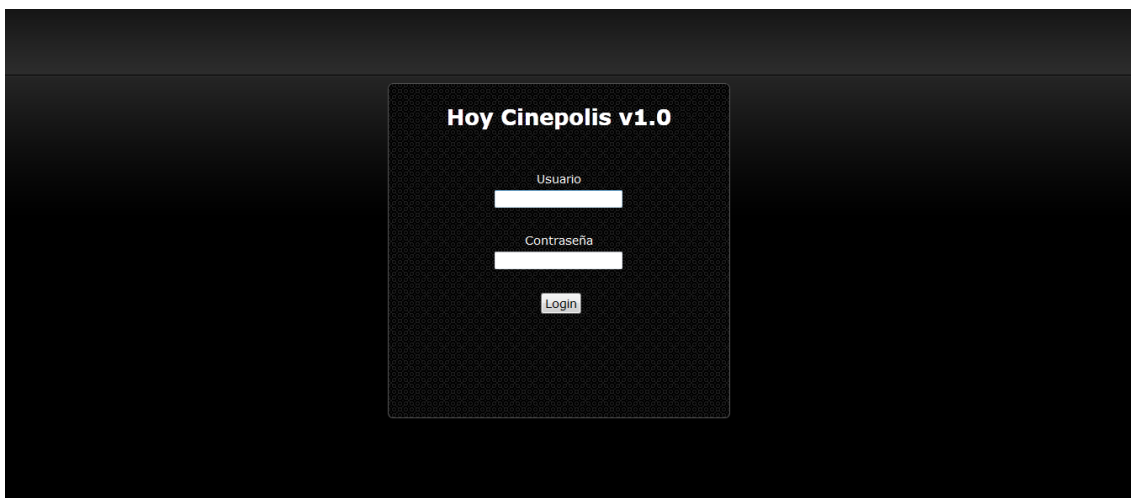


Figura 3.14 Interfaz Login terminada.

Las validaciones que se mencionaron anteriormente se realizan en el BE⁴, y se manifiestan en el UI como se muestra en la figura 3.15 en donde se ilustra los mensajes que serán visualizados si es que ocurre un error o un intento de ingreso erróneo.



Figura 3.15 Login con error de validación de entrada.

Cabe señalar que la sesión está abierta por un periodo de 5 minutos, si en ese tiempo el usuario tiene inactividad, el sistema automáticamente cerrará la sesión y pedirá de nuevo el ingreso por medio del usuario y la contraseña.

3.6.2 Edición

De acuerdo a las US que se escribieron en la entrevista donde se realizó la recopilación de requerimientos, el proceso de edición en el sistema será como se ilustra en el diagrama de la Figura 3.16, en donde se plasman los pasos que seguirán los diseñadores al momento de crear un nuevo boletín o de realizar una edición de uno ya creado anteriormente.

⁴ Back End (Programación Lógica de la Aplicación)

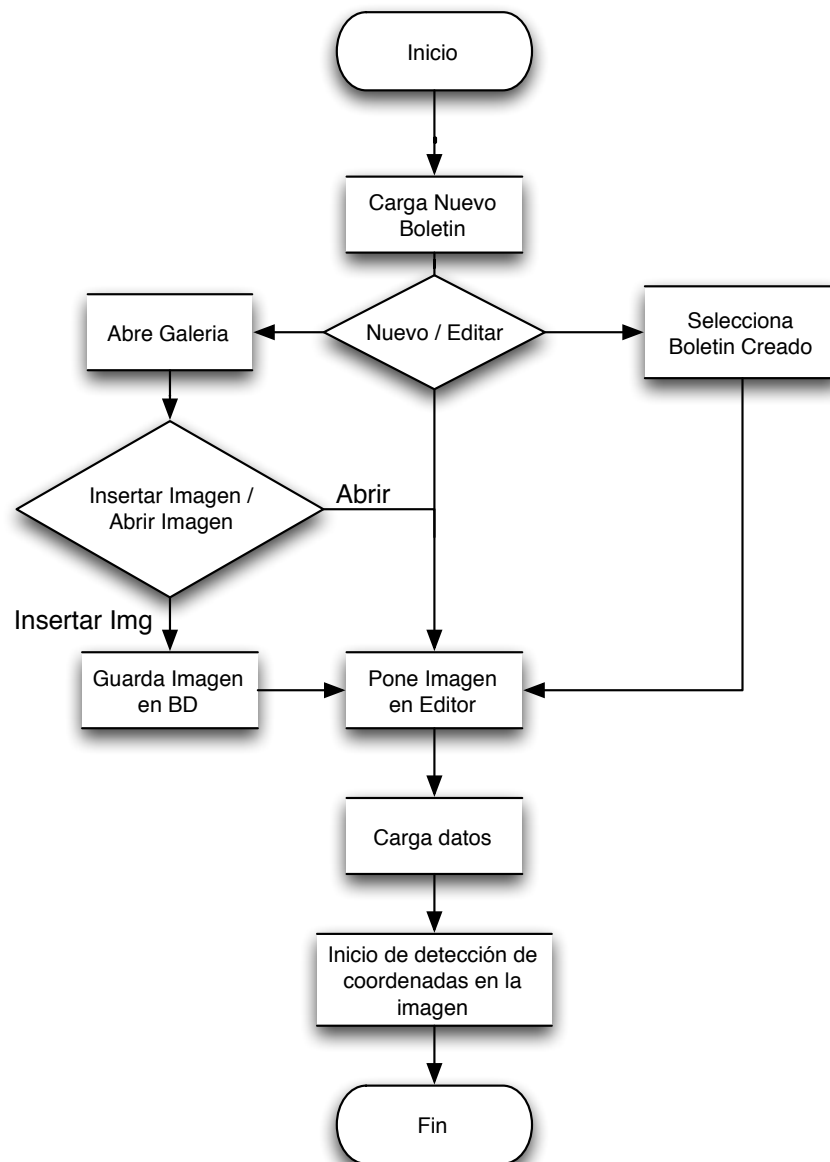


Figura 3.16 Diagrama de procesos de la edición para un boletín informativo.

