PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

DISERTACIÓN PREVIA A LA OBTECIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS Y COMPUTACIÓN

"IMPLEMENTACIÓN DE UN SISTEMA CMS PARA EL DEPARTAMENTO DE ACCIÓN SOCIAL UNIVERSITARIA DE LA FACULTAD DE INGENIERÍA"

AUTORES:

Luz Marina Unda Ulloa

Edwin Valentin Zapata Ochoa

DIRECTOR:

Ing. Fabián De La Cruz

QUITO, Abril 2013

Tabla de contenido

1. INTRODUCCIÓN 1
2. FUNDAMENTOS TEÓRICOS 2
2.1. CMS 2
2.1.1. Historia 2
2.1.2. Arquitectura 3
2.2. Extreme Programming 7
2.2.1. Metodologías ágiles de programación 8
2.2.2. Extreme Programming 19
2.2.3. Cuadro de entregables 27
3. CARACTERIZACIÓN DEL DEPARTAMENTO DE ACCIÓN SOCIAL PUCE 29
3.1. Motivación 29
3.2. Objetivos 29
3.2.1. General 29
3.2.2. Específicos 29
3.3. Alcance 30
4. DESARROLLO DEL CMS 33
4.1. Requerimientos funcionales 33
4.2. Casos de uso 34
4.2.1. Diagrama general de casos de uso 35
4.2.2. F0: Ingreso al sistema 35
4.2.3. F1: Administrar usuarios 36
4.2.4. F2: Administrar idiomas 38
4.2.5. F3: Administrar sitios 41
4.2.6. F4: Administrar layouts (plantillas) 43
4.2.7. F5: Administrar páginas 45
4.2.8. F6: Administrar secciones 47
4.2.9. F7: Administrar multimedia 50
4.2.10. F8: Administrar Galerías 57
4.2.11. F9: Administrar guestbook 60
4.2.12. F10: Administrar comentarios 63
4.2.13. F11: Seleccionar menú 65
4.2.14. F12: Visualizar página web 66
4.3. Requerimientos no funcionales 67
4.4. Diseño 67

4.4.1. Diagramas de secuencia 68
4.4.2. Diseño de clases 82
4.4.3. Diseño de pruebas 113
4.5. Implementación 131
4.5.1. Herramientas 131
4.5.2. Transformación del modelo de clases Entidad a Entidad-Relación 151
4.5.3. Estándares de codificación 158
4.5.4. Aplicación de pruebas de desarrollo 158
5. PUESTA EN FUNCIONAMIENTO 160
5.1. Implantación 160
5.1.1. Instalación 160
5.1.2. Capacitación 161
5.2. Pruebas 162
5.2.1. Aplicación de pruebas de usuario 162
5.2.2. Ajustes en base a sugerencias del usuario 166
5.2.3. Presentación de la versión final 167
6. CONCLUSIONES Y RECOMENDACIONES 168
6.1. Conclusiones 168
6.2. Recomendaciones 168
BIBLIOGRAFÍA 169

Tabla de figuras

Figura 1: Funcionamiento de un CMS 5
Figura 2: Clasificación de las metodologías Crystal 13
Figura 3: Proceso de desarrollo Crystal 15
Figura 4: Ciclo de vida de FDD 19
Figura 5: Iteración de un proyecto de XP 20
Figura 6: Reglas de un proyecto de XP 21
Figura 7: Costo de los cambios en función del tiempo utilizando metodologías ágiles y metodologías tradicionales de programación 23
Figura 8: Probabilidad de codificar características equivocadas en función del tiempo entre entregas 24
Figura 9: Separación de la lógica de la aplicación de la interfaz gráfica 132
Figura 10: Varios modelos para un mismo objeto real 132
Figura 11: MVC clásico 134
Figura 12: MVC para web 134
Figura 14: MVC Web de Spring 142
Figura 15: MVC Web de Spring aplicado a Grails 142

1. INTRODUCCIÓN

La presente disertación, ***Diseño, Desarrollo e Implementación de un CMS (Content Management System) para diferentes proyectos del departamento de Acción Social Universitaria de la Facultad de Ingeniería***, surge de la necesidad del Departamento de Acción Social de la Facultad de Ingeniería de la PUCE, que necesitaba contar con una herramienta informática que le permitiera crear y mantener sitios web de contenidos dinámicos.

Para responder a este requerimiento, nuestro trabajo contiene 6 capítulos, incluyendo la Introducción. En el *capítulo 2* describimos los conceptos teóricos básicos, explicando inicialmente lo que es un CMS, su historia, su arquitectura y su funcionamiento, y ofreciendo un resumen de las tecnologías ágiles de programación, enfocado en la tecnología seleccionada para nuestro desarrollo, el Extreme Programming. En el *capítulo 3* describimos el Departamento de Acción Social y su necesidad del CMS desarrollado, así como los objetivos y el alcance de la presente disertación. El *capítulo 4* describe las etapas del ciclo de vida del desarrollo del sistema, señalando sus requerimientos funcionales y no funcionales, los casos de uso y los procesos de diseño e implementación del sistema. En el *capítulo 5* explicamos la puesta en funcionamiento del sistema una vez que estuvo terminado: detallamos el proceso de instalación, capacitación y ejecución de pruebas de usuario. Finalmente, en el *capítulo 6* presentamos nuestras conclusiones y recomendaciones.

El desarrollo del sistema que proponemos se realizó utilizando cien por ciento software libre, y aplicando métodos y metodologías aprendidos durante nuestro recorrido universitario. El producto final fue sometido a un meticuloso plan de pruebas para asegurar su calidad.

Confiamos que el trabajo realizado se constituye en un aporte para el Departamento de Acción Social de la Facultad de Ingeniería.

2. FUNDAMENTOS TEÓRICOS

2.1. CMS

Un sistema gestor de contenidos o CMS (del inglés Content Management System) es una aplicación informática que permite crear, editar y publicar contenido a través de diversos formatos, siendo el más común de todos, una página web. La principal utilidad de un CMS es generar páginas dinámicas interactuando con el servidor para presentar el contenido bajo petición del usuario, dicho contenido es extraído de una o varias bases de datos y presentado bajo un formato predeterminado.

2.1.1. Historia

Para entender de donde provienen los gestores de contenido es fundamental tomarse un momento para comprender como surgió la necesidad de una aplicación que se encargue de administrar contenidos susceptibles a cambios continuos. Para lograrlo hay que remontarse a los orígenes del Internet como tal.

La World Wide Web o WWW fue creada por Tim Berners-Lee como un lenguaje sencillo basado en etiquetas. Su propósito principal era crear un estándar para compartir información de forma global. Esto comprendía dos dificultades principales: los usuarios deberían ser capaces de visualizar el contenido sin importar el país en el que se encontrasen y desde cualquier computadora sin importar su marca. Para lograrlo Berners-Lee creó el protocolo HyperText transfer protocol (HTTP), construyó el primer servidor web (CERN httpd) y el primer navegador web (Nexus).

En sus inicios el lenguaje de etiquetas era bastante simple y cumplía con su función eficientemente, pero, para ese entonces, el Internet era una red cerrada y principalmente usada por universidades. El trabajo de mantener y publicar la información era relativamente simple debido a la pequeña cantidad de etiquetas y a que su contenido se conformaba mayormente de documentos formales.

Tras la conversión del Internet a una red pública en los primeros años de la década de los noventa, la cantidad de usuarios aumentó drásticamente, lo cual trajo consigo una mayor exigencia a las capacidades del lenguaje de etiquetas. Se creó una feroz competencia entre nuevos navegadores los cuales intentaban alcanzar estándares más altos en cuanto a la presentación y a la estética de las páginas. El resultado fue una drástica evolución del lenguaje HTML, el número de etiquetas aumentó vertiginosamente y volvió mucho más

complicada la tarea de escribir páginas web que cumplan con los nuevos estándares de presentación de la época. Por esta razón, los desarrolladores se vieron obligados a buscar herramientas para facilitar su trabajo, que solo meses atrás podía ser realizado con un simple editor de texto. Uno de los primeros éxitos fue la herramienta Lotus Notes cuya principal característica fue el incluir la posibilidad de insertar etiquetas XHTML entre las opciones ya existentes para el manejo de documentos. Claramente, esta no era una solución definitiva al problema pero sirvió para demostrar una de las principales cualidades presentes en los gestores de contenido de la actualidad. Lotus Notes fue el primer intento de separar las habilidades de un diseñador web del conocimiento y destrezas de las personas que generan el contenido.

El siguiente momento crucial fue la aparición de los lenguajes de programación que se ejecutan en el servidor, construidos bajo conceptos de tercera generación pero con características específicas para ser utilizados en la web. Estos serían la base sobre la cual se construirían los nuevos sistemas de manejo de contenidos.

Durante los siguientes años una gran cantidad de aplicaciones salieron al mercado y fueron puliendo poco a poco los conceptos básicos de un gestor de contenidos. Uno de los principales avances vino con la publicación de Mambo 4.5, este ofrecía todas las características esenciales que reconocemos en un CMS en la actualidad e implementó por primera vez el uso de plantillas (templates) para dar una presentación atractiva al contenido. Mambo se convertiría en la base para el desarrollo de los modernos gestores de contenido. El 17 de agosto de 2005 nace Joomla como una bifurcación del proyecto Mambo. Por su naturaleza open source y su distribución gratuita rápidamente ganó popularidad y se convirtió en un referente entre los gestores de contenido.

2.1.2. Arquitectura

La estructura de un gestor de contenidos consta de tres partes principales:

- Front-end
- Área de administración
- Plugins

Estos tres componentes forman la columna vertebral de un CMS y se encargan de controlar la base de datos y las interacciones con el navegador web.

2.1.2.1. Front-end

El front-end es la sección pública del CMS, es decir, aquí puede ingresar cualquier usuario sin necesidad de autenticarse. Su función principal es encargarse de la visualización de las páginas web. Para hacerlo debe resolver los urls, procesarlos y presentar al usuario la información solicitada.

En un sitio web tradicional sin la intervención de un CMS, el servidor resuelve las peticiones procesando el url. Su funcionamiento es simple: si el url solicita un archivo que se encuentra dentro de la carpeta del sitio, el servidor lo envía como respuesta a la solicitud, caso contrario envía un error 404 o de archivo no encontrado. En el caso de un gestor de contenido es un poco más complicado, puesto que el url solicita páginas que por lo general no existen físicamente sino como registros en la base de datos.

Debido a que un CMS funciona sobre una base de datos no es necesario crear un archivo físico para cada una de las páginas del sitio, en lugar de eso el gestor de contenidos debe encargase de almacenar los registros necesarios en la base de datos de tal forma que la página web pueda ser reconstruida por el front-end cuando reciba una petición por parte de un usuario.

2.1.2.1.1 Funcionamiento

El front-end trabaja mediante un lenguaje de servidor que se encarga de comunicarse con la base de datos para presentar el contenido solicitado por el usuario (páginas dinámicas). De esta forma se reemplazan las antiguas páginas web estáticas cuyo contenido estaba escrito directamente dentro de las etiquetas HTML.

Por lo general, un CMS separa el diseño gráfico y el contenido de las páginas. Para lo cual el gestor tiene dos opciones: almacenar en la base de datos información con la cual pueda reconstruir gráficamente la página o utilizar plantillas (templates) para presentar de forma agradable el contenido. Sin importar cuál sea el caso el sistema deberá administrar dicha información de forma totalmente independiente del contenido. De esta forma el usuario será capaz de cambiar el aspecto de su sitio web sin que esto afecte a la información presentada en el mismo.

Para resolver una petición, un CMS utiliza un controlador cuyo funcionamiento es muy parecido al de su homólogo en un modelo MVC. Dentro de éste debe haber una porción de código que se encargue de obtener el contenido de la página solicitada y de realizar

cualquier otro tipo de procesamiento necesario para el correcto despliegue de la misma. Una vez que el controlador termine su trabajo y tenga la información lista para se presentada, éste se encarga de mostrarlo a través de una plantilla, o de reconstruir el diseño gráfico de la página en su totalidad si es necesario. La tarea de decidir qué controlador usar para desplegar una página recae sobre el front-end. Para hacerlo, primero revisa los plugins instalados y en caso de no encontrar uno apropiado utilizará el controlador más adecuado según los parámetros que tenga configurado.

En un ejemplo real, si el CMS recibe una petición a través de un url así:

http:/misitio.com/noticias?pag=2

Primero procesa el url para determinar qué /noticias hace referencia al plugin "noticias" y lo inicializa pasando pag=2 como parámetro. El plugin interactúa con la base datos, obtiene toda la información que necesita para desplegar la página y la envía de regreso al CMS que se encarga de presentarla ya sea reconstruyendo el diseño gráfico o utilizando una plantilla.

Para resumir el funcionamiento del front-end, en la Figura 1 se presenta como un proceso ordenado.

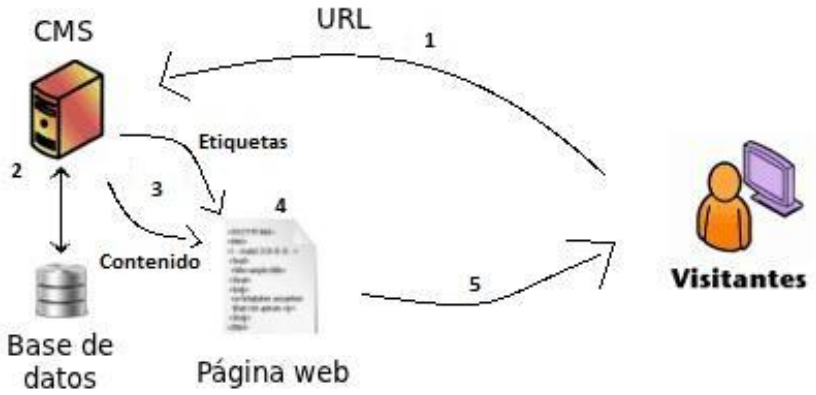


Figura 1: Funcionamiento de un CMS

Autor: Valentin Zapata, Luz Unda

1. El navegador web envía una petición al CMS
2. El CMS descompone la petición y determina sus componentes
3. Si la petición hace referencia a plugins, estos son ejecutados usando los parámetros enviados en caso de haberlos, caso contrario el CMS ejecuta un controlador para resolver la petición.
4. Toda la información procesada por el controlador es enviada al CMS para presentarla a través de una plantilla o reconstruyendo las etiquetas HTML.

5. La página HTML resultante es enviada al navegador web.

2.1.2.2. Área de administración

El área de administración es la sección privada de un CMS. Desde aquí los usuarios pueden realizar tareas de edición a los contenidos o al diseño del sitio web. Por esta razón esta parte del CMS está fuertemente protegida bajo un módulo robusto de seguridades. Existen diferentes formas de presentar el área de administración en un CMS:

- Administración a través de un dominio diferente: en esta opción el administrador tendrá acceso al módulo de edición de contenidos que estará referenciado por un dominio diferente al del sitio público. Lo que se busca en este caso es aislar completamente al área de administración como una medida de seguridad.
- Administración a través de un módulo dentro del mismo sitio: aquí las tareas de edición se realizan mediante un módulo de administración que está situado físicamente dentro de un subdirectorio del sitio web. El control de usuarios se realiza mediante una página donde el administrador ingresa sus datos para validar su ingreso al sistema.
- Administración a través de un programa externo: la administración se realiza mediante un software que se ejecuta externamente del servidor web donde está hospedado el sitio.

En la actualidad lo más común es que los gestores de contenido implementen la administración a través de un módulo dentro del mismo sitio web, debido a que presenta dos grandes ventajas:

- Las actualizaciones se realizan de forma integral, es decir se actualiza el front-end y el área de administración al mismo tiempo ya que están dentro del mismo paquete de software. En los otros casos esto sería imposible puesto que estas dos partes del CMS están totalmente separadas y por tanto deben ser actualizadas por separado.
- Debido a que el área de administración está dentro del mismo sitio, puede ser accedida desde cualquier lugar donde exista conexión a internet.

Sin embargo existen soluciones en el mercado para cualquiera de las tres opciones y la elección de qué método usar es más de carácter personal.

2.1.2.3. Plugins

Aunque no son indispensables en el funcionamiento de un CMS, sin duda alguna le dan un toque de versatilidad a un gestor de contenido. Los plugins se encargan de resolver todo lo que el front-end no pueda hacerlo por sí solo y añaden nuevas opciones o páginas en el área de administración.

Debido a que cada sitio web es diferente, y por tanto requiere diferentes funcionalidades, resulta más eficiente construir un CMS pequeño que cubra las necesidades básicas y agregarle funcionalidad a este núcleo a través de plugins. De esta forma lograremos una estructura modular más fácil de mantener y refinar. A su vez el usuario final también se ve beneficiado al no sobrepoblar el área de administración con decenas de opciones de configuración para funcionalidades que tal vez no necesite.

El funcionamiento de un plugin varía dependiendo del CMS para el cual haya sido programado pero todos se basan en el mismo principio. La mayor parte de plugins funcionan creando nuevos "tipos" de página. En el área de administración el gestor asocia los diferentes tipos a archivos o porciones de código que se encargan de presentar las opciones de administración y edición. En el caso del front-end, cuando recibe una petición por una página con un tipo asociado a un plugin, lo inicializa y lo ejecuta para que éste se encargue de interactuar con la base de datos y reconstruir la página solicitada.

En el caso de un CMS programado en un lenguaje interpretado que trabaja con directorios, el implementar plugins puede ser relativamente sencillo, bastaría con pocas líneas de código y pre establecer un subdirectorio dentro de la aplicación donde el usuario ubique los archivos del plugin. Con un lenguaje compilado la tarea resulta más demandante y enfrenta algunas limitaciones, entre las más notorias podrían estar la necesidad de tener que recompilar el software y de reiniciar el servidor web dependiendo el caso.

2.2. Extreme Programming

Para entender el concepto de Extreme Programming, es necesario entender primero lo que es una metodología ágil de programación. Para lograrlo, se explicará en primer lugar lo que son las metodologías ágiles en general, para después analizar un poco más a fondo algunas de ellas. Finalmente se estudiará el concepto de XP, así como sus herramientas y sus formas de aplicación.

2.2.1. Metodologías ágiles de programación

El desarrollo ágil de software es un concepto, una filosofía y una metodología que evolucionó durante la década de los 90 como respuesta a las frustraciones generadas por los conceptos de ciclo de vida del desarrollo de software tradicional.