Basándose en el diagrama de procesos para la edición y siguiendo la calendarización, se comenzó a programar la parte de la Galería de Imágenes y es la que se explica en el siguiente subtema.

3.6.2.1 Galería de Imágenes

La galería de imágenes se encargará de almacenar y de organizar las imágenes ya sean principales del boletín ó imágenes contenido. Dentro de los requerimientos de la creación de la galería están los siguientes:

- Ordenamiento por fecha.

- Imagen de Preview.
- Ventana con las siguientes columnas: Imagen Preview, Nombre de Imagen, Fecha de registro de la imagen.
- Paginación.
- Búsqueda de imágenes por nombre y/o fecha de subida.

Una vez desarrollada la UI⁵ el resultado queda como se muestra en la Figura 3.17 donde se ilustra la galería de imágenes con algunos registros con las especificaciones de las columnas que se pidieron.



Figura 3.17 Galería de Imágenes en la edición del boletín.

3.6.2.2 Cerrar Sesión

Al momento de iniciar la sesión o entrar al sistema, el usuario cuenta con 5 minutos como tiempo limite de inactividad para que el sistema le vuelva a pedir que ingrese un usuario y contraseña para reanudar la sesión.

⁵ User Interface (Interfaz de Usuario)

Otra de las formas en la que el usuario puede cerrar el sistema y evitar que el boletín sea modificado es por medio del botón de “Cerrar Sesión” que se ingresó en la parte superior derecha de la pantalla. Con ello el sistema cierra la pagina de edición y redirecciona a la pagina principal del boletín, es decir, a la visualización del ultimo boletín que se creó.

En la Figura 3.18 se muestra dónde se coloca el botón de cerrar sesión en la pantalla, y la redirección al ultimo boletín creado.

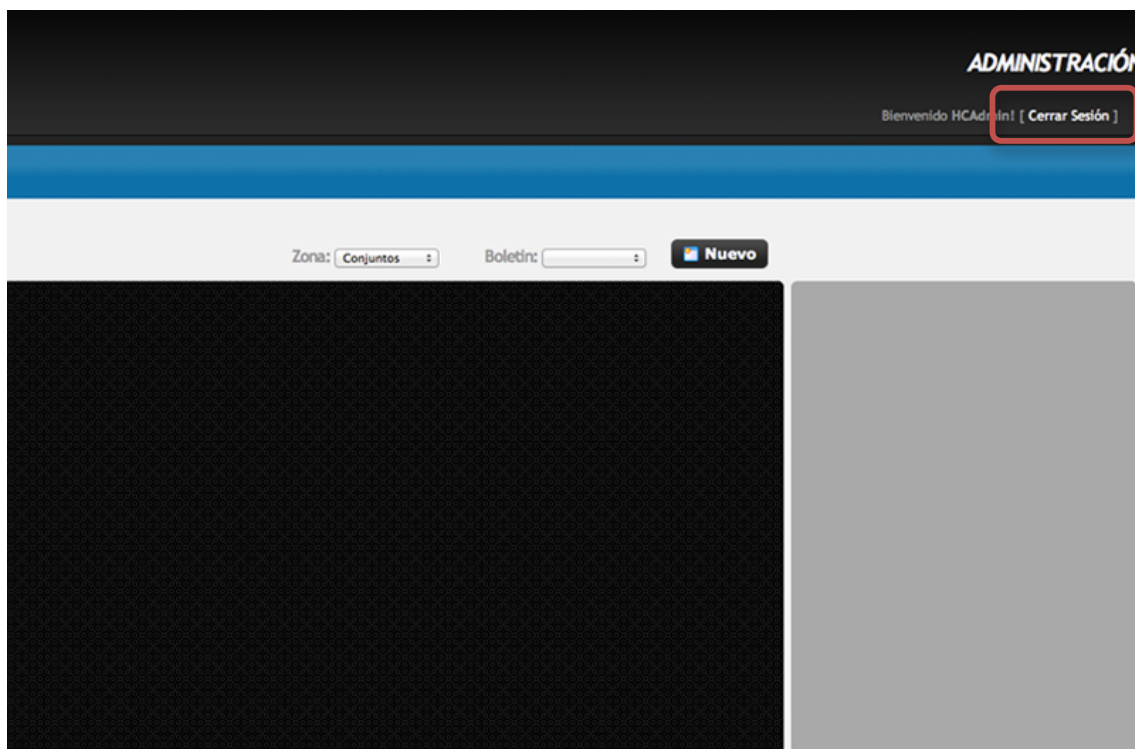


Figura 3.18 Muestra el botón al cierre de sesión.

3.6.2.3 Nuevo Boletín

El siguiente modulo de creación en la calendarización es la elaboración de un nuevo boletín. Por medio del botón “Nuevo” es como el usuario puede crear un boletín desde cero. De acuerdo a las especificaciones y al proceso la lógica se define de la siguiente manera:

1. Limpia Imagen de edición y habilita el botón de “Crear Boletín”.

2. El usuario puede insertar imagen nueva o selecciona imagen ya existente en la galería de imágenes.
3. Al presionar el botón de “Crear Boletín” el sistema genera nuevo registro del boletín con un numero consecutivo al de la sección en la que se encuentra y le asigna el nombre que el usuario desee.
4. El sistema notifica que se ha guardado con éxito el boletín y habilita el panel para la detección de coordenadas de links externos o imágenes contenido, Actualización del boletín, Eliminación del boletín, Vista Previa, Envío del boletín por E-Mail.

En las siguientes imágenes se muestra el proceso que se explica anteriormente.

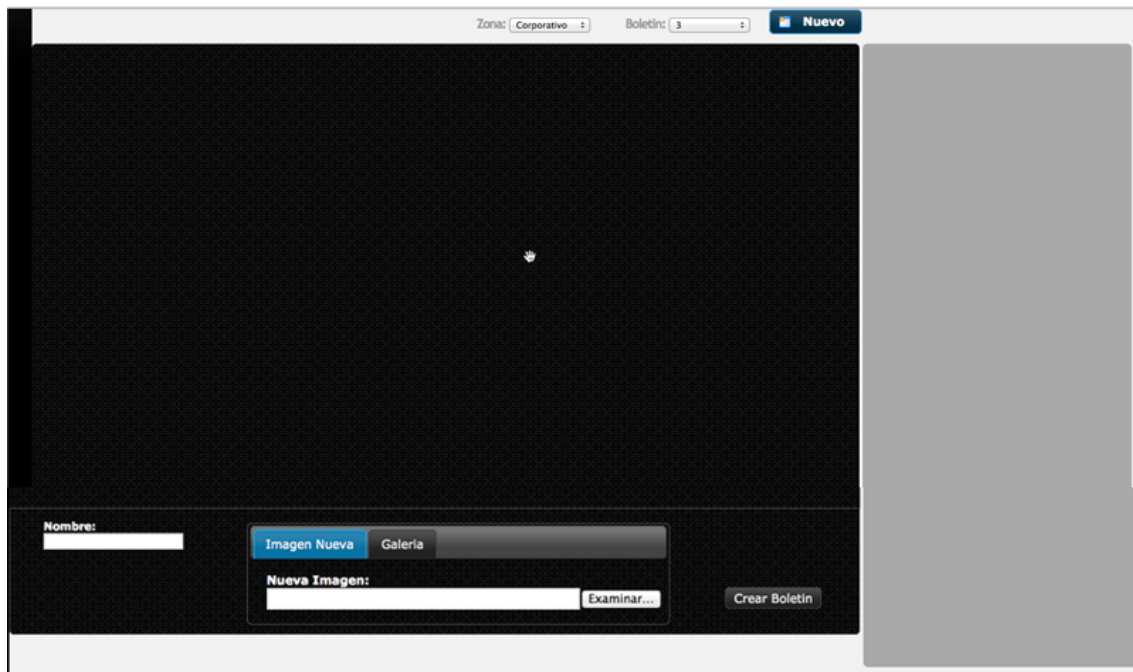


Figura 3.19 Muestra la plantilla limpia después de presionar el botón de “Nuevo” y habilita el botón de Crear Boletín en la parte inferior derecha.



Figura 3.20 Mostrando la imagen seleccionada de la galería ya cargada en el área de edición, en el nombre se le puso BoletinAdrixKi



Figura 3.21 Muestra el mensaje de que el boletín ha sido guardado y creado con éxito.

Una vez que el boletín ha sido creado el panel para la detección de coordenadas es visible en la parte derecha, la cual se muestra en la figura 3.22



Figura 3.22 Muestra la interfaz con el boletín creado; del lado derecho el panel de detección de coordenadas y la posibilidad de añadir imágenes contenido o links externos al boletín.

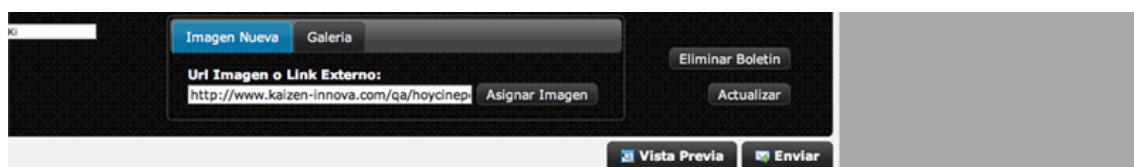


Figura 3.23 Muestra las opciones de Eliminación de boletín, Actualización de boletín, Vista previa y envío de boletín por E-Mail.

Por medio del botón de “Añadir Nuevo Link” que se encuentra ubicado en la parte derecha de la pantalla de edición, el usuario tiene la posibilidad de agregar nuevas imágenes contenido o links externos, en las siguientes figuras se ilustra esta acción.



Figura 3.24 Panel de imágenes contenido o links externos.

Por medio del panel ilustrado en la figura 3.24 es como el usuario puede agregar N cantidad de imágenes o links externos al boletín. En la Figura 3.25 se muestra el proceso de agregar una imagen contenido ya sea de la galería ó de un archivo que el usuario suba, así como el asignar dentro de la imagen principal las coordenadas de una imagen contenido o link externo.