2.2.1.1. Inicios

Los primeros procesos de desarrollo de software desarrollados en los años 80 y 90 eran criticados por ser burocráticos y lentos. Todos los intentos de mejorar se enfocaban en mejorar la definición y detalle de los requerimientos, en diseñar arquitecturas que cumplan con éstos, y desarrollar el software de una manera metódica y regimentada. Esta metodología no permitía manejar cambios en los requerimientos ni lograba acelerar la entrega del software.

Fue entonces que a mediados de los 90 un pequeño grupo de desarrolladores empezó a promover metodologías innovativas para desarrollo de software que sean capaces de reaccionar rápidamente y adaptarse a cambios de requerimientos y tecnología.

Con estos antecedentes, en 2001 se reunió un grupo de representantes de diferentes metodologías ágiles, como Extreme Programming, SCRUM, DSDM, Desarrollo de software adaptable, Crystal, Desarrollo impulsado por características, Programación Pragmática, entre otras, y utilizaron por primera vez el término "desarrollo ágil" cuando publicaron el Manifiesto por el Desarrollo Ágil de Software. Este grupo vio la necesidad de una alternativa al desarrollo tradicional, impulsado por documentación. El movimiento ágil busca restaurar la credibilidad de la palabra metodología y restaurar el balance. Acepta el modelado, pero no como una manera de aumentar un diagrama más a un repositorio empolvado. Acepta la utilidad de la documentación, pero no de cientos de páginas que nunca son mantenidas y raramente utilizadas, sino que estima más útil la comunicación cara a cara con el cliente. Planifica, pero reconoce las limitaciones de la planificación en entornos poco estables.

El manifiesto indica que:

> *Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:*
>
> - *Individuos e interacciones sobre procesos y herramientas*

- *Software funcionando sobre documentación extensiva*
- *Colaboración con el cliente sobre negociación contractual*
- *Respuesta ante el cambio sobre seguir un plan*

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

2.2.1.2. Principios

La mayor prioridad es la satisfacción del cliente, entregando continuamente software que le sea útil y de manera rápida. Se debe aceptar los cambios de requisitos en cualquier etapa del desarrollo y aprovecharlos para proporcionar ventajas competitivas. Las entregas deben ser frecuentes, entre dos semanas y dos meses. El equipo se compone de administradores, responsables del negocio, programadores que se auto-organizan y trabajan juntos durante todo el proyecto, en un ambiente propicio al trabajo, con el apoyo y la confianza necesarios para una buena ejecución del trabajo. El equipo debe poder mantener un ritmo constante de trabajo de manera indefinida, promoviendo así el desarrollo sostenible. La comunicación es muy importante, y no hay nada mejor que una comunicación directa, de cara a cara. A intervalos regulares debe reflexionar en los problemas encontrados, las soluciones proporcionadas y como ser más efectivo y eficiente. El progreso se mide principalmente con software funcional. El trabajo debe efectuarse de la manera más simple posible, minimizando el trabajo innecesario.

2.2.1.3. Metodologías

Antes de empezar con el análisis de Extreme Programming, se analizarán otras metodologías ágiles para determinar las razones de utilización de XP sobre las otras.

2.2.1.3.1 SCRUM

Scrum es una metodología de trabajo en equipo para desarrollar productos complejos que ha estado en uso desde el inicio de la década de los 90. No es un proceso o una técnica, sino un conjunto de reglas que se pueden utilizar para diferentes procesos y técnicas. Está basado en el control empírico de los procesos, es decir, que el conocimiento viene de la experiencia y de la toma de decisiones basadas en lo que se conoce.

Se basa en la idea de ir formando un producto por partes, cada parte completando lo que ya se hizo con anterioridad. Trabajar en partes pequeñas fomenta la creatividad y permite que el equipo se adapte a cambios y comentarios del cliente para crear exactamente lo que se necesita. Scrum provee unas pocas reglas simples que permiten crear una estructura para que los equipos de trabajo se enfoquen en resolver una tarea que parecería imposible de otra manera. Se enfoca en la parte humana del trabajo: la necesidad de pertenecer a un grupo, de aprender, de hacer, de crear y ser creativo, de crecer y mejorar, de interactuar con otras personas.

Los roles, artefactos, eventos y reglas de Scrum son inmutables: si bien es posible aplicarlo parcialmente, el resultado no sería Scrum. Puede ser aplicado en conjunto con otras técnicas, metodologías y prácticas.

El proceso fundamental de Scrum divide al grupo de trabajo en tres subgrupos con diferentes roles:

- El *Dueño de producto* (Product Owner) determina lo que se debe hacer en 30 días o menos. Además valoriza el producto y el trabajo del equipo de desarrollo. Es una sola persona que puede representar a un grupo, tiene que ser respetada por todos.
- El *Equipo de desarrollo* (Development team) se compone de profesionales que crean lo que debe estar completado en 30 días o menos y demuestra lo que han hecho. En base a esta demostración el dueño de producto determina lo que se debe hacer después. Es un grupo estructurado que administra y organiza su propio trabajo. Todos en el equipo tienen un mismo status, sin importar el trabajo que realicen. En el equipo tienen que haber los profesionales y expertos necesarios para completar todo el trabajo y no debe dividirse en subgrupos. El grupo debe ser suficientemente pequeño para ser ágil y manejable, pero suficientemente grande para completar el trabajo.
- Los *Maestros Scrum* (Scrum Masters) aseguran que el proceso funcione sin complicaciones y ayudan continuamente a mejorarlo, así como al equipo y al producto. Son los responsables de que el equipo en su totalidad entienda y actúe según las reglas de Scrum. Ayuda a las personas externas al equipo a entender que interacciones son adecuadas y cuáles no con el equipo.

Los tres pilares principales de Scrum son:

- *Transparencia*: Los aspectos importantes del proceso deben ser visibles para todos los responsables del producto final. Tienen que existir estándares para que los observadores puedan entenderlo todo.
- *Inspección*: Deben existir inspectores calificados y eficientes que deben inspeccionar frecuentemente todas las partes del proceso para detectar aspectos indeseados, sin que estas inspecciones lleguen a ser un estorbo al trabajo.
- *Adaptabilidad*: Si un inspector determina que uno o más aspectos del proceso se desvía de los límites aceptables, éstos deben ser ajustados lo más pronto posible para evitar un problema mayor.

Scrum define eventos regulares que permiten minimizar las reuniones. Cada evento dura el tiempo necesario para que la planificación no resulte una pérdida de tiempo y presenta la oportunidad de inspeccionar y adaptar algo. Estos eventos no pueden sobrepasar el tiempo asignado a cada uno de ellos para mejorar la productividad y la eficiencia del equipo.

- *Sprint*: Son los 30 días durante los cuales se debe terminar algo utilizable. Un sprint comienza inmediatamente al terminarse el anterior. Pueden considerarse pequeños proyectos de no más de un mes y tienen que incluir una descripción de la tarea a realizarse y un diseño y plan flexibles que guíen la producción. Puede ser cancelado únicamente por el Dueño de producto cuando su objetivo se ha vuelto obsoleto.
- *Reunión de planificación del Sprint*: El equipo completo colabora en la planificación del Sprint. Una reunión se divide en dos partes iguales. Durante la primera parte se define el trabajo a efectuarse durante el siguiente Sprint, durante la segunda se define su objetivo principal.
- *Scrum diario*: Es un período de 15 minutos durante el cual el equipo de desarrollo sincroniza sus actividades y planifica las siguientes 24 horas. Se inspecciona el trabajo realizado durante el día anterior y se determina lo que debe hacerse durante el día. Se determina el progreso en relación al objetivo. El Scrum diario evita que se creen más reuniones, fomenta la comunicación del equipo, identifica y elimina cualquier impedimento al desarrollo del producto y aumenta el conocimiento general del equipo sobre el producto.
- *Revisión del Sprint:* Es una reunión de cuatro horas al final de cada Sprint durante la cual se inspecciona el producto, se adapta lo que sea necesario y se revisa lo hecho. El

tiempo asignado a esta reunión es proporcional al tiempo de duración del Sprint, por ejemplo, a un Sprint de dos semanas le correspondería una reunión de dos horas. Durante esta reunión el Dueño de producto determina lo que está completado y lo que no; el equipo de desarrollo discute los éxitos del período, los problemas y como se solucionaron, demuestra el trabajo completado y responde a posibles preguntas por parte del resto del equipo Scrum. El equipo completo discute en el camino a seguir, creando una base para la siguiente reunión del planificación de Sprint.

- *Retrospectiva de Sprint*: Es una reunión de tres horas (para un Sprint de un mes, proporcional si el Sprint duró menos) que tiene lugar después de la revisión del Sprint y antes de la planificación del siguiente ciclo. Durante este tiempo el equipo se analiza a sí mismo para planificar mejoras para el siguiente Sprint. Se analiza el éxito del Sprint anterior en relación a personas, relaciones, procesos y herramientas. Se identifican los principales éxitos y las mejoras potenciales y se planifica como implementarlas teniendo en cuenta la manera de trabajo del equipo.

2.2.1.3.2 Familia de metodologías Crystal

Se la llama así porque es un conjunto de metodologías basadas en los principios básicos de Crystal. El objetivo de presentar variaciones de esta metodología es brindar al usuario la posibilidad de escoger la que más se ajuste a la realidad del proyecto que desee atacar. Cada miembro de la familia Crystal está representado por un color que indica su robustez y complejidad, de tal forma que mientras más obscuro sea el color, la metodología a la que representa es más robusta y dirigida a proyectos más grandes y complejos.

Crystal sugiere escoger el color de la metodología a usar, basándose en dos factores principales: el tamaño del proyecto, principalmente representado por el número de integrantes del equipo de trabajo y el riesgo asociado al proyecto (Figura 2). Proyectos grandes suelen necesitar de más coordinación y por tanto es conveniente usar metodologías más robustas, a su vez un proyecto de alto riesgo necesita de una metodología con más medios de control y de revisiones rigurosas. Las letras dentro de cada cuadricula en la figura 2 indican el tipo de pérdidas potenciales en caso de una falla crítica del sistema desarrollado en el proyecto: Confort (C), Cantidades despreciables de dinero (D), Cantidades representativas de dinero (E), vidas humanas (L). En otras palabras, una falla crítica de un sistema de nivel C causa incomodidad en la experiencia del usuario final, mientras que un fallo crítico en un sistema de nivel L literalmente causaría la pérdida de

una vida humana. El número después de cada letra en los casilleros de la Figura 2 representa la cantidad máxima de integrantes del equipo recomendada por Crystal.

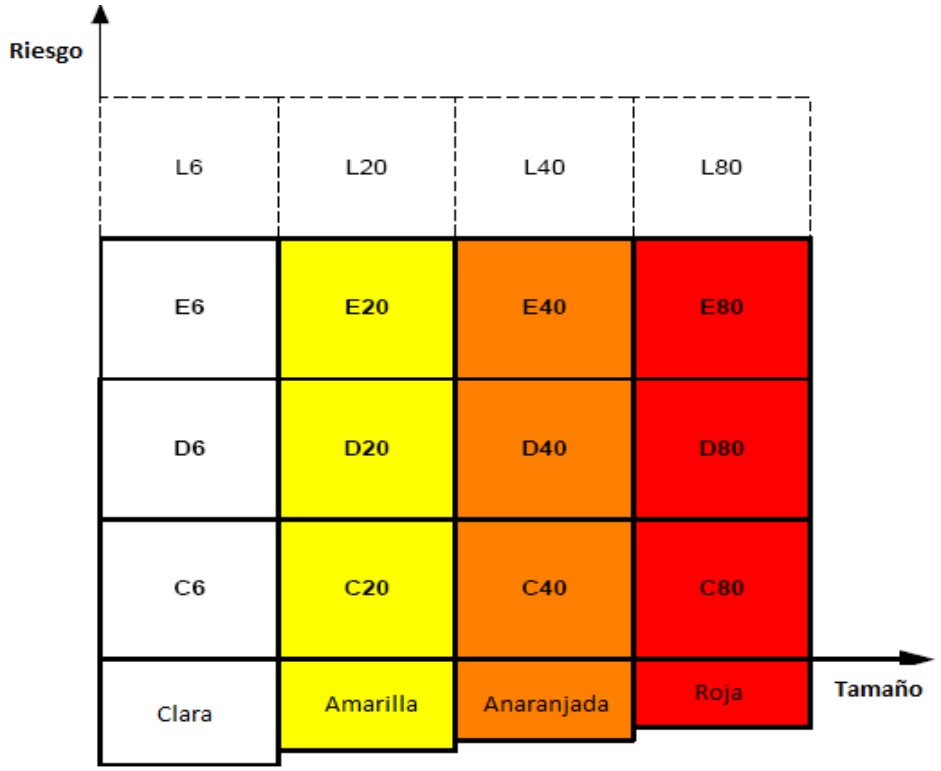


Figura 2: Clasificación de las metodologías Crystal

Autor: Pekka Abrahamsson, Out Salo, Jussi Ronkainen

Fuente: Agile software development methods, 2002

Claro y naranja son las dos metodologías más usadas de la familia Crystal. Por esta razón usaremos estas dos versiones para explicar el estándares, roles y prácticas seguidas en el proceso de desarrollo.

Crystal claro ha sido diseñada para proyectos pequeños o de categoría D6 (Figura 2), que contengan hasta 6 miembros, sin embargo aplicando extensiones al plan de comunicaciones puede ser aplicada en proyectos de categoría E8 y D10 (Figura 2). Para aplicar Crystal claro de forma efectiva el equipo de trabajo debe estar físicamente presente en su totalidad dentro de una misma oficina.

Crystal naranja fue diseñada para proyectos medianos, el equipo de trabajo puede tener entre 10 y 40 miembros (D40) y su duración debe ser de uno a dos años. De la misma forma que Crystal claro, esta metodología puede ser extendida para enfrentar proyectos con hasta 50 miembros (E50) aplicando mejoras al plan de comunicación y al plan de

verificación y pruebas. Crystal naranja separa a los miembros del proyecto en pequeños equipos multifuncionales usando la estrategia de diversidad holística, la cual describiremos más adelante. Esta versión de la familia de metodologías Crystal se enfoca principalmente en cumplir los tiempos de comercialización de los entregables, para logarlo trata de compensar retrasos por posibles cambios en los requerimientos y diseño, limitando el número de entregables por ciclo y manteniendo una comunicación fluida y eficiente entre los equipos de trabajo.

A continuación analizamos las similitudes y diferencias entre claro y naranja en los aspectos principales propuestos por la familia de metodologías Crystal.

- Políticas

Son las prácticas que necesariamente se deben aplicar durante el proceso de desarrollo. Crystal claro y Crystal naranja sugieren las siguientes políticas:

  - Desarrollo incremental
  - Seguimiento del avance del proyecto usando hitos basados en entregas de nuevas versiones o nuevos módulos del software a desarrollar.
  - Involucramiento directo del usuario final
  - Pruebas automatizadas de funcionalidad
  - Dos revisiones del producto por parte del cliente por ciclo
  - Talleres para puesta a punto del producto y la metodología al inicio y a la mitad de cada ciclo

La única diferencia en cuantos a las políticas entre ambas metodologías es que Crystal claro sugiere un tiempo máximo de duración de cada ciclo de tres meses, mientras que Crystal anaranjado permite incrementar la duración del ciclo de desarrollo hasta a cuatro meses.

- Prácticas

En esta sección se describen las prácticas sugeridas por Crystal, para cada ciclo de desarrollo incremental. La Figura 3 ilustra su funcionamiento:

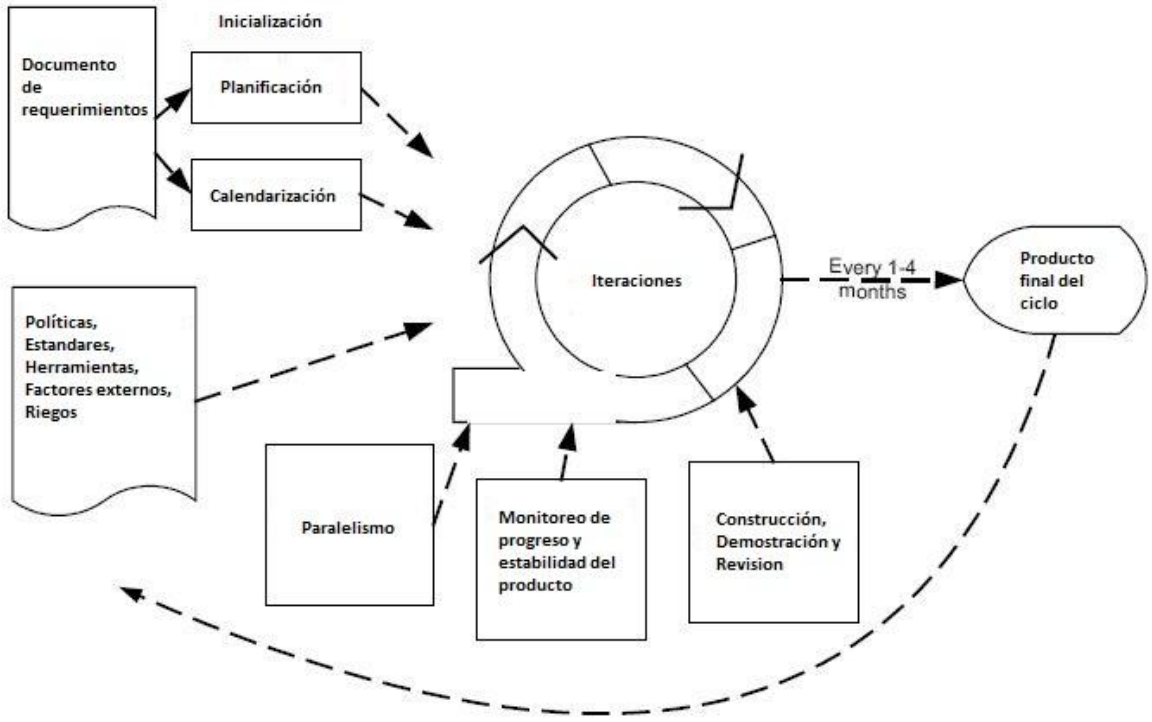


Figura 3: Proceso de desarrollo Crystal

Autor: Pekka Abrahamsson, Out Salo, Jussi Ronkainen

Fuente: Agile software development methods, 2002

- Inicialización

En esta etapa se realiza la planificación y calendarización de las actividades para el ciclo. El equipo selecciona que requerimientos implementar durante el ciclo, teniendo en cuenta las limitaciones de tiempo y la complejidad de los mismos, de tal forma que todos los miembros del equipo se sientan cómodos y se comprometan a cumplir con lo calendarizado. Al concluir esta fase, el equipo debe contar con un calendario específico cuyo fin sea la entrega de una versión totalmente funcional y lista para el usuario dentro de los siguientes tres o cuatro meses.

- Revisión

En esta etapa el equipo se encarga de revisar los objetivos planteados para el ciclo incremental de desarrollo. De ser necesario se plantean soluciones y correctivos para compensar retrasos en el calendario.

- Monitoreo

El proceso de desarrollo es monitoreado en dos aspectos principales: progreso y estabilidad. El progreso es medido basándose en el complimiento de hitos, mientras que la estabilidad se refiere a la frecuencia de cambios durante el proceso de diseño.

- Paralelismo

Una vez que el monitoreo de la estabilidad arroje el resultado "Suficientemente estable para revisión", los equipos pueden proceder con total paralelismo, es decir, cada uno de ellos se encargan de revisar la documentación de requerimientos y diseño del componente que hayan sido asignados para desarrollar, para posteriormente codificarlo.

- Estrategia de diversidad holística

Crystal sugiere dividir grandes grupos de trabajadores en pequeños equipos multifuncionales. La idea principal es incluir diferentes especializaciones en un mismo grupo. Esta estrategia también permite crear grupos pequeños con conocimiento especial, y considera temas como ubicación de los equipos, comunicación, documentación y coordinación entre equipos.

- Técnica de afinamiento de metodologías

Esta es una de las técnicas base de Crystal Claro y Naranja. Utiliza entrevistas y talleres de equipo para determinar una metodología Crystal adecuada a cada proyecto específico. Una de las principales ideas del desarrollo incremental es permitir arreglar o mejorar el proceso de desarrollo. En cada iteración el grupo aprende y utiliza el conocimiento ganado en la iteración anterior para mejorar el proceso de desarrollo durante la siguiente.

- Visualizaciones del cliente

En Crystal Claro se recomienda dos visualizaciones del producto por parte del cliente en una iteración. Crystal Naranja recomienda tres por ciclo.

- Talleres de reflexión

Tanto Crystal Claro como Naranja recomiendan que los equipos efectúen talleres de reflexión antes y después de cada iteración

Sin embargo, ni Crystal Claro ni Crystal Naranja definen prácticas ni técnicas específicas para el desarrollo de software, sino que permiten adoptar prácticas de otras metodologías como Scrum o XP para reemplazar o complementar las de Crystal.

2.2.1.3.3 Desarrollo impulsado por características

El desarrollo impulsado por características (FDD) es una de las metodologías ágiles menos conocidas y mencionadas en los libros de desarrollo ágil de software. Fue inventado por Jeff De Luca e introducido por primera vez en 1999 en el libro "Java Modeling in Color with UML" (Modelamiento Java a Color con UML), presentando una combinación del proceso seguido en la compañía de De Luca y de los conceptos de características de Peter Coad.

El FDD es un acercamiento adaptativo y ágil para el desarrollo de software, centrado en el cliente y en la arquitectura que puede aplicarse en equipos pequeños, pero fue específicamente diseñado para equipos grandes. Un equipo pequeño de desarrolladores disciplinados y hábiles tiene muchas posibilidades de tener éxito sin importar la metodología ágil seleccionada. Sin embargo es poco realista esperar que en un equipo grande todos sean igual de hábiles y disciplinados.

Esta metodología no abarca todo el proceso de desarrollo sino que se enfoca en las fases de diseño y construcción. Fue diseñado para funcionar junto con otras actividades del desarrollo de software y no requiere la utilización de un modelo específico. Utiliza un desarrollo iterativo con las mejores y más efectivas prácticas de la industria. Provee métodos, técnicas y guías para entregar un sistema y también incluye roles, elementos, objetivos y rangos temporales necesarios para el proyecto.

Las características son el aspecto más importante del FDD. Una característica se define como una pequeña función importante para el cliente y se expresa en la forma de: *<acción> un <resultado> <para | por | de> un <objeto>*, donde el objeto es una persona, un lugar o una cosa (incluyendo roles, momentos, intervalos de tiempo, descripciones,…). Por ejemplo, *calcular el total de una venta*, o *calcular el total de compras de un cliente*. El cliente sabe exactamente lo que es, puede darle una prioridad, puede definir lo que es necesario y lo que no, y puede determinar si es realmente necesario o no. Son el equivalente de los casos de uso de RUP y de las historias de usuario de XP: la principal fuente de requerimientos y la principal contribución para la planificación. Son pequeños bloques para planificar, reportar y rastrear que se pueden entender, medir y completar en ciclos de dos semanas. Un grupo de características es la agrupación de características relacionadas con el negocio. Un grupo se nombra con la forma *<acción infinitiva> un <objeto>*, por ejemplo, *Vender un producto*. Los grupos principales se nombran como

Administración <objeto>, por ejemplo, *Administración de ventas*. Se hace un listado general de características mientras se crea el modelo general. Este listado se aumenta con características encontradas en documentos. Se hace una lista más detallada después de haber desarrollado el modelo general.

El FDD consiste en 5 actividades secuenciales e iterativas (ver Figura 4)

- Desarrollar un modelo general: el resultado inicial es un modelo de alto nivel y notas. El iniciar un proyecto el objetivo es identificar y entender los fundamentos del sistema a desarrollar. Durante el proceso de desarrollo se construye sobre este modelo, que refleja lo que se está creando.
- Crear una lista de características, agrupando las que se relacionan
- Planificar por característica: el resultado final es el desarrollo en sí, la identificación de los dueños de las clases y la identificación de los dueños de los grupos de características
- Diseñar por característica
- Desarrollar por característica

Estas últimas dos actividades representan la mayor parte del esfuerzo de un proyecto de FDD, cerca del 75%. Incluyen modelamiento, programación, pruebas, y empaquetamiento del sistema.

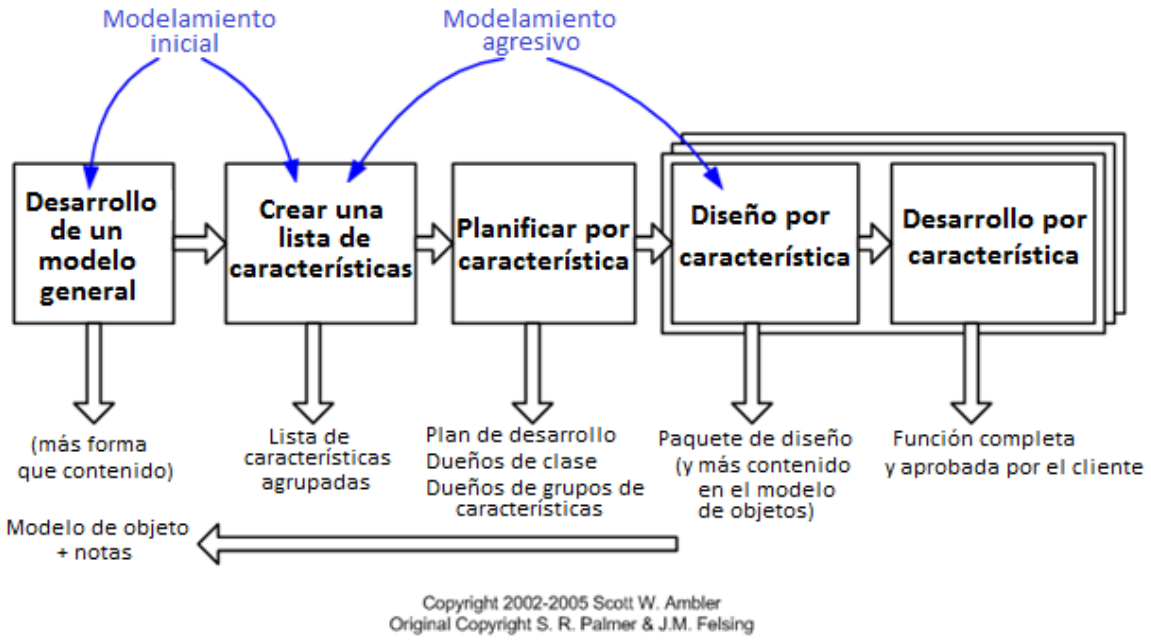


Figura 4: Ciclo de vida de FDD

Autor: R. Palmer y J.M. Felsing

Fuente: http://www.agilemodeling.com/essays/fdd.htm, última visualización el 13-10-2012

2.2.2. Extreme Programming

El análisis de Extreme Programming comenzará con una breve descripción, seguida de sus principales reglas y valores, para finalizar con algunas de las razones de su utilización.

2.2.2.1. ¿Qué es Extreme Programming?

Extreme Programming (Programación Extrema en inglés, también llamada XP) es un proceso ágil de programación utilizado en todo el mundo.

Se basa en la satisfacción del cliente, entregando el producto constantemente, a medida que el cliente lo necesita.

Es una manera de trabajar en armonía con un equipo, con valores personales e institucionales. Los equipos comprenden administradores, programadores, clientes que colaboran para implementar entornos simples pero efectivos que permitan ser altamente productivos.

La Figura 5 muestra cuál es el proceso a seguir en cada una de las iteraciones de un proyecto de XP.

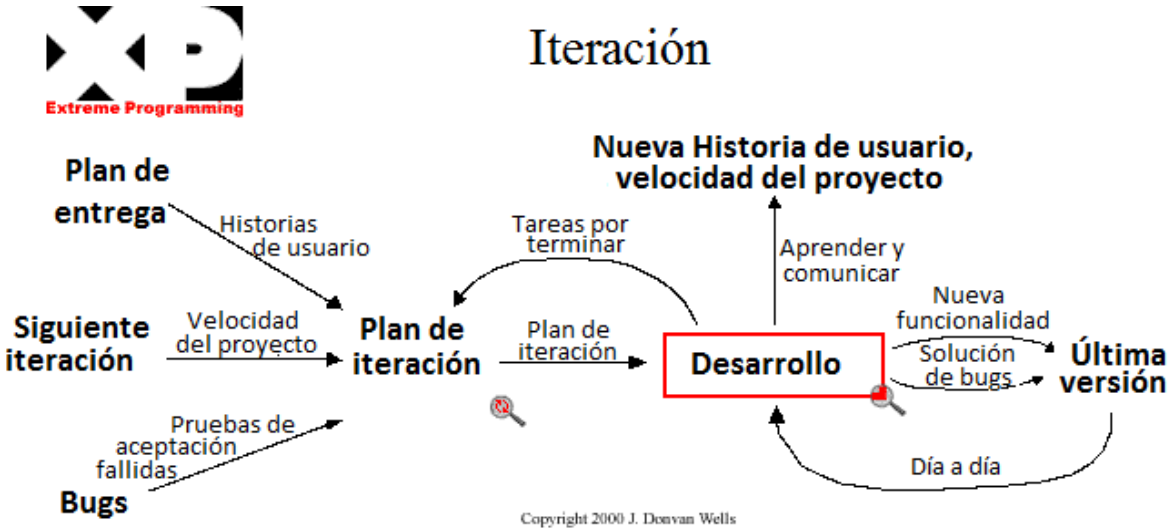


Figura 5: Iteración de un proyecto de XP

Autor: J. Donovan Wells

Fuente: http://www.extremeprogramming.org/rules/iterative.html, última visualización el 13-10-2012

2.2.2.2. Reglas de XP

Planificación: Se escriben los requerimientos del cliente y se planifican entregas frecuentes. Al inicio del proyecto se hace la planificación de publicación, cuando el cliente propone un plan inicial que permita al equipo comenzar a planificar y tomar decisiones. Este plan inicial será revisado y modificado regularmente. A continuación se divide al proyecto en iteraciones de aproximadamente dos semanas. Al inicio de cada iteración se hace una planificación que permite guiar al equipo. El cliente presenta sus requerimientos para las siguientes dos semanas y los programadores crean tareas y estiman costos para entregar software funcional al final del período.

Administración: El equipo debe disponer de un área dedicada al trabajo, abierta y tranquila. Se debe fijar un ritmo de trabajo que todo el equipo pueda mantener durante todo el proyecto. Se deben efectuar reuniones diarias para comunicar problemas, ideas, etc. y medir la velocidad a la que está avanzando el proyecto. Todo el equipo debe tener conocimiento de todo el proyecto. Se pueden modificar las reglas de XP adaptándolas mejor a la realidad del proyecto o del equipo.

Diseño: Debe mantenerse simple, de preferencia escogiendo una metáfora para poder explicar el proyecto a cualquiera que lo necesite, sin tener que recurrir a documentación complicada. Se recomienda utilizar tarjetas CRC (Clase, Responsabilidad, Colaboración: tarjetas que representan objetos, con el nombre de la clase en la parte superior, y con las

responsabilidades y clases colaborativas en la parte inferior) y se debe mejorar el código siempre que sea posible.

Desarrollo: Se deben establecer estándares de codificación que todos deben cumplir, codificando las pruebas unitarias en primer lugar. Todo el código se debe programar en parejas, lo que permite encontrar errores y corregirlos de manera más efectiva. Todo el código se debe integrar frecuentemente, por una sola pareja a la vez y en una computadora dedicada a esto. El código debe ser propiedad de todos los desarrolladores, para que cualquiera pueda cambiar o mejorar cualquier parte del código, aportar ideas o sugerencias. El cliente debe formar parte del equipo para ayudar en el proceso de planificación, despejar dudas, etc.

Pruebas: Deben existir pruebas unitarias para todo el código y para todo error detectado, que deben ser aprobadas para que éste pueda ser publicado.

La Figura 6 muestra la manera en la que las reglas de XP funcionan juntas para ofrecer al cliente la oportunidad de participar en el proceso de desarrollo del software, para que los desarrolladores contribuyan activamente sin importar su nivel de experiencia, y para que los administradores se concentren en las relaciones y la comunicación.

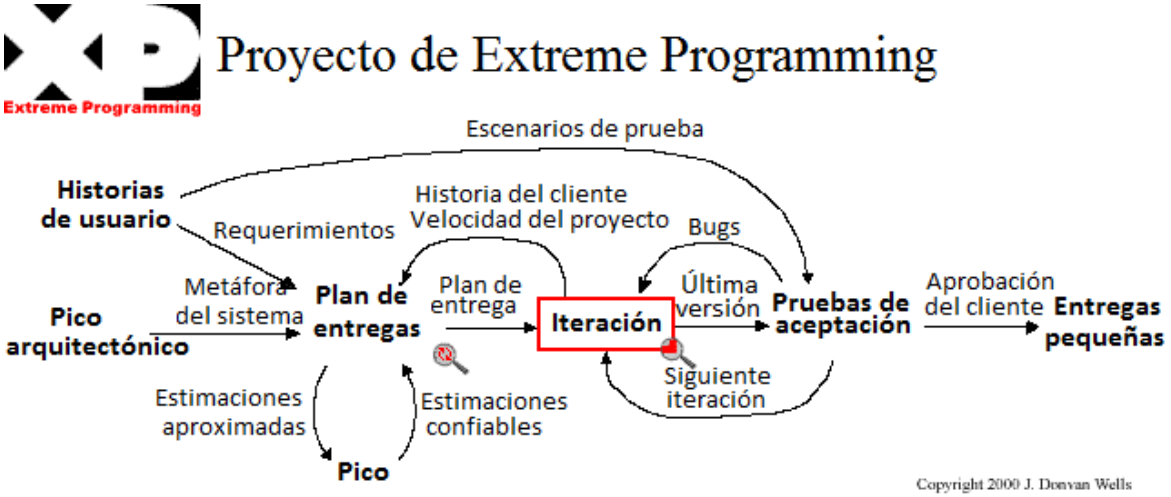


Figura 6: Reglas de un proyecto de XP

Autor: J. Donovan Wells

Fuente: http://www.extremeprogramming.org/map/project.html, última visualización el 13-10-2012

2.2.2.3. Valores de XP

Extreme Programming se basa en cuatro principales valores: comunicación, simplicidad, retro alimentación, respeto y coraje.

2.2.2.3.1 Comunicación

Los programadores deben estar en constante comunicación con los clientes y con el resto del equipo. Debe existir comunicación diaria y se debe trabajar en equipo para cumplir los requerimientos. El código es un medio de comunicación entre programadores, por lo que debe ser lo más simple posible. Si una parte es compleja debe ser simplificada lo más posible para que sea perfectamente entendible. Aunque escribir comentarios en el código es una buena manera de comunicar lo que hace una parte determinada, los comentarios suelen quedar obsoletos después de unos cuantos cambios, por eso es mejor que el código sea auto-documentado, es decir, que se entienda fácilmente lo que hace sin tener que recurrir a comentarios u otras explicaciones. En esto también ayuda tener convenciones de programación, como nombres de métodos y variables. Los programadores se comunican entre sí constantemente ya que programan en parejas. Programar en pareja significa dos programadores en una misma computadora efectuando todas las tareas, compartiendo teclado, mouse y CPU. Uno escribe mientras el otro analiza posibles errores de diseño, ofrece sugerencias de pruebas, ayuda a validar ideas, ayuda a revisar el código. Se intercambian posiciones con frecuencia para que no haya un observador pasivo y un programador fijos.

2.2.2.3.2 Simplicidad

El principio de la simplicidad debe aplicarse en todo lo que se hace. El diseño tiene que mantenerse simple. Se debe hacer únicamente lo necesario, nada más, dando pequeños pasos hacia el objetivo final y corrigiendo los errores cuando ocurren. Esto permite evitar pérdidas de tiempo programando características sofisticadas que el cliente no quiere o no necesita. Un código simple es más fácil de entender, por lo tanto más fácil de mantener. Adoptar técnicas simples de comunicar los requerimientos y monitorear el progreso facilita al equipo seguir el proceso. Además, crear soluciones simples a los problemas minimiza el costo de los cambios a través del tiempo (ver Figura 7).