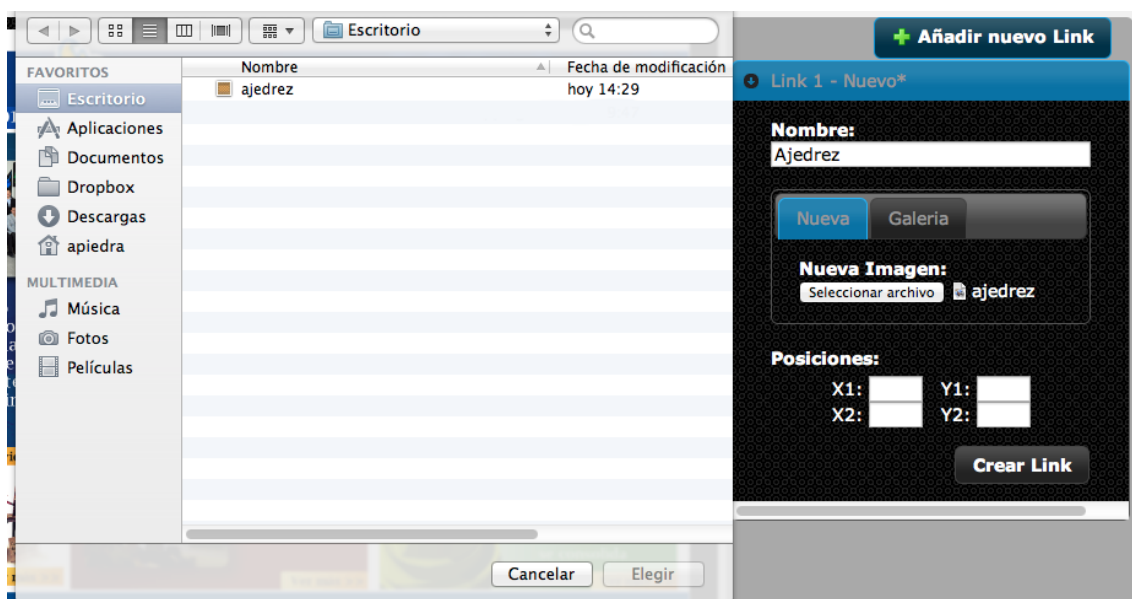


Figura 3.25 Ventana al seleccionar una imagen nueva para la imagen contenido.

3.6.2.4 Detección de coordenadas

La detección de coordenadas en la imagen se da mediante un disparador o trigger en jquery que nos da el pixel de la imagen en coordenadas (x,y) empezando desde la esquina superior izquierda la cual seria (0,0). Para el mapeo de links en la imagen se utilizó el tag de HTML llamado <map>.



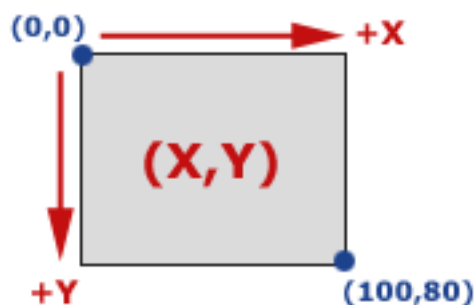
Figura 3.26 Muestra la detección de la coordenada en la imagen para indicar donde se creará el link.

Este tipo de tag puede hacer que parte de la imagen sea un enlace a otra página, es decir, puede hacer un mapa sobre la imagen de manera que secciones de la imagen sean enlaces. Las tags usadas para esto son:

1. Las tags <map>.....</map> identifican que se va a crear un mapa de imágenes. Generalmente, lleva asociado el atributo name= al que le sigue entre comillas el nombre del mapa.
2. La tag <area> define las áreas que se pueden activar en esa imagen. A esta tag le acompañan los siguientes atributos:
 - a. shape= Entre comillas estableceremos el tipo de área a definir. Puede tratarse de rect (rectangular), poly (poligonal) o circle

- (circular). En nuestro caso realizamos un tipo de shape rectangular.
- b. Coords= Entre comillas se indicaran los pares de coordenadas de cada punto del área a activar. Estas coordenadas se ajustan desde el sistema mediante la operación de detección de coordenadas que se desarrollo y que se explico previamente. En las áreas rectangulares deben especificarse las coordenadas de la esquina superior izquierda y las de la esquina inferior derecha.
 - c. Href= Como ya sabe, indica la dirección, entre comillas, de la página web a la que accede si se hace click en un área determinada, en nuestro caso, direccionará a la imagen que seleccionamos anteriormente en la galería de imágenes, link externo o a la imagen contenido que hayamos subido al sistema.

En la figura 3.27 muestra una imagen de las coordenadas según una imagen de 100 x 80 pixeles.



**El recuadro es una supuesta
imagen de 100x80 pixels**

Figura 3.27 Mapeo de imagen de 100 x 80 pixeles.

Como habíamos mencionado anteriormente, usamos un tipo de shape RECT que dibuja un rectángulo como link que es el que se muestra en la figura 3.28.

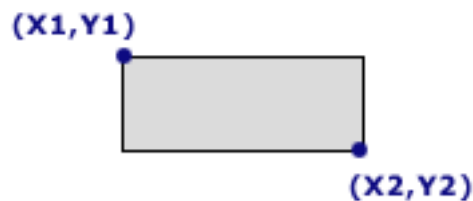


Figura 3.28 Rectángulo con las coordenadas [x1, y1], [x2, y2] para formar el link.

3.6.2.5 Inserción de Links Externos

Una de las peticiones del cliente fue que se pudiera realizar la inserción de N links a una pagina diferente del boletín, para ello dentro del panel de diseño se implemento un campo de texto en el cual el usuario puede ingresar una dirección URL de una pagina fuera del sitio del boletín, como se muestra en la figura 3.29.

Una captura de pantalla de una interfaz de usuario para configurar un link. El formulario tiene un fondo oscuro y un título '+ Añadir nuevo Link' en la parte superior. Debajo del título, hay un campo de texto con el texto 'Nuevo Link'. Hay dos pestañas, 'Nueva' y 'Galeria', con 'Galeria' seleccionada. Debajo de las pestañas, hay un campo de texto con el texto 'http://www.google.com' y un botón 'Asignar Imagen'. En la parte inferior, hay un campo 'Posiciones:' con cuatro sub-campos: 'X1: 896', 'Y1: 443', 'X2: 966' y 'Y2: 457'. En la parte inferior derecha, hay un botón 'Crear Link'.

Figura 3.29 URL externa en el link del boletín.

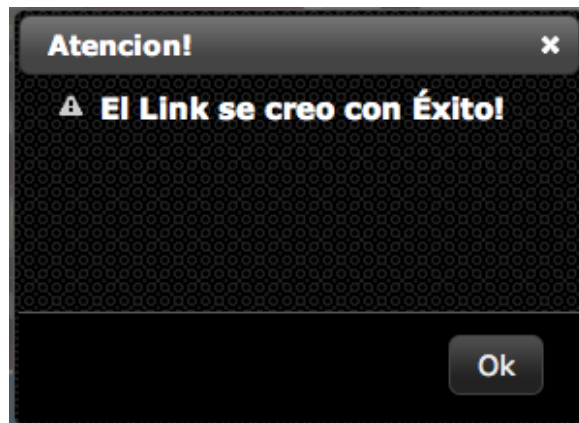


Figura 3.30 Mensaje de confirmación del registro del link al presionar el botón de “Crear Link”

Si el usuario requiere agregar un link adicional al boletín, lo puede realizar por medio del botón de “Añadir nuevo Link” ubicado en el panel de imágenes contenido como se indica en la figura 3.31

A screenshot of a mobile application interface for adding a new link. At the top, a button with a red arrow and a green plus sign is labeled "Añadir nuevo Link". Below it, a list shows "Link#1 - NuevoLink" and "Link 2 - Nuevo*" (the latter is highlighted in blue). The form for "Link 2" includes a "Nombre:" field with the value "LinkSecundario". Below this are two buttons: "Nueva" and "Galeria". The "Galeria" button is highlighted. Underneath is a "Url Imagen o Link Externo:" field and an "Asignar Imagen" button. At the bottom, there are four input fields for "Posiciones:" labeled "X1:", "Y1:", "X2:", and "Y2:". A "Crear Link" button is at the very bottom.

Figura 3.31 Muestra la creación de un nuevo campo para otro link o imagen contenido por medio del control de “Añadir nuevo Link”

3.6.2.6 Actualización del boletín.

En algunos casos el boletín llega a cambiar de imagen, aun ya creado. Para ello se realizó la función de actualización de boletín, en ella el usuario puede realizar cambios a la imagen y volverla a subir, así las coordenadas de la imagen siguen siendo las mismas. Muchas veces los cambios son mínimos, tipos de letra, colores, logos, etc. Sin embargo muy poco cambia la posición de los botones en el lugar que los designaron, es por ello que la imagen puede cambiar pero las coordenadas siguen estando donde se asignaron la primera vez, con ello el usuario puede hacer los cambios que desee sin preocuparse de volver a elaborar los links y posicionar las coordenadas de los botones de nuevo.

En la figura 3.32 puede mostrarse el cambio de una imagen a otra mediante el botón que se ilustra en la imagen ya mencionada.

The image shows a screenshot of a web interface for updating a newsletter. The main content area displays a newsletter preview with the following sections:

- Hoy Cinépolis Conjuntos** (No. 35, 23 de febrero de 2012) with a quote: "Conquista tus miedos y te prometo que conquistarás la muerte" by Alejandro Magno.
- 2º International Summit** with a photo of a group of people and text: "Del 15 al 17 de febrero los directores de los países donde tenemos presencia sesionaron en Morelia, con el objetivo de dar seguimiento a la estrategia internacional Cinépolis y alinear los objetivos del negocio."
- Cinépolis será sede del Festival Internacional de Cine en Guadalajara (FIGG27)** with a photo of a person's face and text: "Reconocen mejores prácticas de RH Cinépolis".
- 2012: Año de crecimiento internacional** with a green arrow graphic and text: "Nuestra estrategia internacional se consolida".
- IDE PANZOZO!** with a photo of a person and text: "La Asociación Civil Mexicanos Primero y Cinépolis, presentaron en rueda de prensa el pasado 7 de febrero en la ciudad de México al documental 'De Panzazo!'".

At the bottom of the interface, there is a form for updating the newsletter:

- Nombre:** BoletínAdrixKi
- Imagen Nueva** and **Galería** buttons.
- Url Imagen o Link Externo:** http://www.kaizen-innova.com/qa/hoycinepi
- Asignar Imagen** button.
- Eliminar Boletín** and **Actualizar** buttons.
- Vista Previa** and **Enviar** buttons.

Figura 3.32 Muestra la imagen actual del boletín.



Figura 3.33 Al asignar la nueva imagen en la galería (recuadro rojo) podemos proceder a actualizar el boletín con la nueva imagen (recuadro azul)