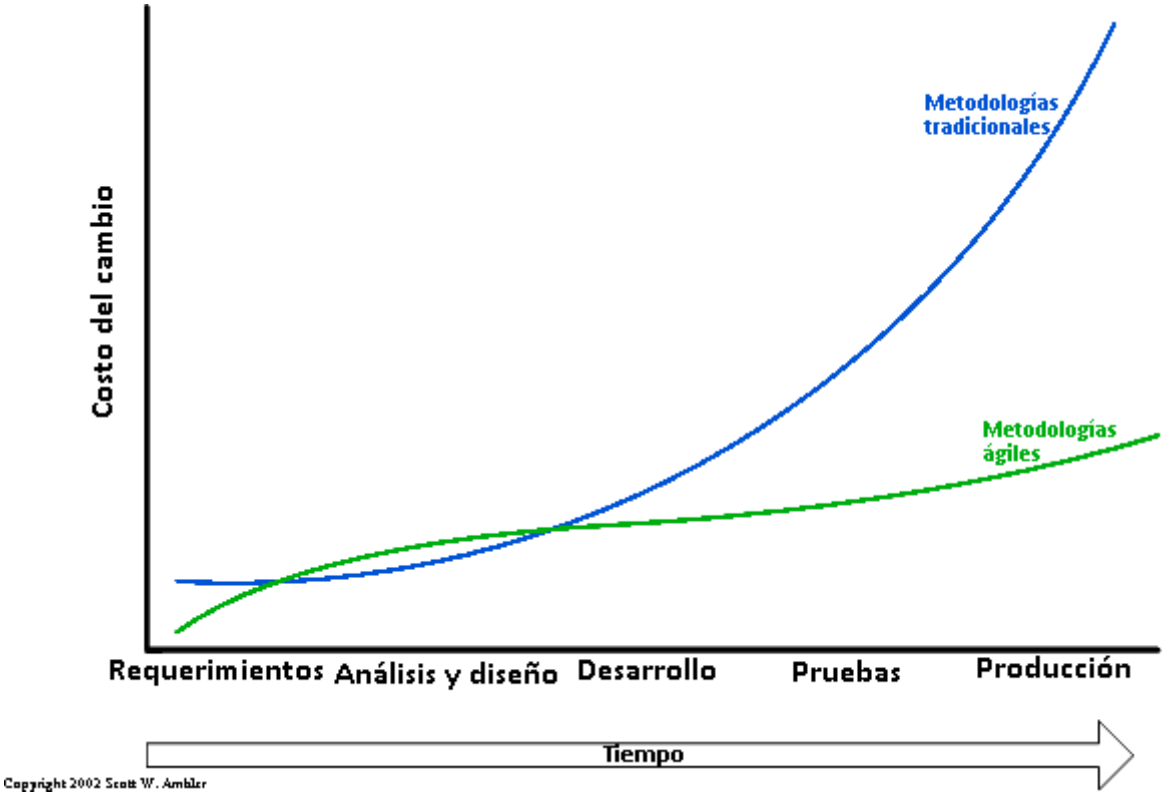


Figura 7: Costo de los cambios en función del tiempo utilizando metodologías ágiles y metodologías tradicionales de programación

Autor: Scott W. Ambler

Fuente: http://www.agilemodeling.com/essays/costOfChange.htm, última visualización el 13-10-2012

Utilizando un acercamiento tradicional a la programación, el costo de un cambio aumenta significativamente conforme se avanza en el ciclo de vida del software. Por ejemplo, corregir un error de requerimiento durante la fase de diseño es diez veces más difícil, mientras que si se está en la fase de programación este valor aumenta a 100. Esto es así porque la cantidad de código aumenta cada vez más. Si no se utilizó un diseño simple, un cambio puede afectar a muchas otras partes. Además conforme pasa el tiempo varios programadores pueden haber trabajado sobre la porción de código a corregir, volviendo cada vez más complicado y dificultando la comprensión para el encargado del cambio.

XP reconoce que los cambios tienen un costo y una dificultad, pero también entiendo que son inevitables. Se debe hacer recodificaciones frecuentes para ir mejorando el código y manteniéndolo lo más simple posible para evitar que con cada cambio se complique. El hecho de hacer pruebas para todas las características también ayuda a hacer cambios en cualquier momento del desarrollo, ya que no se debe hacer pruebas manuales para verificar que el cambio efectuado no haya afectado a otras partes del sistema. Otra característica de

XP que reduce el costo de los cambios es el hecho de que todos los programadores conocen el código: todos los miembros del equipo tienen el mismo nivel de conocimiento de todo el código por lo que cualquiera de ellos puede hacer un cambio, sin tener que analizar el código en primer lugar.

2.2.2.3.3 Retro alimentación

Se entrega software funcional al final de cada iteración. Tener un cliente in-situ ayuda a tener una retroalimentación inmediata sobre el estado del proyecto. Esto permite mostrar el software regularmente, hacer los cambios necesarios y adaptarlo según sea necesario. Si se muestra el software con algunos meses de separación cada vez, el cliente puede cambiar de opinión o puede no gustarle lo que se ha hecho, teniendo así que desechar varios meses de trabajo.

Con iteraciones cortas el cliente puede evaluar cada característica mientras está siendo desarrollada, minimizando así las probabilidades de tener que repetirlas o de entregar características equivocadas (ver Figura 8).El cliente tiene que definir qué características le parecen más importantes para programarlas en orden de prioridad. De esta manera el cliente puede cancelar un proyecto sabiendo cuales características fueron desarrolladas y cuáles no.

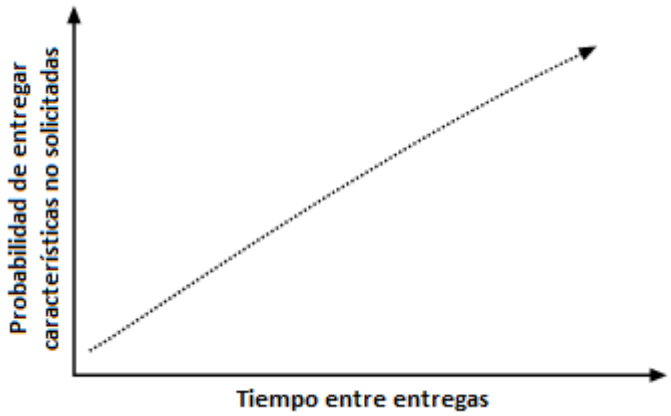


Figura 8: Probabilidad de codificar características equivocadas en función del tiempo entre entregas

Autor: Eric M. Burke, Brian M. Coyner

Fuente: Java™ Extreme Programming Cookbook, Marzo 2003

El código también puede ofrecer una valiosa retro alimentación para los programadores. Las pruebas constantes indican inmediatamente la calidad del código. Después de cada

cambio mayor se deben correr todas las pruebas. Una prueba fallida indica inmediatamente que un cambio reciente dañó una parte específica del sistema.

2.2.2.3.4 Valor

Se necesita valor para aplicar estos principios, aunque parezcan simples y de sentido común. Para comenzar, el concepto de programación en parejas tiene tendencia a ser difícil de aceptar, sobre todo por la administración, ya que sienten que al haber solo la mitad de los programadores trabajando a la vez se va a reducir el rendimiento. Se necesita valor para confiar en que ayuda a mejorar la calidad y a reducir los tiempos y costos de cambios.

La simplicidad es lo que en general más les cuesta a los programadores. Se necesita valor para implementar la característica que necesita el cliente y utilizar una aproximación simple, ya que existe la tendencia de querer ser flexible para características futuras. Sin embargo esto debe evitarse para que el cliente pueda ver el desarrollo cada poco tiempo y no tener que esperar meses para ver el progreso.

2.2.2.4. Cuadro comparativo de las metodologías

Metodología	**Características**	
Scrum	Diseñada para productos complejos Equipos de desarrollo pequeños, auto organizados e independientes Desarrollo incremental en ciclos de 30 días.	Se va formando el producto por partes, cada parte completando lo que ya se hizo con anterioridad. Asigna roles a los miembros del equipo
Crystal	Familia de métodos, todos basados en los mismos principios. Sus técnicas, roles, herramientas y estándares pueden variar Desarrollo en paralelo Desarrollo incremental	Se puede seleccionar el método más apropiado al tamaño del equipo y al riesgo del proyecto Pruebas automatizadas de funcionalidad Permite adoptar prácticas de otras metodologías como Scrum o XP para reemplazar o complementar las de Crystal.
FDD	Proceso de 5 pasos desarrollo de componentes orientados a objetos. Iteraciones muy cortas (entre varias horas y 2 semanas) Modelamiento de objetos Simplicidad del método	Diseño e implementación basados en características Diseñada para equipos grandes No abarca todo el proceso de desarrollo sino que se enfoca en las fases de diseño y construcción.
XP	Desarrollo con alta participación del cliente. Equipos de trabajo pequeños. Versionamiento diario Programación en parejas Basado en el principio de simplicidad	Refactoring: mejoras constantes del desempeño y respuesta a cambios Retroalimentación inmediata por parte del cliente

2.2.2.5. Java y XP

Extreme Programming es un conjunto de principios y recomendaciones independiente del lenguaje de programación que se va a utilizar, y funciona muy bien con diferentes tecnologías. Sin embargo, las características de Java se adaptan muy bien a este tipo de desarrollo de software.

En primer lugar, XP se centra en desarrollar una prueba, compilar el código, ver la prueba fallar, escribir el código necesario para pasar la prueba, y ver la prueba ejecutarse correctamente. Esto hace que se compile y recompile código con mucha frecuencia. Java compila el código de manera rápida, lo que hace que estas compilaciones sucesivas no representen una pérdida de tiempo inútil. Groovy, al estar basado en Java, tiene igualmente

tiempos de compilación muy cortos. Además, al utilizar Grails, estos tiempos de compilación se reducen aún más, ya que una vez que se ejecuta la aplicación, Grails compila de manera automática los cambios.

En segundo lugar, existen numerosas herramientas de pruebas unitarias e integración continua. Junit, compatible también con Groovy/Grails, es un framework ligero para escribir pruebas unitarias. Viene incluido con Grails y genera automáticamente las pruebas iniciales.

Finalmente, la sintaxis de Java, y más aún de Groovy, es fácil de entender lo cual facilita el mantenimiento del código por parte de los miembros del equipo.

Todas estas características hacen que utilizar Extreme Programming para desarrollar con Java sea la elección de metodología ágil más adecuada

2.2.3. Cuadro de entregables

Fase	Entregable	Contenido
Lanzamiento	Justificación	Texto explicativo de los motivos y antecedentes que llevaron a las realización del presente proyecto
	Objetivos	Objetivos del desarrollo del sistema
Requerimientos	Requerimientos funcionales	Especificación de requerimientos funcionales. Diagramas de casos de uso
	Requerimientos no funcionales	Especificación de requerimientos no funcionales
Diseño	Diagramas de secuencia	Diagramas de secuencia para cada caso uso
	Especificación de clases UI	Diseño de las pantallas Cuadro con especificaciones de las pantallas
	Cuadro de clases controlador	Cuadro con especificaciones de las clases controlador
	Diagrama de clases entidad	Diagrama de clases entidad Diccionario de datos de clases entidad
	Plan de pruebas	Diseño de los formularios para las pruebas de desarrollo y de usuario final
Implementación	Selección de herramientas	Descripción de las herramientas seleccionadas Cuadro resumen con las características principales de las herramientas

	Reglas para la transformación de clases entidad a entidad-relación	Listado de reglas a seguir en el proceso de transformación de las clases entidad a un modelo entidad-relación
	Diagrama entidad-relación	Diagrama entidad-relación con campos y tipos de datos
	Diccionario de datos	Cuadros explicativos de cada tabla y campo del diagrama de entidad-relación
Pruebas	Pruebas de desarrollo	Formularios de pruebas de desarrollo llenados
Implantación	Plan de implantación	Resumen del proceso de implantación Cuadro con las actividades realizadas y sus responsables
	Pruebas de usuario	Formularios llenos con los resultados de las pruebas de usuario final
	Capacitación	Resumen del proceso de capacitación

3. CARACTERIZACIÓN DEL DEPARTAMENTO DE ACCIÓN SOCIAL PUCE

3.1. Motivación

Ante el innegable crecimiento de la Internet como método de difusión masivo en nuestro país y el mundo, resulta imperativo que toda institución o fundación se promocione o se dé a conocer a través de un sitio web.

Actualmente la mayoría de las fundaciones que tienen un convenio con el departamento de acción social de la Universidad no poseen un sitio web, debido en gran parte a falta de recursos para cubrir los gastos que implica pagar por el diseño y desarrollo o por falta de asistencia técnico-tecnológica. Por otra parte las pocas fundaciones que cuentan con un sitio web no lo pueden personalizar a su gusto, es decir, no pueden cambiar los contenidos y mucho menos cambiar el diseño de la página.

Durante años la respuesta de la Universidad a este problema ha sido asignar pequeños proyectos para crear sitios web poco o nada personalizables para las fundaciones que lo requieran. En el caso específico del departamento de acción social de la Facultad Ingeniería, el departamento cuenta con un sitio web totalmente estático que no satisface las necesidades de promocionar sus eventos, dar a conocer sus actividades y publicar artículos y noticias sobre el día a día de su destacable labor.

Por todos estos motivos decidimos proponer el desarrollo de un manejador de contenido integral que permita construir sitios web desde cero, de forma rápida y prestando particular atención a la interfaz de usuario para disminuir lo más posible la cantidad de conocimientos específicos en el área de desarrollo necesarios para usar el sistema.

3.2. Objetivos

3.2.1. General

Desarrollar un sistema manejador de contenido (CMS) que permita diseñar y editar páginas web, siguiendo el ciclo estándar de desarrollo de software, para que el departamento de acción social de la Facultad de Ingeniería de la PUCE los implemente en los diferentes proyectos con los que trabaja.

3.2.2. Específicos

a) Diseñar detalladamente el sistema manejador de contenido.
b) Construir la base del programa en una plataforma Java

c) Diseñar y construir una interfaz gráfica amigable e intuitiva para facilitar el uso del sistema
d) Desarrollar un módulo de seguridad robusto que garantice la integridad del sitio web.
e) Realizar la documentación técnica y de usuario
f) Realizar las pruebas necesarias para asegurar la completa funcionalidad del CMS

3.3. Alcance

El presente proyecto contiene el diseño, desarrollo e implementación de un Sistema Manejador de Contenidos que permita crear y mantener sitios web con contenidos editables, en varios idiomas, y aplicables a diferentes proyectos del departamento de acción social universitaria de la Facultad de Ingeniería de la PUCE.

Este sistema incluirá:

- Administración de usuarios: En este módulo el sistema permitirá crear, editar, activar y desactivar usuarios. El gestor de contenidos soportara dos tipos de usuarios: un administrador que tendrá acceso a todas las configuraciones generales del sistema y podrá ver y editar información de todos los sitios web registrados en el gestor, y un usuario de tipo editor que estará ligado a un sitio web especifico y solo podrá editar el diseño y contenidos del mismo.
- Administración de fotos: Aquí el sistema permitirá a los usuarios subir fotografías al servidor web para posteriormente usarlas en las páginas creadas en el gestor de contenidos. Para optimizar el funcionamiento y facilitar las actualizaciones, el sistema almacenará las fotografías dentro de la base de datos y no como archivos físicos. De esta forma los directorios del sistema se mantendrán limpios y las fotografías no tendrán que ser respaldadas cuando sea necesario actualizar el servidor web o el sistema gestor de contenidos. Además, administrar las fotografías usando la base de datos crea una capa más de seguridad, puesto que el control de acceso a las imágenes será manejado por el sistema gestor de contenidos y no por el servidor web.
- Administración de videos: Para gestionar los videos el sistema usará los servicios gratuitos de YouTube. Cada sitio web que requiera incluir videos dentro de sus páginas, deberá crear una cuenta en YouTube y registrarla en el gestor de contenidos. Una vez que el sistema compruebe la comunicación con la cuenta registrada, permitirá a los usuarios subir, seleccionar y eliminar videos directamente desde la aplicación sin necesidad de abrir otras páginas web o usar otros utilitarios externos.

- Administración de galerías: En este módulo el usuario podrá crear, editar y eliminar galerías. El sistema soportara tres tipos de galerías: de fotos, de videos y de contenido. Una galería de contenido muestra pequeñas subpáginas, su funcionamiento es muy similar a una presentación de diapositivas. Muy útil para promocionar productos, servicios o cualquier tipo de contenido cuyo texto debe ser desplegado en varios idiomas por lo que una imagen no bastaría. La aplicación ofrece cuatro plantillas diferentes que el usuario puede escoger para crear una galería, con la posibilidad de aumentar el número de plantillas en futuras versiones.
- Administración de publicaciones: Dentro del sistema gestor de contenidos una publicación es cualquier tipo de contenido que se publica periódicamente, ya sean noticias, eventos, publicidades, etc. Una publicación se manejará como una página independiente, sobre la cual el usuario podrá hacer uso de todas las herramientas de edición que ofrece el sistema. En este módulo el usuario podrá definir qué tipo de publicaciones usará en su sitio y como el gestor deberá presentarlas.
- Administración de idioma: Uno de los primeros pasos al crear un sitio web, es registrar en el gestor en que idiomas serán presentados los contenidos. Por defecto el sistema solo tendrá registrado el idioma español.
- Administración de páginas web: Permite crear, editar y eliminar páginas web. Sin duda será el módulo más grande del gestor de contenidos, puesto que se pondrá especial énfasis en el interfaz de usuario. La clave para lograr reducir al máximo el nivel de conocimientos técnicos necesarios para usar el sistema, está en desarrollar una interfaz de usuario totalmente intuitiva y lo más gráfica posible. Para lograrlo la interfaz será desarrollada bajo el principio WYSIWYG (Acrónimo para what you see is what you get, lo que ves es lo que recibes en español), mediante el uso de librerías de JavaScript de última generación. El objetivo es que el usuario final pueda diseñar y construir páginas web sin necesidad de conocimientos específicos de HTML o CSS, mediante el uso de herramientas gráficas provistas por el sistema, cuyo funcionamiento sea intuitivo y principalmente que permitan visualizar directamente en el editor como lucirá la página web con cada cambio realizado.
- Editor de contenidos: El otro gran eje del sistema, aquí el usuario podrá editar el contenido que se visualiza en cada página registrada en gestor. Su funcionamiento garantizará total independencia del editor del diseño gráfico de las páginas, de tal

forma que el usuario pueda cambiar el diseño de las páginas del sitio sin que esto afecte de ninguna manera a la información que se presente en las mismas. Para brindar más versatilidad y opciones al usuario, el sistema incorporará un editor de texto con estilos que permitirá ingresar texto con formato al gestor de contenidos, es decir, se podrán utilizar opciones de formato como: negrilla, subrayado, alineación del texto, viñetas, entre otras.

- Generador de menú de navegación para el sitio web: En este módulo el sistema ofrece al usuario diferentes opciones para personalizar el menú de navegación de sitio. Por defecto el sistema soportará 4 estilos diferentes de menú, con la posibilidad de aumentarlos en futuras versiones.
- Módulo de seguridad: se encargará de proteger al área de administración de accesos no autorizados. También consiste en un log transaccional, donde se registran los cambios realizados en la capa de persistencia del sistema.