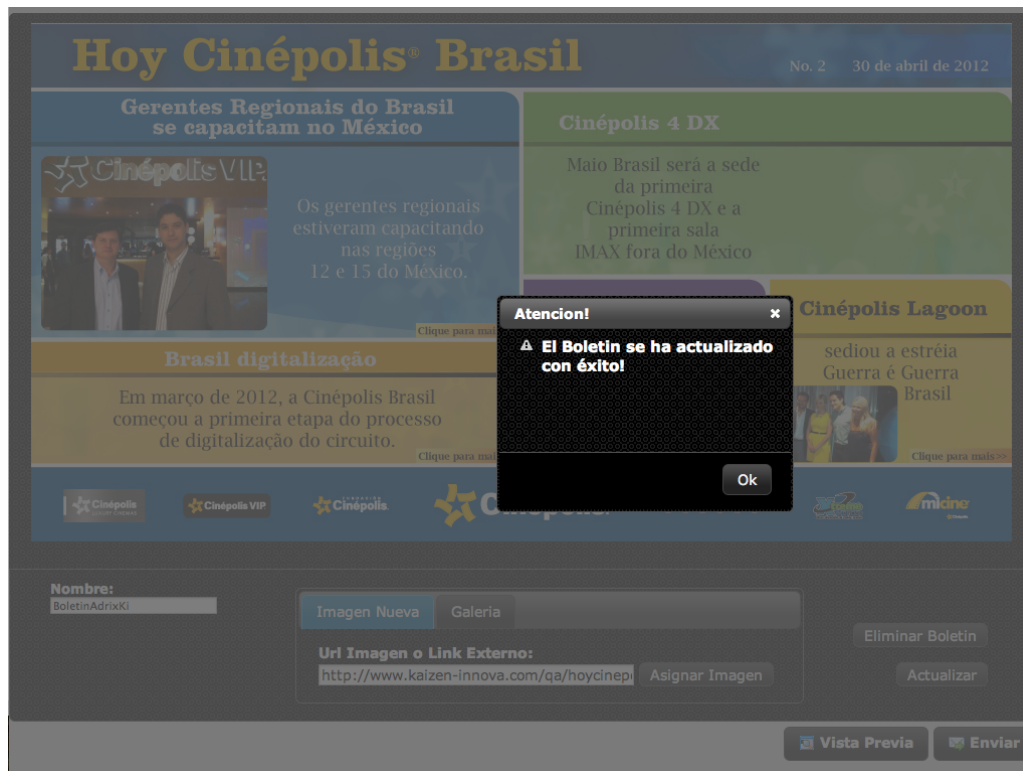


Figura 3.34 Muestra el mensaje de confirmación ya con la imagen actualizada al presionar el botón de “Actualizar”

3.6.2.7 Eliminación de boletín.

Simplemente la eliminación es una función que todos los tipos de operación requieren, sin ser la excepción los boletines también necesitan ser borrados si es que se requiere iniciar de ceros. Para ello la función de Eliminar esta disponible en el sistema. Únicamente lo que se tiene que hacer es dar click en el botón de Eliminar Boletín que se encuentra en la pantalla de edición. El proceso se ilustra en las siguientes figuras:



Figura 3.35 Muestra (marcado con rojo) la opción de Eliminación.

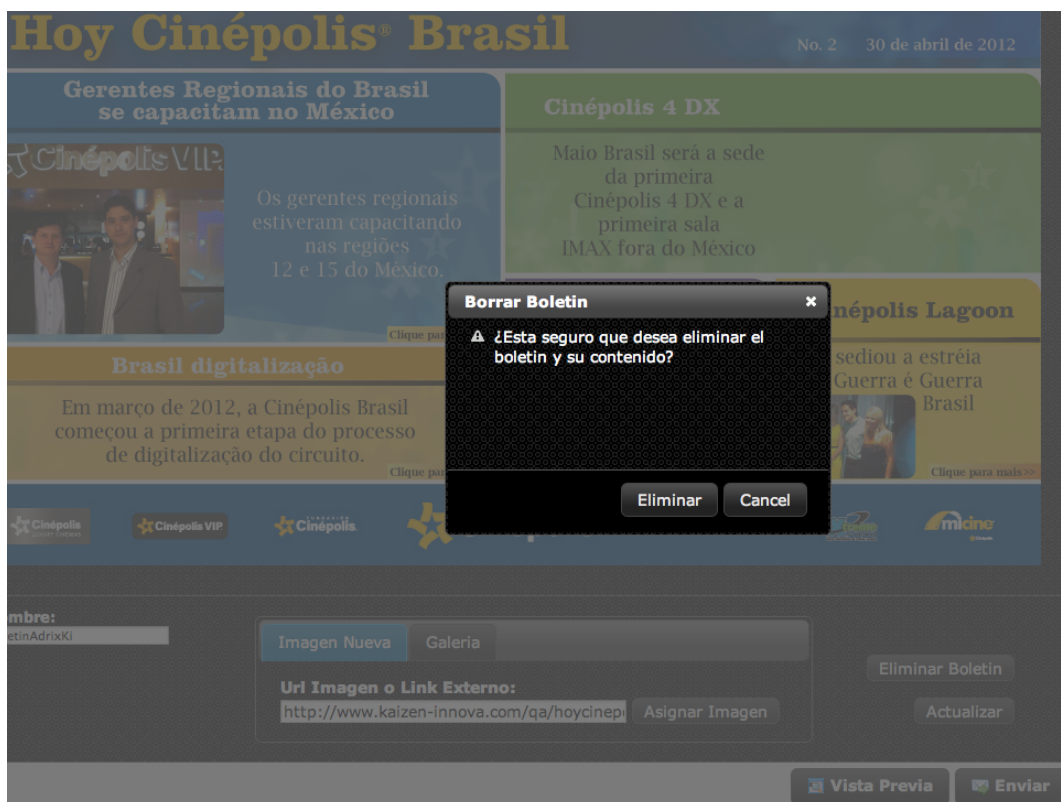


Figura 3.36 Muestra el mensaje de confirmación de eliminación del boletín.

The image shows a web-based editor for a corporate bulletin. At the top, there are dropdown menus for 'Zona: Corporativo' and 'Boletín: 3', with the latter highlighted in a red box. Below this is a header for 'Hoy Cinépolis® Corporativo' with the issue number 'No. 136' and date '6 de marzo de 2012'. The main content area is a grid of article teasers, each with a title, image, and a 'Ver más >>' link. The bottom section features a form for adding content, with fields for 'Nombre:' (containing 'newCORPORATIVO136'), 'Imagen Nueva' and 'Galeria' buttons, 'Url Imagen o Link Externo:' (containing a URL), and 'Asignar Imagen' button. On the right side of this form are buttons for 'Eliminar Boletín' and 'Actualizar'. At the very bottom right, there are buttons for 'Vista Previa' (highlighted in red) and 'Enviar'.

Figura 3.37 Como se observa en el recuadro rojo, el numero de boletín es el anterior del que se estaba editando, ya que al eliminar el boletín actual el sistema carga el anterior o en caso de ser el primero, el sistema deja vacío el espacio.

3.6.2.8 Preview

Dentro de la edición del boletín el usuario quiere ver como es que se visualizaría el boletín con los avances que lleva, por su puesto esto lo puede realizar el sistema mediante la opción de preview que se encuentra en la parte inferior derecha del área donde se realiza la edición del boletín principal, como se muestra en la figura 3.38



Figura 3.38 Indica la opción de Preview (Vista Previa) para la visualización del boletín.



Figura 3.39 Muestra en una ventana por separado la vista previa del boletín.

3.6.2.9 Envío de Email

El cliente lo que pide es tener conocimiento de la elaboración del boletín, el avance que se tiene por parte de la empresa y muchas de las veces el cliente quiere tener la posibilidad de ver cual es el resultado que se tiene actualmente. Para notificarle al cliente sobre estos datos, el sistema permite enviar un e-mail con el boletín integrado en el cuerpo del mensaje.

La figura 3.40 muestra el cuerpo de un mail el cual es un ejemplo de lo que recibiría el cliente en caso de que se le enviara una notificación del avance de su proyecto.



Figura 3.40 Cuerpo del correo electrónico con el boletín.

Como se muestra en la figura anterior, muchas de las bandejas de entrada en algunos servidores de correo electrónico mandan este tipo de correos a correos no deseados, spam ó simplemente no muestran la imagen del boletín; es por ello que se añadió un link en la parte superior el cual indica que si la imagen no puede ser visible el cliente pueda ingresar por medio de un link, esto permite que el cliente pueda observar el boletín por medio del navegador.

El boletín también funciona dentro del mismo cuerpo del correo, sin embargo al ingresar a algún link de contenido, éste abrirá una ventana del explorador predeterminado, mismo que mostrará la pagina con la sección del boletín que abrió en el correo electrónico.

Conclusión

El sistema ha reducido en mas de un 50% del tiempo que se llevaba en la publicación del boletín, en cuestión relación con los usuarios que diseñan, ahora tienen mas organización en los diagramas, ya que tienen identificadas las imágenes que corresponden a cada edición de boletín. El cliente ahora puede tener una idea de los avances que se tienen de su producto antes de ser lanzado, esto beneficia directamente al cliente ya que anteriormente se enviaban solo bocetos sin tener una idea de cómo estaría visualizado en una pagina como boletín. Dentro de la misma implementación se ha visto que también el interés dentro de los mismos usuarios ha sido incrementado en usar tecnología de este tipo.

Posteriormente a la implementación ha habido propuestas de desarrollo dentro de la misma aplicación para hacerla aun mas grande y poder explotar mas el desempeño de la misma, tal es el caso de implementar funciones de envío de correo masivo con los boletines generados, agregar video dentro de los boletines, etc.

Otra de las funcionalidades que se concluyó puede ser potencial en este proyecto fue el poder llevarlo a la red en forma de servicio. No cerrarlo solo a una aplicación sino poder enfocar el proyecto a un software como servicio dentro de un dominio publico de tal manera que pueda ser utilizado por otras empresas. Esto conllevaría a que el desarrollo de la aplicación tendría que ser modificado teniendo una estructura multiempresa con multiusuarios.

Los datos en el desempeño del sistema en la implementación fueron recabados de acuerdo a una entrevista que se le realizó al cliente que solicitó el sistema. Con la aplicación los diseñadores después de realizar los diseños es muy fácil desarrollar el boletín y publicarlo, teniendo en cuenta que no tienen conocimiento técnico de la estructuración HTML y diseño de etiquetas dentro de correo electrónico o en una pagina web, con lo que un usuario sin experiencia en el desarrollo puede publicar el boletín de forma intuitiva.

Bibliografía

- [1]. **MDZ**. Entérate ¿Qué es el cloud computing? . *El Universal*. [En línea] [Citado el: 8 de Junio de 2012.] Artículo obtenido de Wikipedia. <http://www.eluniversal.com.mx/articulos/59620.html>.
- [2]. **Eric Knorr, Galen Gruman**. What cloud computing really means. *InfoWorld*. [En línea] [Citado el: 8 de Junio de 2012.] <http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031>.
- [3]. **Microsoft**. MSDN. *MSDN Microsoft*. [En línea] [Citado el: 11 de Junio de 2012.] <http://msdn.microsoft.com/es-es/library/dd381412>.
- [4]. **Woloski, Matias**. *Aplicaciones de Software como Servicio*. Buenos Aires, Argentina : s.n.
- [5]. **Ryan Asieson, Nathaniel T. Schutta**. *Foundations of Ajax*. 1º Edicion. EUA : Apress, 2006. págs. 1-20.
- [6]. **Gutiérrez, Javier J**. ¿Que es un framework web? *CssBlog*. [En línea] [Citado el: 13 de Junio de 2012.] www.cssblog.es/guias/Framework.pdf.
- [7]. **Kniberg, Henrik**. *Scrum y XP desde las trincheras*. s.l. : InfoQ, 2007. págs. 20-28.
- [8]. **Aroca, Albaro**. Genbeta:Dev. *Mockups, Diseñando Utilidad*. [En línea] [Citado el: 14 de Junio de 2012.] <http://www.genbetadev.com/herramientas/mockups-disenando-usabilidad>.
- [9]. **Camps Paré, Rafael, y otros, y otros**. *Software Libre, Bases de Datos*. Barcelona : UOC, 2005. págs. 10-50.