El sistema permitirá al usuario:

- Crear y diseñar páginas web que podrán ser utilizadas por acción social para implementarlos en los proyectos con los que trabaja
- Editar sus contenidos en varios idiomas
- Crear y editar artículos o noticias
- Crear y editar galerías de fotos y video
- Crear, editar y programar eventos
- Publicar el sitio web para el acceso del público en internet

Todo bajo un módulo de seguridad robusto para proteger los contenidos y la integridad de la página.

4. DESARROLLO DEL CMS

En este capítulo cubriremos los aspectos técnicos del desarrollo del gestor de contenidos. Empezamos defiendo los requerimientos funcionales y no funcionales para después describir la arquitectura y estructura del sistema.

4.1. Requerimientos funcionales

F1. Administrar Usuarios: permitirá crear, ver, editar y eliminar usuarios. Por defecto el sistema creará un usuario de tipo administrador la primera vez que se corra. Dicho usuario tendrá privilegios para crear sitios y crear otros usuarios en el sistema.

F2. Administrar Idiomas: permitirá administrar los idiomas en los cuales se presentarán las páginas web. Los idiomas serán propios de cada sitio registrado en el gestor y por defecto el sistema creará el idioma español la primera vez que se ejecute.

F3. Administrar Sitios: permitirá crear, ver y editar sitios web, el gestor de contenidos soportará el hospedaje de varios sitios web, cada sitio tendrá uno o más usuarios los cuales tendrán privilegios para editar el contenido y las páginas del mismo.

F4. Administrar Layouts: un layout en la aplicación es una especie de plantilla que da una noción o lineamentos de donde ubicar los elementos de una página web. La administración de layouts perime crear, editar y eliminar plantillas. Por defecto el sistema mostrará una plantilla básica con las secciones ubicadas en el centro de la pantalla, una debajo de otra.

F5. Administrar Páginas web: en esta administración el usuario podrá crear, ver, editar y eliminar páginas web de un sitio determinado. En cuanto a la interfaz gráfica, la pantalla de edición tendrá un alto contenido de programación en JavaScript, para lograr un interfaz de tipo WYSIWYG (what you see is what you get, en español: lo que ves es lo que obtienes).

F6. Administrar Secciones: permitirá crear, editar y eliminar secciones. Una sección es un contenedor que albergara el contenido de la página web. A cada sección se le podrán aplicar propiedades de estilo HTML para darle vistosidad.

F7. Administrar Contenido: el sistema soportará tres tipos de contenido: texto, fotos y videos. Con esta administración el usuario podrá crear, editar y eliminar los tres tipos de contenido antes mencionados.

F8. Administrar Galerías: permitirá crear, ver, editar y eliminar galerías. El sistema ofrecerá cuatro diseños de galerías predefinidos, cada uno de los cuales tendrá varias opciones de edición para que el usuario pueda personalizarlas a su gusto.

F9. Administrar Publicaciones: dentro del sistema entenderemos por publicación a cualquier tipo de contenido el cual es publicado periódicamente, ya sean noticias, eventos, promociones, etc. Mediante esta administración, el usuario podrá crear, editar y eliminar cualquier tipo de publicaciones.

F10. Administrar Guestbook: permitirá crear, editar y eliminar Guestbooks (libros de visitas). Un guestbook es un medio por el cual los visitantes de la página web, podrán dejar mensajes o comentarios.

F11. Administrar comentarios de guestbook: permitirá crear, eliminar y aprobar los mensajes dejados dentro de un guestbook. Los usuarios encargados del mantenimiento del sitio, podrán decidir cuales mensajes se publican en la página web y cuáles deben ser eliminados.

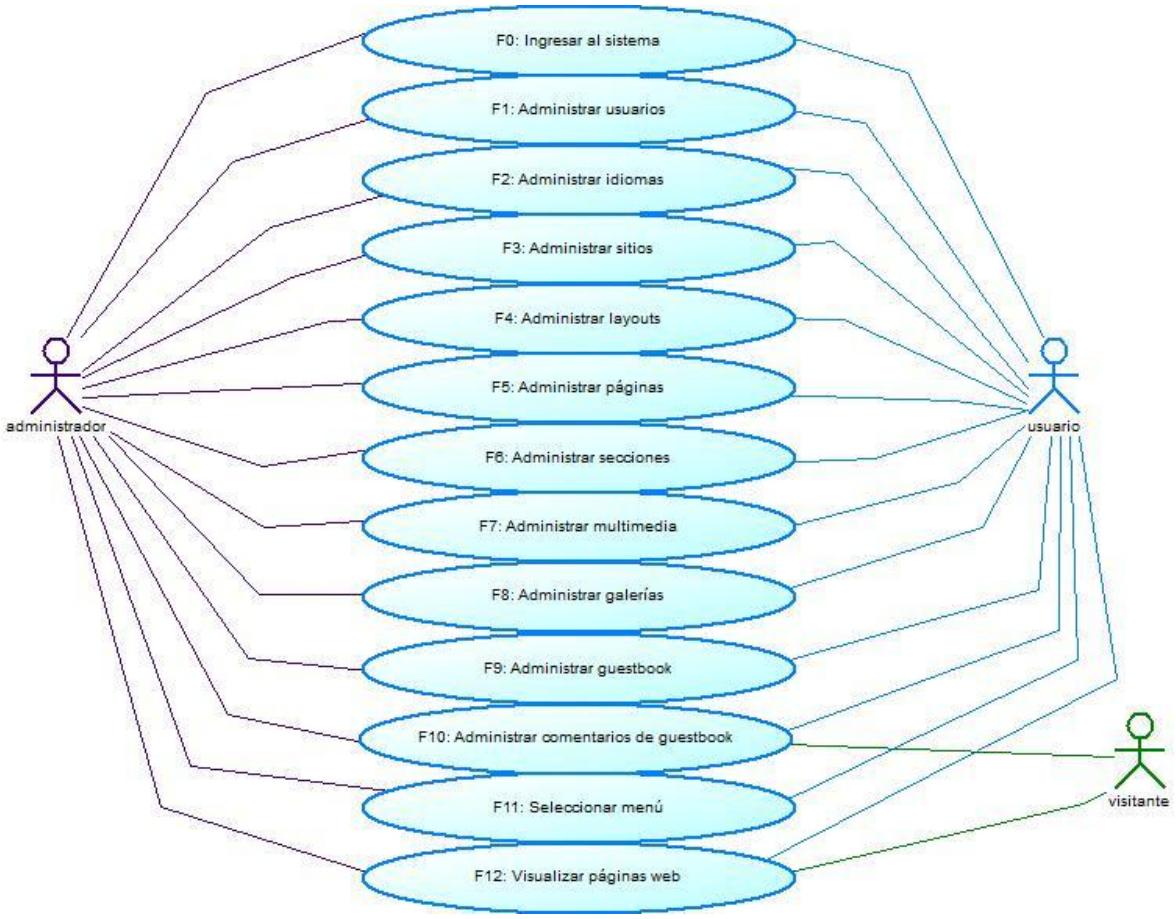
F12. Generación de Menús: aquí el usuario podrá elegir qué tipo de menú será utilizado en todo el sitio web. El sistema ofrecerá 4 tipos diferentes de menú, los cuales varían en color y diseño. Una vez escogido el diseño el sistema generara el menú como tal de forma dinámica con las páginas que hayan sigo registradas en el sistema.

F13. Visualización de las páginas web: esta administración se encargará de resolver las peticiones de los visitantes del sitio web, su labor incluye recibir los parámetros enviados en el URL y presentar la página solicitada.

4.2. Casos de uso

A continuación presentamos los casos de uso más relevantes para el desarrollo del gestor de contenido. La totalidad de casos de uso, junto con los gráficos en tamaño real, se encontrarán en el Anexo 1: Casos de uso.

4.2.1. Diagrama general de casos de uso



4.2.2. F0: Ingreso al sistema

Definición

Permite ingresar al sistema después de validar nombre de usuario y contraseña.

Actores

Administrador, Usuario

Gráfico



Flujo principal

Paso	Actor	Paso	Sistema	Excepción
1	Digita su nombre de usuario y contraseña	2	Comprueba el nombre de usuario	E1, E2
		3	Despliega la pantalla con la lista de sitios existentes y el formulario de creación de nuevo sitio	

Flujo alternativo

Paso	Acción
3	Se carga la pantalla de login con el mensaje de error correspondiente

Excepciones

Código	Descripción	Acción
E1	Problema de conexión con la base de datos	Notificar
E2	Usuario y/o contraseña incorrectos	Notificar

4.2.3. F1: Administrar usuarios

Definición

Permite ingresar usuarios nuevos, así como visualizar, editar y eliminar los usuarios existentes

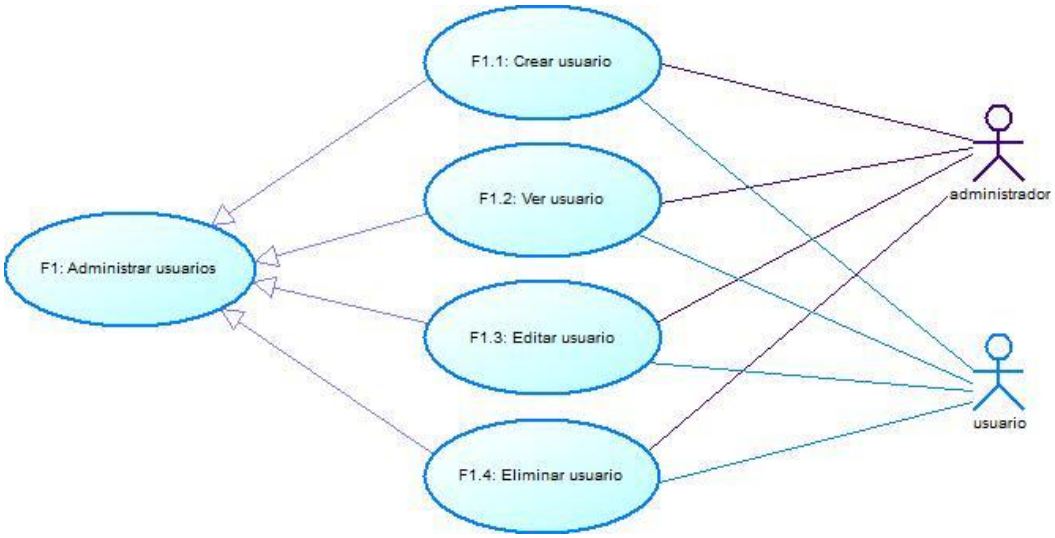
Actores

Administrador, Usuario

Precondiciones

Ninguna

Gráfico



Flujo principal

Escenario 1: crear usuario	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “Administración de usuarios” en el menú	2	Muestra la lista de usuarios existentes y el submenú de administración de usuarios	E1
	3	Selecciona “Crear usuario”	4	Muestra la pantalla con el formulario de creación de usuario	
	5	Completa la información necesaria			
	6	Clic en el botón guardar	7	Valida la información ingresada	E2
			8	Guarda el nuevo usuario	E1,E3
Escenario 2: ver usuario	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “usuarios” en el menú	2	Muestra la lista de usuarios existentes y el submenú de administración de usuarios	E1
	3	Clic en el id del usuario que desea visualizar	4	Muestra la pantalla de visualización de datos de usuario	E1
Escenario 3: editar usuario	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “usuarios” en el menú	2	Muestra la lista de usuarios existentes y el submenú de administración de usuarios	E1
	3	Clic en el id del usuario que desea editar	4	Muestra la pantalla de visualización de datos de usuario	E1
	5	Clic en el botón “Editar”	6	Muestra la pantalla con el formulario de edición de datos	
	7	Modifica los datos necesarios			E2
	8	Clic en el botón guardar	9	Valida la información ingresada	E1, E3
			10	Guarda el usuario modificado	
Escenario 4: eliminar usuario	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “usuarios” en el	2	Muestra la lista de usuarios existentes y el	E1

		menú		submenú de administración de usuarios	
	3	Clic en el id del usuario que desea eliminar	4	Muestra la pantalla de visualización de datos de usuario	E1
	5	Clic en el botón “Eliminar”	6	Muestra un mensaje de confirmación	
	7	Clic en el botón “Aceptar”	8	Modifica el estado del usuario a “inactivo”	E1, E2

Flujo alternativo

Escenario 1: crear usuario	**Paso**	**Acción**
	2	Muestra la tabla vacía con el mensaje “No existen usuarios en el sistema”
	7	Muestra la información de los errores encontrados al validar el formulario
Escenario 2: ver usuario	**Paso**	**Acción**
	2	Muestra la tabla vacía con el mensaje “No existen usuarios en el sistema”
Escenario 3: editar usuario	**Paso**	**Acción**
	2	Muestra la tabla vacía con un mensaje de “No existen usuarios en el sistema”
	10	Muestra la información de los errores encontrados al validar el formulario
Escenario 4: eliminar usuario	**Paso**	**Acción**
	2	Muestra la tabla vacía con un mensaje de “No existen usuarios en el sistema”

Excepciones

Código	**Descripción**	**Acción**
E1	Error de conexión con la base de datos	Notificar
E2	Error de validación, como caracteres especiales en el nombre de usuario, caracteres mínimos no cumplidos, etc.	Notificar
E3	Error al guardar los datos, por ejemplo, nombre de usuario repetido	Notificar

4.2.4. F2: Administrar idiomas

Definición

Permite ingresar idiomas nuevos, así como visualizar, editar y eliminar los existentes

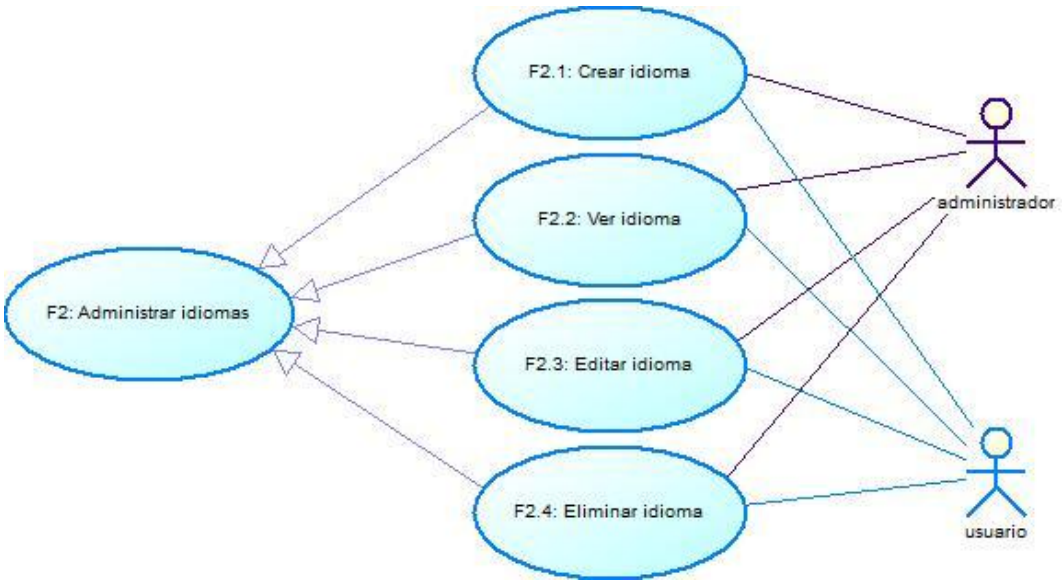
Actores

Administrador, Usuario

Precondiciones

Un usuario registrado en el sistema

Gráfico



Flujo principal

Escenario 1: crear idioma	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón “Idiomas” en el menú	2	Muestra la lista de idiomas existentes y el submenú de administración de idiomas	E1
	3	Selecciona “Crear idioma”	4	Muestra la pantalla con el formulario de creación de idioma	
	5	Completa la información necesaria			
	6	Clic en el botón guardar	7	Valida la información ingresada	E2
			8	Guarda el nuevo idioma	E1,E3
Escenario 2: ver idioma	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón “Idiomas” en el menú	2	Muestra la lista de idiomas existentes y el submenú de administración de idiomas	E1
	3	Clic en el id del idioma que desea visualizar	4	Muestra la pantalla de visualización de datos de idioma	E1
Escenario 3: editar idioma	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón “Idiomas” en el menú	2	Muestra la lista de idiomas existentes y el submenú de administración de idiomas	E1

	3	Clic en el id del idioma que desea editar	4	Muestra la pantalla de visualización de datos de idioma	E1
	5	Clic en el botón “Editar”	6	Muestra la pantalla con el formulario de edición de datos	
	7	Modifica los datos necesarios			E2
	8	Presiona el botón guardar	9	Valida la información ingresada	E1, E3
			10	Guarda el idioma modificado	
Escenario 4: eliminar idioma	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón “Idiomas” en el menú	2	Muestra la lista de idiomas existentes y el submenú de administración de idiomas	E1
	3	Clic en el id del idioma que desea eliminar	4	Muestra la pantalla de visualización de datos de idioma	E1
	5	Clic en el botón “Eliminar”	6	Muestra un mensaje de confirmación	
	7	Clic en el botón “Aceptar”	8	Elimina el idioma	E1, E2

Flujo alternativo

Escenario 1: crear idioma	**Paso**	**Acción**
	2	Muestra la tabla vacía con el mensaje “No existen idiomas en el sistema”
	7	Muestra la información de los errores encontrados al validar el formulario
Escenario 2: ver idioma	**Paso**	**Acción**
	2	Muestra la tabla vacía con el mensaje “No existen idiomas en el sistema”
Escenario 3: editar idioma	**Paso**	**Acción**
	2	Muestra la tabla vacía con un mensaje de “No existen idiomas en el sistema”
	10	Muestra la información de los errores encontrados al validar el formulario
Escenario 4: eliminar idioma	**Paso**	**Acción**
	2	Muestra la tabla vacía con un mensaje de “No existen idiomas en el sistema”

Excepciones

Código	**Descripción**	**Acción**
E1	Error de conexión con la base de datos	Notificar

E2	Error de validación, como caracteres especiales en el nombre de idioma, caracteres mínimos no cumplidos, etc.	Notificar
E3	Error al guardar los datos	Notificar

4.2.5. F3: Administrar sitios

Definición

Permite ingresar, visualizar, editar y eliminar sitios

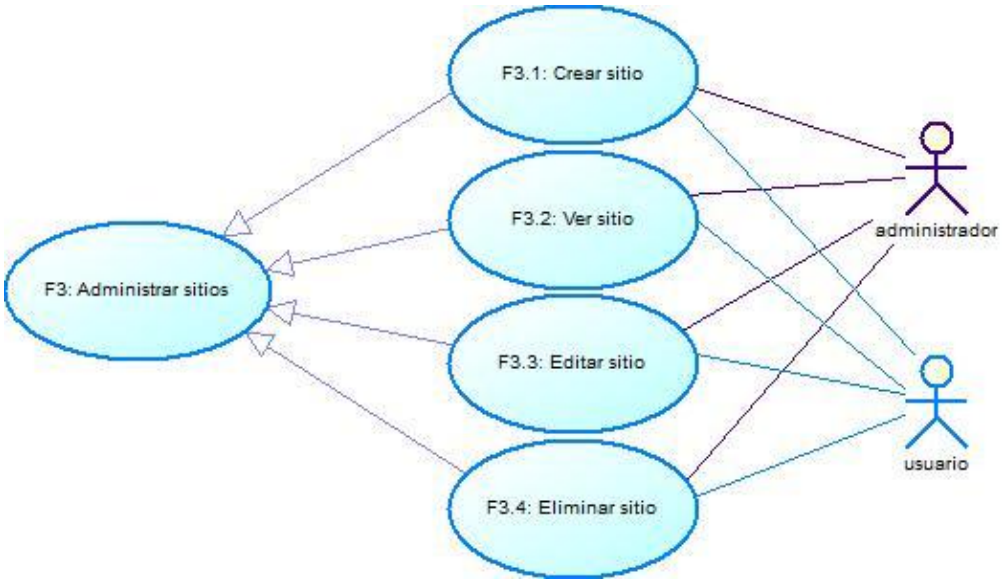
Actores

Administrador, Usuario

Precondiciones

Un usuario registrado en el sistema

Gráfico



Flujo principal

Escenario 1: crear sitio	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
			1	Muestra la lista de sitios disponibles y el formulario de creación de sitio	E1
	2	Completa el formulario con los datos necesarios para la creación del sitio			
	3	Clic en el botón guardar	4	Valida los datos ingresados	E2
			5	Guarda el nuevo sitio	E1, E3

Escenario 2: ver sitio	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “sitios” en el menú	2	Muestra la lista de sitios existentes y el submenú de administración de sitios	E1
	3	Clic en el id del sitio que desea visualizar	3	Muestra la pantalla de visualización de datos de sitio	E1
Escenario 3: editar sitio	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “sitios” en el menú	2	Muestra la lista de sitios existentes y el submenú de administración de sitio	E1
	3	Clic en el id del sitio que desea editar	4	Muestra la pantalla de visualización de datos de sitio	E1
	5	Clic en el botón “Editar”	6	Muestra la pantalla con el formulario de edición de datos	
	7	Modifica los datos necesarios			E2
	8	Presiona el botón guardar	9	Valida la información ingresada	E1, E3
			10	Guarda el sitio modificado	
Escenario 4: eliminar sitio	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “sitios” en el menú	2	Muestra la lista de sitios existentes y el submenú de administración de sitios	E1
	3	Clic en el id del sitios que desea eliminar	4	Muestra la pantalla de visualización de datos de sitio	E1
	5	Clic en el botón “Eliminar”	6	Muestra un mensaje de confirmación	
	7	Clic en el botón “Aceptar”	8	Elimina el sitio	E1, E2

Flujo alternativo

Escenario 1: crear sitio	**Paso**	**Acción**
	2	Muestra la lista vacía ya que no existen sitios aún
	4	Muestra la información de los errores encontrados al validar el formulario
Escenario 2: ver sitio	**Paso**	**Acción**
	2	Muestra la tabla vacía con el mensaje “No existen sitios en el sistema”
Escenario 3: editar sitio	**Paso**	**Acción**
	2	Muestra la tabla vacía con un mensaje de “No existen sitios en el sistema”

	10	Muestra la información de los errores encontrados al validar el formulario
Escenario 4: eliminar sitio	**Paso**	**Acción**
	2	Muestra la tabla vacía con un mensaje de “No existen sitios en el sistema”

Excepciones

Código	Descripción	Acción
E1	Error de conexión con la base de datos	Notificar
E2	Error de validación, como caracteres especiales en el nombre de idioma, caracteres mínimos no cumplidos, etc.	Notificar
E3	Error al guardar los datos	Notificar

4.2.6. F4: Administrar layouts (plantillas)

Definición

Permite ingresar, visualizar, editar y eliminar layouts.

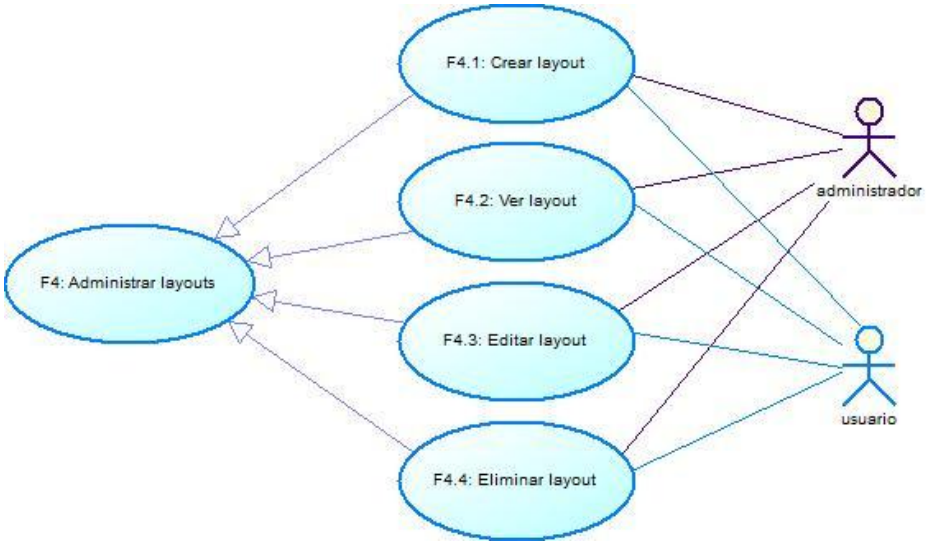
Actores

Administrador, usuario

Precondiciones

Un usuario registrado en el sistema

Gráfico



Flujo principal

Escenario 1: crear layout	Paso	Actor	Paso	Sistema	Excepción
	1	Clic en el botón “layouts” en el menú	2	Muestra el submenú de	

				layouts	
	3	Clic en el botón “nuevo”	4	Muestra la pantalla de edición de layouts	
	4	Arrastra con el mouse los recuadros de las secciones de menú, banner y contenido, hasta ubicarlos en el lugar deseado			
	6	Clic en el botón “guardar”	7	Muestra un diálogo con el formulario de layout	
	8	Ingresa el nombre y la descripción del layout			
	9	Clic en el botón guardar	10	Valida los datos ingresados	
			11	Guarda los datos en la base de datos	E1, E2
Escenario 2: ver layout	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “layouts” en el menú.	2	Muestra un submenú con la lista de los layouts existentes	E1
	3	Clic en el layout que desea visualizar	4	Carga los datos del layout y muestra la pantalla de edición de layouts	
Escenario 3: editar layout	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “layouts” en el menú.	2	Muestra un submenú con la lista de los layouts existentes	E1
	3	Clic en el layout que desea editar	4	Carga los datos del layout y muestra la pantalla de edición de layouts	
	5	Arrastra con el mouse los recuadros de las secciones de menú, banner y contenido, hasta ubicarlos en el lugar deseado			
	6	Clic en el botón	7	Muestra un	

		“guardar”		diálogo con el formulario de layout	
	8	edita el nombre y la descripción del layout			
	9	Clic en el botón guardar	10	Valida los datos ingresados	
			11	Guarda los datos en la base de datos	E1, E2

Escenario 4: eliminar layout	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “layouts” en el menú.	2	Muestra un submenú con la lista de los layouts existentes	E1
	3	Clic en el layout que desea eliminar	4	Carga los datos del layout y muestra la pantalla de edición de layouts	
	5	Clic en el botón eliminar	6	Muestra un mensaje de confirmación	
	7	Clic en aceptar	8	Elimina el layout	E3

Flujo alternativo

Escenario 1: crear layout	**Paso**	**Acción**
	11	Muestra los errores en los datos ingresados por el usuario
Escenario 2: ver layout	**Paso**	**Acción**
	2	Se muestra el submenú vacío debido a que no hay layouts registrados en el sistema
Escenario 3: editar layout	**Paso**	**Acción**
	11	Muestra los errores de los datos ingresados por el usuario

Excepciones

Código	**Descripción**	**Acción**
E1	Error de conexión con la base de datos	Notificar
E2	Error de actualización	Notificar
E3	Error de elimination	Notificar

4.2.7. F5: Administrar páginas

Definición

Permite crear, visualizar, editar y eliminar páginas

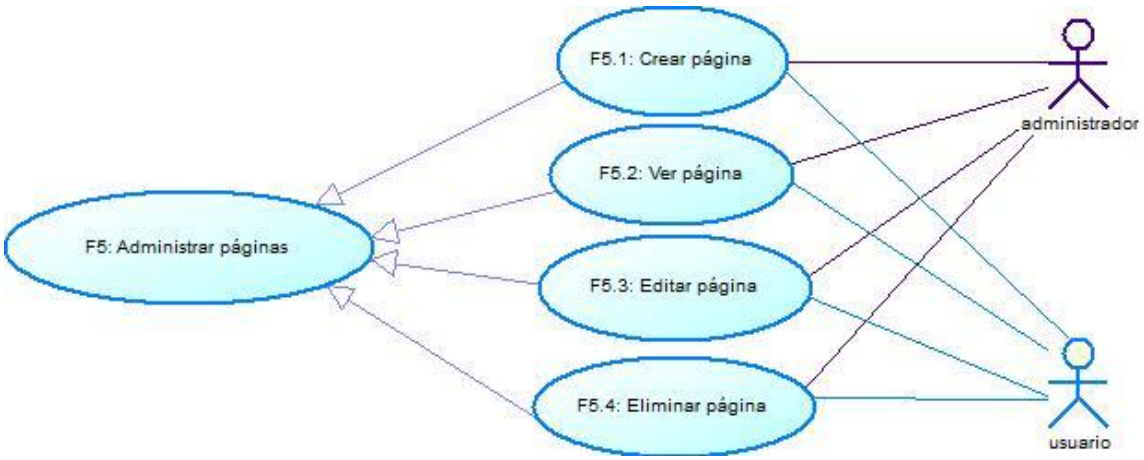
Actores

Administrador, Usuario

Precondiciones

Un usuario registrado en el sistema, un layout registrado en el sistema

Gráfico



Flujo principal

Escenario 1: crear página	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “Páginas” en el menú	2	Muestra el submenú de páginas	E1
	3	Selecciona “Nueva página”	4	Muestra la pantalla de creación de páginas	
	5	Escoge una plantilla			
	6	Llena los datos del formulario			
	7	Clic en guardar	8	Guarda los datos en la base de datos	E1, E2
			9	Presenta la pantalla de edición de contenidos	
Escenario 2: ver página	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “páginas” en el menú	2	Muestra el submenú de páginas que contiene las páginas creadas en el sitio	E1
	3	Clic en el botón con el nombre de la página que desee editar	3	Despliega la pantalla de edición de páginas con los datos de la página seleccionada	E1
Escenario 3: editar página	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón “páginas” en el menú del sistema.	2	Muestra el submenú de páginas que contiene las páginas creadas en el sitio	E1
	3	Clic en el botón con	4	Despliega la pantalla	E1

		el nombre de la página que se desee editar		de edición de páginas	
	5	Clic en el botón "editar"	6	Muestra la pantalla con el formulario de edición de datos	
	7	El actor modifica los datos necesarios			E2
	8	El usuario presiona el botón guardar	9	Guarda los datos en la base de datos	E1, E3
Escenario 4: eliminar página	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón "páginas" en el menú del sistema.	2	Muestra el submenú de páginas que contiene las páginas creadas en el sitio	E1
	3	Clic en el botón con el nombre de la página que se desee eliminar	4	Despliega la pantalla de edición de páginas	E1
	5	Clic en el botón "Eliminar"	6	Muestra un mensaje de confirmación	
	7	Clic en el botón "Aceptar"	8	Elimina el registro en la base de datos	E1, E2

Flujo alternativo

Escenario 1: crear página	**Paso**	**Acción**
	2	Muestra el submenú vacío pues no hay páginas creadas
	9	Muestra un mensaje con los errores encontrados en los datos ingresados
Escenario 2: ver página	**Paso**	**Acción**
	2	Muestra el submenú vacío pues no hay páginas creadas
Escenario 3: editar página	**Paso**	**Acción**
	2	Muestra el submenú vacío pues no hay páginas
	10	Muestra la información de los errores encontrados
Escenario 4: eliminar página	**Paso**	**Acción**
	2	Muestra el submenú vacío pues no existen páginas en el sistema

Excepciones

Código	Descripción	Acción
E1	Error de conexión con la base de datos	Notificar
E2	Error de validación, como caracteres especiales, caracteres mínimos no cumplidos, etc.	Notificar
E3	Error al guardar los datos	Notificar

4.2.8. F6: Administrar secciones

Definición

Permite ingresar, visualizar, editar y eliminar secciones.

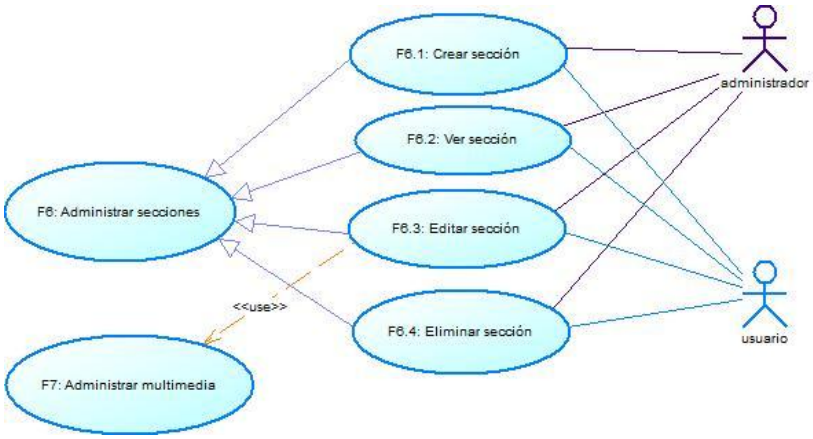
Actores

Administrador, Usuario

Precondiciones

Un usuario, layout y página registrados en el sistema

Gráfico



Flujo principal

Escenario 1: crear sección	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “Páginas” en el menú	2	Muestra un submenú con la lista de las páginas existentes	E1
	3	Clic en el botón con el nombre de la página que va a contener la nueva sección	4	Muestra la pantalla de edición de páginas web	
	5	Clic en el botón insertar sección	6	Despliega una lista con los tipos de secciones disponibles para insertar	
	6	Clic en el tipo de sección que se desea insertar	8	Guarda la nueva sección en la base de datos	E1
			9	Crea un recuadro que representa a la nueva sección	
Escenario 2: ver sección	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “Páginas” en el menú	2	Muestra un submenú con la	E1

				lista de las páginas existentes	
	3	Clic en el botón con el nombre de la página que contiene la sección que se desea eliminar	4	Muestra la pantalla de edición de páginas web	E1
			5	Carga las secciones de la página	

Escenario 3: editar sección	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “Páginas” en el menú	2	Muestra un submenú con la lista de las páginas existentes	E1
	3	Clic en el botón con el nombre de la página que contiene la sección que se desea editar	4	Muestra la pantalla de edición de páginas web	E1
	5	Selecciona el recuadro que representa la sección	6	Muestra una barra de herramientas con opciones de edición	
	7	Clic en el botón propiedades	8	Muestra una ventana con las propiedades de la sección	
	9	Edita las propiedades			
	10	Clic en el botón aplicar	11	Aplica las propiedades al recuadro que representa a la sección	
	12	Clic en el botón “editar”	13	Abre una nueva ventana en el navegador con la pantalla de edición de contenido multimedia	
			14	Ver caso de uso F7 administrar multimedia	
	15	Clic en el botón guardar	16	Valida los datos ingresados	
			17	Guarda los datos en la base de datos	E1, E2
			18	Cierra la ventana de edición	

Escenario 4: eliminar sección	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “Páginas”	2	Muestra un	E1

		en el menú		submenú con la lista de las páginas existentes	
	3	Clic en el botón con el nombre de la página que contiene la sección que se desea eliminar	4	Muestra la pantalla de edición de páginas web	E1
	5	Selecciona el recuadro que representa la sección	6	Muestra una barra de herramientas con opciones de edición	
	7	Clic en el botón eliminar	8	Muestra un mensaje de confirmación	
	8	Clic en aceptar	9	Elimina la sección	E1,E2
			10	Elimina el recuadro	

Flujo alternativo

Escenario 1: crear sección	**Paso**	**Acción**
	2	Se muestra el submenú de páginas vacío debido a que no hay páginas registradas
Escenario 2: ver sección	**Paso**	**Acción**
	2	Se muestra el submenú de páginas vacío debido a que no hay páginas registradas
Escenario 3: editar sección	**Paso**	**Acción**
	2	Se muestra el submenú de páginas vacío debido a que no hay páginas registradas
Escenario 4: eliminar sección	**Paso**	**Acción**
	2	Se muestra el submenú de páginas vacío debido a que no hay páginas registradas

Excepciones

Código	Descripción	Acción
E1	Error de conexión con la base de datos	Notificar
E2	Error de inserción	Notificar

4.2.9. F7: Administrar multimedia

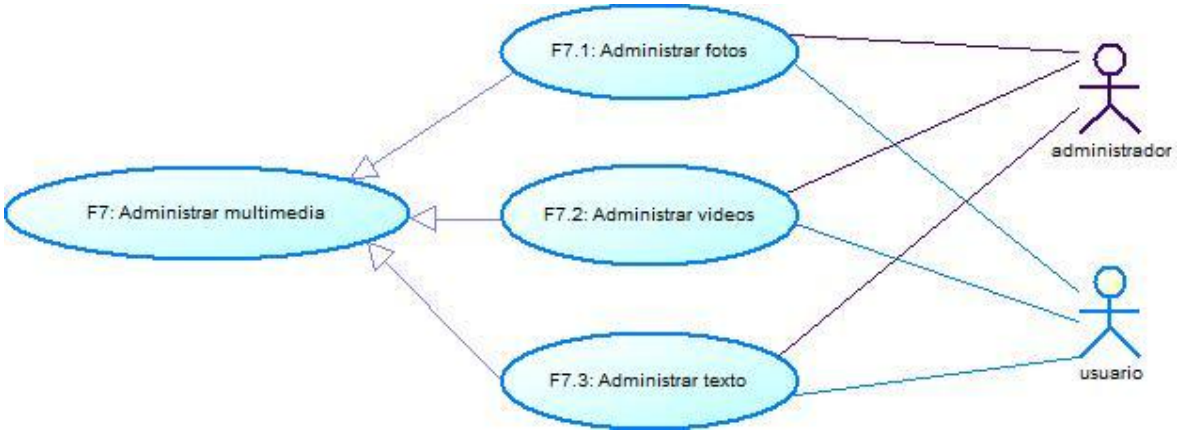
Definición

Permite ingresar, visualizar, editar y eliminar fotos, videos y textos. Se analizará cada uno de estos casos de forma individual.