Índice Figuras

Figura 2.1: ANÁLISIS DEL DOMINIO DE SOFTWARE COMO SERVICIO.	15
Figura 2.2: DIAGRAMA PETICIÓN AJAX	20
Figura 2.1 Características En Los Frameworks.....	25
Figura 2.4: Seccionamiento de Historias de Usuario para el Sprint.	30
Figura 2.5: Diseño de las tablas en una base de datos relacional.	35
Figura 3.1 Diseño del dialogo para iniciar sesión en la aplicación.	43
Figura 3.2 Diseño para el entorno de desarrollo de los diseñadores.....	43
Figura 3.3 Mockup del entorno de la galería de imágenes.....	44
Figura 3.4 Mockup de la parte cliente donde se mostrará el boletín final.....	44
Figura 3.5 Mockup de la parte de la sección de un boletín.	45
Figura 3.6 Muestra las tablas correspondientes a la aplicación de publicación de Boletín.	46
Figura 3.7 Muestra los campos de la tabla IndexGallery que servirán para el almacenado de las imágenes principales del boletín.	47
Figura 3.8 Muestra la llave foránea IdZone en la tabla IndexGallery	47
Figura 3.9 Tabla ContentGallery la cual almacena las imágenes contenido del boletín.....	48
Figura 3.10 Llave foránea hacia IndexGallery que contiene la imagen principal del boletín.	49
Figura 3.11 Muestra la tabla Zone y su Foreign key hacia Path.....	49
Figura 3.12 Tabla de Path que contendrá las rutas de las imágenes por zona.....	50
Figura 3.13 Tabla ExternalLink mostrando la llave foránea hacia ContentGallery ya que está relacionada con la imagen contenido.	50
Figura 3.14 Interfaz Login terminada.....	51
Figura 3.15 Login con error de validación de entrada.	52
Figura 3.16 Diagrama de procesos de la edición para un boletín informativo.	53
Figura 3.17 Galería de Imágenes en la edición del boletín.	54
Figura 3.18 Muestra el botón al cierre de sesión.	55

Figura 3.19	Muestra la plantilla limpia después de presionar el botón de “Nuevo” y habilita el botón de Crear Boletín en la parte inferior derecha. ...	56
Figura 3.20	Mostrando la imagen seleccionada de la galería ya cargada en el área de edición, en el nombre se le puso BoletinAdrixKi.....	57
Figura 3.21	Muestra el mensaje de que el boletín ha sido guardado y creado con éxito.....	57
Figura 3.22	Muestra la interfaz con el boletín creado; del lado derecho el panel de detección de coordenadas y la posibilidad de añadir imágenes contenido o links externos al boletín.....	58
Figura 3.23	Muestra las opciones de Eliminación de boletín, Actualización de boletín, Vista previa y envío de boletín por E-Mail.....	58
Figura 3.24	Panel de imágenes contenido o links externos.....	59
Figura 3.25	Ventana al seleccionar una imagen nueva para la imagen contenido.....	59
Figura 3.26	Muestra la detección de la coordenada en la imagen para indicar donde se creará el link.....	60
Figura 3.27	Mapeo de imagen de 100 x 80 pixeles.....	61
Figura 3.28	Rectángulo con las coordenadas [x1, y1], [x2, y2] para formar el link.....	62
Figura 3.29	URL externa en el link del boletín.....	62
Figura 3.30	Mensaje de confirmación del registro del link al presionar el botón de “Crear Link”	63
Figura 3.31	Muestra la creación de un nuevo campo para otro link o imagen contenido por medio del control de “Añadir nuevo Link”	63
Figura 3.32	Muestra la imagen actual del boletín.....	64
Figura 3.33	Al asignar la nueva imagen en la galería (recuadro rojo) podemos proceder a actualizar el boletín con la nueva imagen (recuadro azul).....	65
Figura 3.34	Muestra el mensaje de confirmación ya con la imagen actualizada al presionar el botón de “Actualizar”.....	66
Figura 3.35	Muestra (marcado con rojo) la opción de Eliminación.....	67
Figura 3.36	Muestra el mensaje de confirmación de eliminación del boletín.....	67

Figura 3.37 Como se observa en el recuadro rojo, el numero de boletín es el anterior del que se estaba editando, ya que al eliminar el boletín actual el sistema carga el anterior o en caso de ser el primero, el sistema deja vacío el espacio.	68
Figura 3.38 Indica la opción de Preview (Vista Previa) para la visualización del boletín.....	69
Figura 3.39 Muestra en una ventana por separado la vista previa del boletín.	69
Figura 3.40 Cuerpo del correo electrónico con el boletín.	70

Índice Tablas

Tabla 2.1 Características En Los Frameworks.....	21
Tabla 2.2 Tabla Comparativa de Metodología Ágil	25

Glosario de Términos

Cloud Computing: *Tendencia a basar las aplicaciones en servicios alojados de forma externa, en la propia web sin la necesidad de instalar elementos en la PC.*

TI: *Siglas que hacen referencia a Tecnologías de la Información*

.NET: *Framework creado por Microsoft para el desarrollo rápido de aplicaciones en la que contiene también herramienta para la creación web.*

MVC: *Patrón de diseño para el desarrollo de aplicaciones que separa en 3 capas los módulos a programar: Datos, Interfaz y Lógica de Negocio.*

Asíncrono: *Propiedad que tiene la transferencia de datos al no depender de una respuesta para procesarse.*