Actores

Administrador, usuario

Gráfico



4.2.9.1. F7.1: Administrar fotos

Definición

Permite ingresar, visualizar, editar y eliminar fotos.

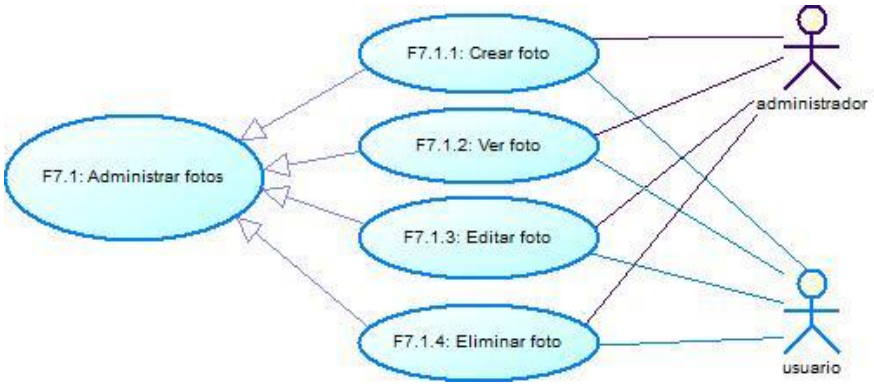
Actores

Administrador, usuario

Precondiciones

Un usuario, layout, página y sección registrados en el sistema

Gráfico



Flujo principal

Escenario 1: crear foto	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón editar sección de fotos	2	Abre una ventana en el navegador con el formulario de fotos	
	3	Llena los campos del formulario			
	4	Clic en el botón “seleccionar	5	Abre una ventana para escoger la imagen	

		archivo”			
	6	Doble clic en la foto que desee subir	7	Cierra la ventana para escoger imágenes	
	8	Clic en guardar	9	Valida los datos ingresados	
			10	Guarda los datos ingresados	E1,E2
			11	Cierra la ventana de edición	
Escenario 2: ver foto	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona páginas en el menú.	2	Muestra un submenú con la lista de páginas existentes	E1
	3	Clic en la página que contiene la foto	4	Muestra la pantalla de edición de páginas y carga las secciones contenidas en la misma	E1
			5	Muestra las fotos contenidas en las secciones de la página	
Escenario 3: editar foto	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón editar sección de fotos	2	Abre una ventana en el navegador con el formulario de fotos con los datos de la foto	
	3	Edita los campos del formulario			
	4	Clic en guardar	5	Valida los datos ingresados	
			6	Guarda los datos ingresados	E1,E2
			7	Cierra la ventana de edición	
Escenario 4: eliminar foto	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el bóton eliminar en la barra de herramientas de la sección	2	Muestra un mensaje de confirmación	
	3	Clic en aceptar	4	Elimina la foto	E1,E2
			5	Elimina el recuadro en la pantalla de edición de páginas	

Flujo alternativo

Escenario 1: crear foto	**Paso**	**Acción**
	10	Presenta un mensaje con los errores de los datos ingresados

Escenario 2: ver foto	Paso	Acción
	2	Se muestra el submenú vacío debido a que no hay páginas registradas en el sistema
Escenario 3: editar foto	**Paso**	**Acción**
	6	Muestra un mensaje con los errores encontrados

Excepciones

Código	Descripción	Acción
E1	Error de conexión con la base de datos	Notificar
E2	Error de inserción	Notificar

4.2.9.2. F7.2: Administrar videos

Definición

Permite ingresar, visualizar, editar y eliminar videos.

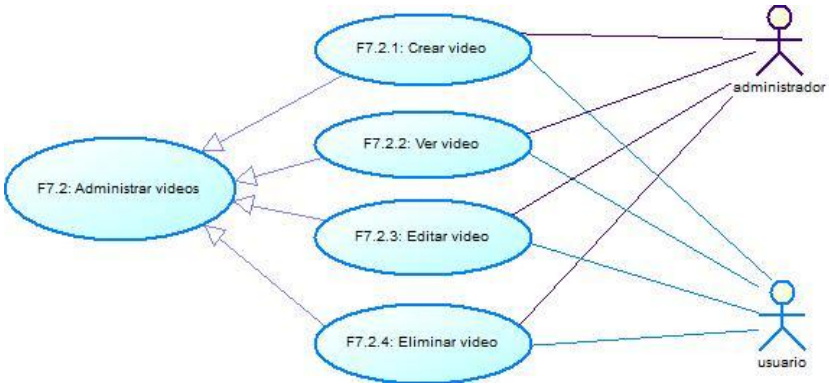
Actores

Administrador, usuario

Precondiciones

Un usuario, layout, página y sección registrados en el sistema

Gráfico



Flujo principal

Escenario 1: crear video	Paso	Actor	Paso	Sistema	Excepción
	1	Clic en el botón editar sección de videos	2	Abre una ventana en el navegador con la notificación de uso de YouTube	
	3	Presiona “continuar”	4	Muestra la ventana de login de YouTube	
	4	Ingresa usuario y contraseña			

	5	Clic en el botón “login”	7	Muestra la ventana de autorización de contenidos de YouTube	
	8	Clic en “otorgar acceso”	9	Muestra la ventana de selección de acción	
	10	Clic en subir video	11	Muestra el formulario de datos del video	
	12	Completa el formulario			
	13	Clic en “continuar”	14	Valida la información ingresada	
			15	Muestra la ventana de selección de archivo de video	
	16	Clic en el botón “seleccionar archivo”	17	Abre una ventana para escoger el video	
	18	Doble clic en el video que desee subir	19	Cierra la ventana para escoger videos	
	20	Clic en “guardar”	21	Guarda los datos y carga el video a YouTube	E1, E2, E3
			22	Muestra la ventana de pre visualización del video	
	23	Clic en “terminar”	24	Cierra la ventana de edición	
Escenario 2: ver video	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona páginas en el menú.	2	Muestra un submenú con la lista de páginas existentes	E1
	3	Clic en la página que contiene el video	4	Muestra la pantalla de edición de páginas y carga las secciones contenidas en la misma	E1
			5	Muestra os videos contenidos en las secciones de la página	
Escenario 3: editar video	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón editar sección de videos	2	Abre una ventana en el navegador con la notificación de uso de YouTube	
	3	Presiona “continuar”	4	Muestra la ventana de YouTube de login	
	4	Ingresa usuario y contraseña			

	5	Clic en el botón “login”	6	Muestra la ventana de autorización de contenidos de YouTube	
	6	Clic en “otorgar acceso”	8	Muestra la ventana de selección de acción	
	9	Clic en “ver lista”	10	Muestra los videos cargados en la cuenta de YouTube	E3
	11	Selecciona el video deseado			
	12	Clic en “continuar”	13	Guarda la selección	E1,E2
			13	Muestra la ventana de pre visualización del video	
	14	Clic en “terminar”	15	Cierra la ventana de edición	

Escenario 4: eliminar video	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón eliminar en la barra de herramientas de la sección	2	Muestra un mensaje de confirmación	
	3	Clic en aceptar	4	Elimina el video de la página (no de YouTube)	E1
			5	Elimina el recuadro en la pantalla de edición de páginas	

Flujo alternativo

Escenario 1: crear video	**Paso**	**Acción**
	2	Ya tiene autorización de YouTube, pasa al paso 9
	12	Presenta un mensaje con los errores de los datos ingresados
Escenario 2: ver video	**Paso**	**Acción**
	2	Se muestra el submenú vacío debido a que no hay páginas registradas en el sistema
Escenario 3: editar video	**Paso**	**Acción**
	2	Ya tiene autorización de YouTube, pasa al paso 7

Excepciones

Código	**Descripción**	**Acción**
E1	Error de conexión con la base de datos	Notificar
E2	Error de inserción	Notificar
E3	Error de conexión con YouTube	Notificar

4.2.9.3. F7.3: Administrar texto

Definición

Permite ingresar, visualizar, editar y eliminar textos.

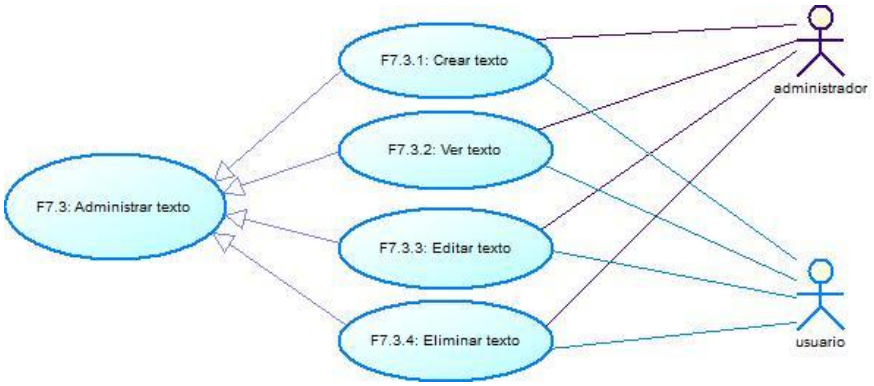
Actores

Administrador, usuario

Precondiciones

Un usuario, layout, página y sección registrados en el sistema

Gráfico



Flujo principal

Escenario 1: crear texto	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón editar sección de texto	2	Abre una ventana en el navegador con un editor de texto con formato	
	3	Escribe el texto deseado, incluyendo el formato (color, tamaño, etc)			
	4	Clic en guardar	5	Guarda los datos ingresados	E1,E2
			6	Cierra la ventana de edición	

Escenario 2: ver texto	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona páginas en el menú.	2	Muestra un submenú con la lista de páginas existentes	E1
	3	Clic en la página que contiene el texto	4	Muestra la pantalla de edición de páginas y carga las secciones contenidas en la misma	E1
			5	Muestra los textos contenidos en las secciones de la página	

Escenario 3: editar texto	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón editar sección de texto	2	Abre una ventana en el navegador con el editor de texto con formato	
	3	Edita el texto y/o el formato del mismo			
	4	Clic en guardar	5	Guarda los datos ingresados	E1,E2
			6	Cierra la ventana de edición	
Escenario 4: eliminar texto	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el bóton eliminar en la barra de herramientas de la sección	2	Muestra un mensaje de confirmación	
	3	Clic en aceptar	4	Elimina el texto	E1
			5	Elimina el recuadro en la pantalla de edición de páginas	

Flujo alternativo

Escenario 2: ver texto	**Paso**	**Acción**
	2	Se muestra el submenú vacío debido a que no hay páginas registradas en el sistema

Excepciones

Código	**Descripción**	**Acción**
E1	Error de conexión con la base de datos	Notificar
E2	Error de inserción	Notificar

4.2.10. F8: Administrar Galerías

Definición

Permite ingresar, visualizar, editar y eliminar galerías.

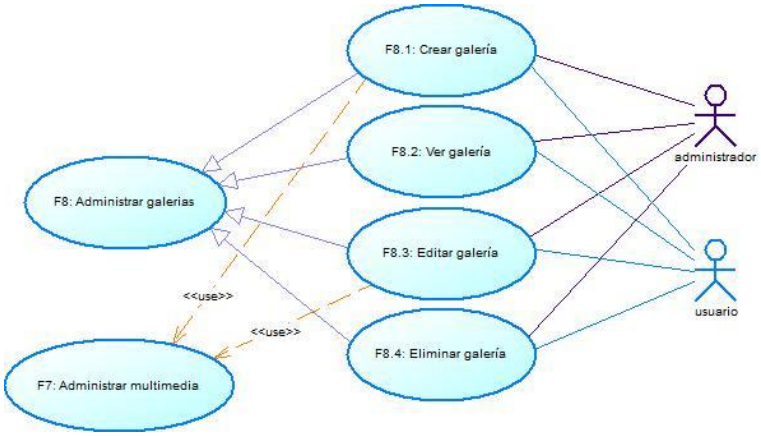
Actores

Administrador, Usuario

Precondiciones

Un usuario, sitio registrados en el sistema

Gráfico



Flujo principal

Escenario 1: crear galería					
	Paso	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón “Galerías” en el menú	2	Muestra un submenú con las diferentes opciones: Texto y Fotos	
	3	Clic en el botón con el tipo de galería a crear	4	Muestra un submenú con las galerías de ese tipo existentes	E1
	5	Clic en el botón “Crear nueva”	6	Muestra una pantalla con diferentes modelos de galerías posibles	
	7	Selecciona el modelo de galería	8	Muestra una vista previa del modelo de galería con fotos de demostración	E2
	9	Completa el formulario de creación de galería			
	10	Clic en el botón “Guardar”	11	Crea una nueva galería vacía	E1
			12	Muestra la pantalla de edición de multimedia	
			13	Ver caso de uso F7 administrar multimedia	
	14	Clic en botón “Guardar”	15	Guarda los datos en la base de datos	E1
Escenario 2: ver galería	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón “Páginas” en el menú	2	Muestra un submenú con la lista de las páginas existentes	E1
	3	Clic en el botón con el nombre de la página que contiene la galería	4	Muestra la pantalla de edición de páginas web	E1

	que se desea ver				
			5	Carga las secciones de la página, incluyendo las galerías	
Escenario 3: editar galería	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón “Galerías” en el menú	2	Muestra un submenú con las diferentes opciones: Texto y Fotos	
	3	Clic en el botón con el tipo de galería a editar	4	Muestra un submenú con las galerías de ese tipo existentes	E1
	5	Clic en el botón con el nombre de la galería que se desea editar	6	Muestra la pantalla de edición de multimedia	E1
			7	Ver caso de uso F7 administrar multimedia	
	8	Clic en el botón guardar	9	Guarda los datos en la base de datos	E1, E2
Escenario 4: eliminar galería	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Clic en el botón “Galerías” en el menú	2	Muestra un submenú con las diferentes opciones: Texto y Fotos	
	3	Clic en el botón con el tipo de galería a eliminar	4	Muestra un submenú con las galerías de ese tipo existentes	E1
	5	Clic en el botón con el nombre de la galería que se desea eliminar	6	Muestra la pantalla de edición de multimedia	E1
	7	Clic en el botón “Eliminar”	8	Muestra un mensaje de confirmación	
	9	Clic en el botón “Aceptar”	10	Elimina las fotos de la galería	E1, E2
			11	Elimina la galería	E1, E2
			12	Elimina el botón con el nombre de la galería eliminada	

Flujo alternativo

Escenario 1: crear galería	**Paso**	**Acción**
	4	Se muestra el submenú de galerías vacío debido a que no hay galerías registradas
Escenario 2:	**Paso**	**Acción**

ver galería	**2**	Se muestra el submenú de páginas vacío debido a que no hay páginas registradas
Escenario 3: editar galería	**Paso**	**Acción**
	4	Se muestra el submenú de galerías vacío debido a que no hay páginas registradas
Escenario 4: eliminar galería	**Paso**	**Acción**
	2	Se muestra el submenú de galerías vacío debido a que no hay galerías registradas

Excepciones

Código	Descripción	Acción
E1	Error de conexión con la base de datos	Notificar
E2	Error de acceso a los archivos de demostración	Notificar

4.2.11. F9: Administrar guestbook

Definición

Permite crear, visualizar, editar y eliminar guestbooks.

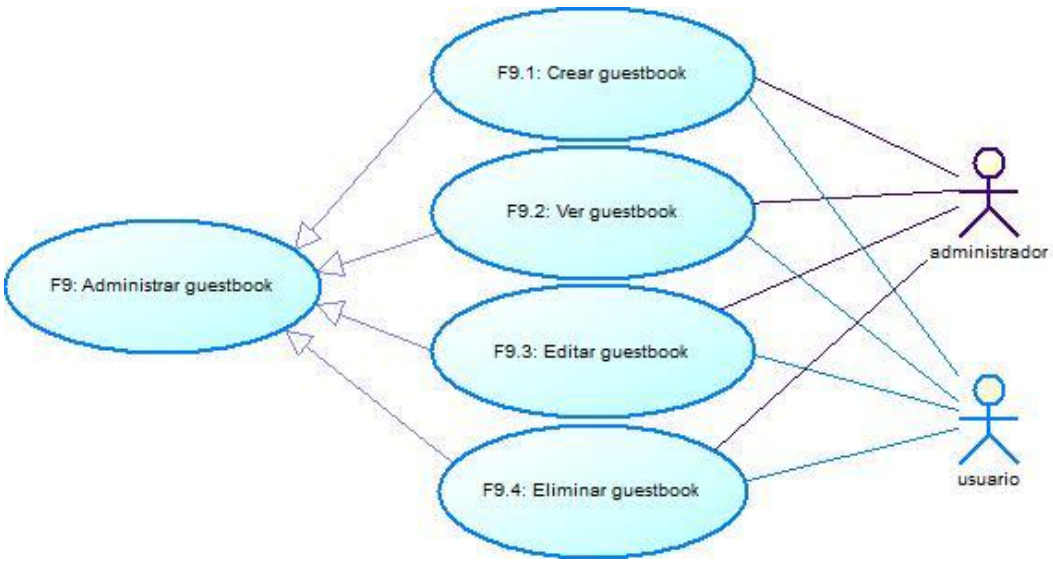
Actores

Administrador, Usuario

Precondiciones

Un usuario, sitio, layout, y página registrados en el sistema

Gráfico



Flujo principal

Escenario 1:	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**

crear guestbook	**1**	Selecciona “Páginas” en el menú	2	Muestra un submenú con la lista de las páginas existentes	E1
	3	Clic en el botón con el nombre de la página que va a contener el guestbook	4	Muestra la pantalla de edición de páginas web	
	5	Clic en el botón insertar sección	6	Despliega una lista con los tipos de secciones disponibles para insertar	
	7	Clic en “Guestbook”	8	Guarda la nueva sección en la base de datos	E1, E2
			9	Crea un recuadro que representa a la nueva sección	

Escenario 2: ver guestbook	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Escribe el url de la página web que contiene el guestbook	2	Recibe la petición	
			3	Busca en la base de datos la página solicitada	E1, E2
			4	Carga las secciones, fotos, videos y otros elementos de la página solicitada, incluyendo el guestbook	
			5	Muestra la página web	

Escenario 3: editar guestbook	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona “Páginas” en el menú	2	Muestra un submenú con la lista de las páginas existentes	E1
	3	Clic en el botón con el nombre de la página que contiene el guestbook que se desea editar	4	Muestra la pantalla de edición de páginas web	E1
	5	Selecciona el recuadro que representa la sección	6	Muestra una barra de herramientas con opciones de edición	
	7	Clic en el botón propiedades	8	Muestra una ventana con las propiedades de la sección	
	9	Edita las propiedades			

	10	Clic en el botón aplicar	11	Aplica las propiedades al recuadro que representa a la sección	
	12	Clic en el botón "Editar"	13	Abre una nueva ventana en el navegador con la pantalla de edición de guestbook	
	14	Modifica las etiquetas y/o el orden de los campos			
	15	Clic en el botón guardar	16	Valida los datos ingresados	
			17	Guarda los datos en la base de datos	E1, E2
			18	Cierra la ventana de edición	
Escenario 4: eliminar guestbook	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona "Páginas" en el menú	2	Muestra un submenú con la lista de las páginas existentes	E1
	3	Clic en el botón con el nombre de la página que contiene el guestbook que se desea eliminar	4	Muestra la pantalla de edición de páginas web	E1
	5	Selecciona el recuadro que representa la sección	6	Muestra una barra de herramientas con opciones de edición	
	7	Clic en el botón eliminar	8	Muestra un mensaje de confirmación	
	8	Clic en aceptar	9	Elimina el guestbook	E1,E2
			10	Elimina el recuadro	

Flujo alternativo

Escenario 1: crear guestbook	**Paso**	**Acción**
	2	Muestra el submenú vacío pues no hay páginas en el sistema
Escenario 2: ver guestbook	**Paso**	**Acción**
	4	No encuentra la página solicitada, muestra la página de error 404
Escenario 3: editar guestbook	**Paso**	**Acción**
	2	Muestra el submenú vacío pues no existen páginas en el sistema
	10	Muestra la información de los errores encontrados al validar los

	datos
Escenario 4: eliminar guestbook	**Paso** **Acción** **2** Muestra el submenú vacío pues no existen páginas en el sistema

Excepciones

Código	Descripción	Acción
E1	Error de conexión con la base de datos	Notificar
E2	Error de validación	Notificar
E3	Error al guardar los datos	Notificar

4.2.12. F10: Administrar comentarios

Definición

Permite ingresar, visualizar y eliminar comentarios de guestbook.

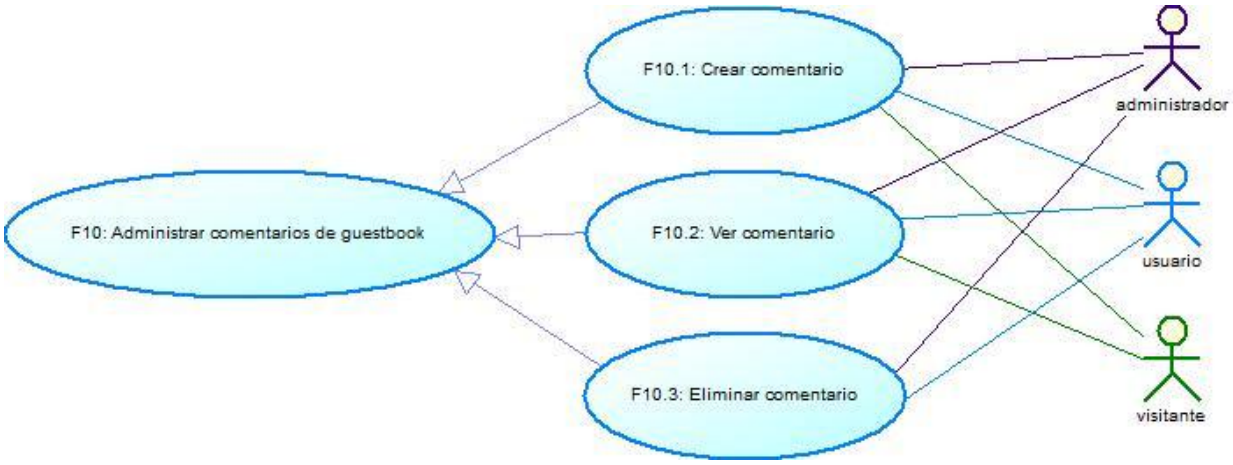
Actores

Administrador, Usuario, Visitante

Precondiciones

Un sitio, layout, página y sección registrados en el sistema

Gráfico



Flujo principal

Escenario 1: crear comentario	Paso	Actor	Paso	Sistema	Excepción
	1	Escribe el url de la página web que contiene el guestbook	2	Recibe la petición	
			3	Busca en la base de datos la página solicitada	E1, E2

			4	Carga las secciones, fotos, videos y otros elementos de la página solicitada, incluyendo el guestbook	
			5	Muestra la página web	
	6	Clic en el botón "Nuevo comentario"	7	Muestra un diálogo con el formulario de creación de comentario	E1
	8	Completa el formulario			
	9	Clic en el botón "Guardar"	10	Valida los datos ingresados	E2
			11	Guarda los datos en la base de datos	E1, E3
			12	Refresca el área de comentarios del guestbook	
Escenario 2: ver comentario	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Escribe el url de la página web que contiene el guestbook	2	Recibe la petición	
			3	Busca en la base de datos la página solicitada	E1, E2
			4	Carga las secciones, fotos, videos y otros elementos de la página solicitada, incluyendo el guestbook	
			5	Muestra la página web, incluyendo los comentarios ingresados en el guestbook	
Escenario 3: eliminar comentario	**Paso**	**Actor**	**Paso**	**Sistema**	**Excepción**
	1	Selecciona "Páginas" en el menú	2	Muestra un submenú con la lista de las páginas existentes	E1
	3	Clic en el botón con el nombre de la página que el guestbook con el comentario a eliminar	4	Muestra la pantalla de edición de páginas web	E1
	5	Selecciona el recuadro que representa la sección	6	Muestra una barra de herramientas con opciones de edición	

	7	Clic en el botón “editar”	8	Abre una nueva ventana en el navegador con la pantalla de edición de guestbook, que contiene los comentarios actualmente ingresados	E1
	9	Clic en el hipervínculo “Eliminar” en la esquina superior derecha del comentario que desee eliminar	10	Muestra un mensaje de confirmación	
	11	Clic en el botón “Aceptar”	12	Elimina el comentario de la base de datos	E1, E2
			13	Eliminar el comentario de la página de visualización	

Flujo alternativo

Escenario 1: crear comentario	**Paso**	**Acción**
	10	Muestra la información de los errores encontrados al validar el formulario
Escenario 2: ver comentario	**Paso**	**Acción**
	5	Muestra una área vacía pues no existen comentarios en el guestbook
Escenario 3: eliminar comentario	**Paso**	**Acción**
	2	Muestra un submenú vacío pues no hay páginas ingresadas

Excepciones

Código	Descripción	Acción
E1	Error de conexión con la base de datos	Notificar
E2	Error de validación	Notificar
E3	Error al guardar los datos	Notificar

4.2.13. F11: Seleccionar menú

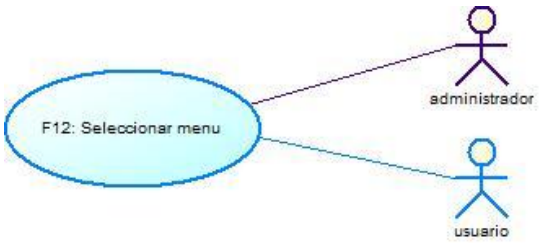
Definición

Permite seleccionar el menú de navegación para el sitio web

Actores

Administrador, usuario

Gráfico



Flujo principal

Paso	Actor	Paso	Sistema	Excepción
1	Clic en el botón “menú” en el menú del sistema	2	Carga la ventana de selección de menú con los diferentes estilos de menú disponibles	
3	Selecciona el estilo de menú que desee utilizar	4	Guarda el estilo seleccionado	E1, E2

Excepciones

Código	Descripción	Acción
E1	Error de conexión con la base de datos	Notificar
E2	Error de inserción	Notificar

4.2.14. F12: Visualizar página web

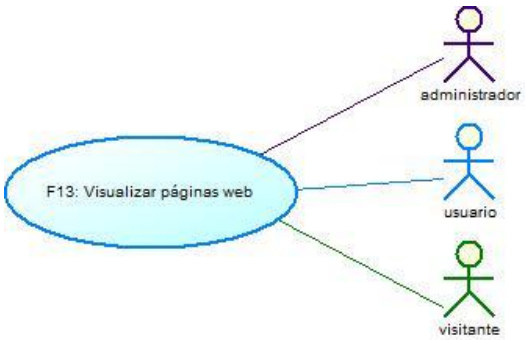
Definición

Permite visualizar, a través de un navegador, una página web registrada en el sistema

Actores

Administrador, usuario, visitante

Gráfico



Flujo principal

Paso	Actor	Paso	Sistema	Excepción
1	Escribe el url de la página web que desea visualizar	2	Recibe la petición	

		3	Busca en la base de datos la página solicitada	E1
		4	Carga las secciones, fotos, videos y otros elementos de la página solicitada	
		5	Muestra la página web	

Flujo alternativo

Paso	Acción
4	No encuentra la página solicitada, muestra la página de error 404

Excepciones

Código	Descripción	Acción
E1	Error de conexión con la base de datos	Notificar

4.3. Requerimientos no funcionales

La solución tendrá las siguientes restricciones:

- El sistema será desarrollado en plataforma Java utilizando Grails.
- El módulo de edición de páginas web funcionará con los navegadores Google Chrome v. 23 y Mozilla Firefox v.17
- La base de datos será Postgresql versión 9.1
- El sistema será desarrollada utilizando el paradigma de orientación a objetos
- Estará construido con un patrón de diseño MVC
- La administración de videos se desarrollará utilizando los servicios de YouTube.
- Los correos electrónicos generados por el sistema serán enviados utilizando los servicios de Gmail o Hotmail.

4.4. Diseño

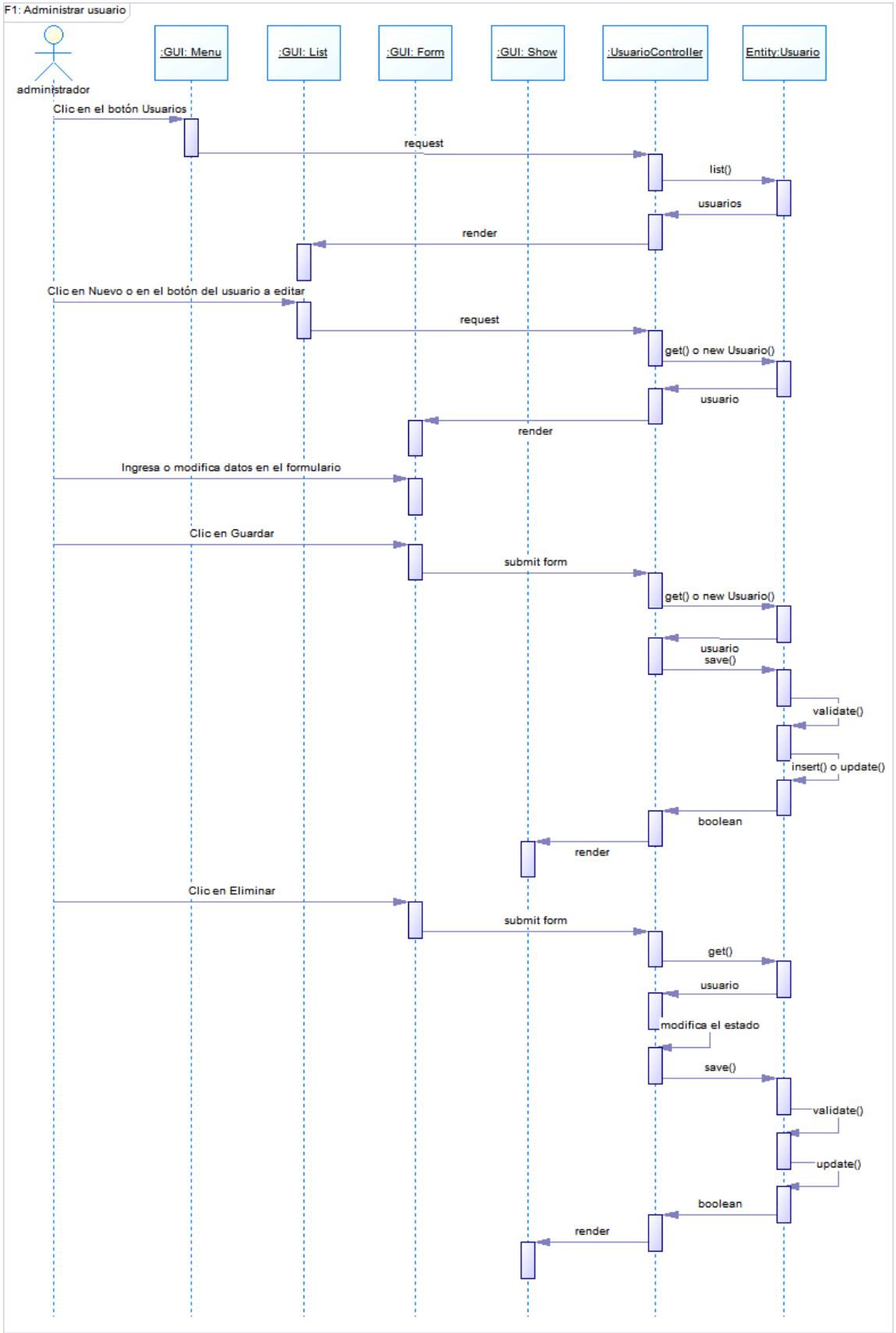
A continuación detallamos el funcionamiento y los componentes del gestor de contenidos.

En esta sección cubriremos el diseño del CMS a través de diagramas de secuencia y especificando las clases de cada una de las tres capaz usadas en un modelo MVC.

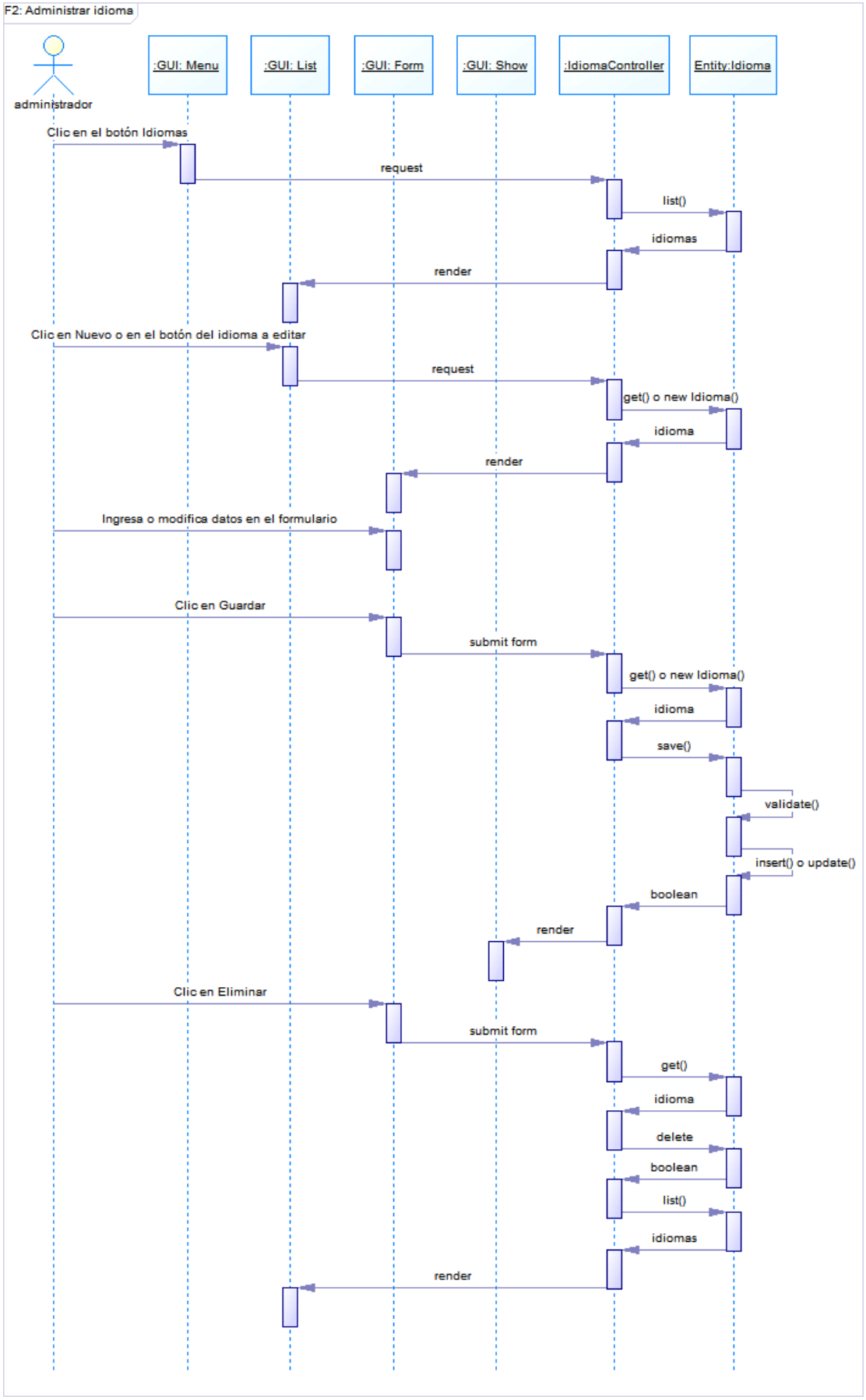
4.4.1. Diagramas de secuencia

La totalidad de diagramas de secuencia en tamaño real se encontrarán en el Anexo2: Diagramas de Secuencia.

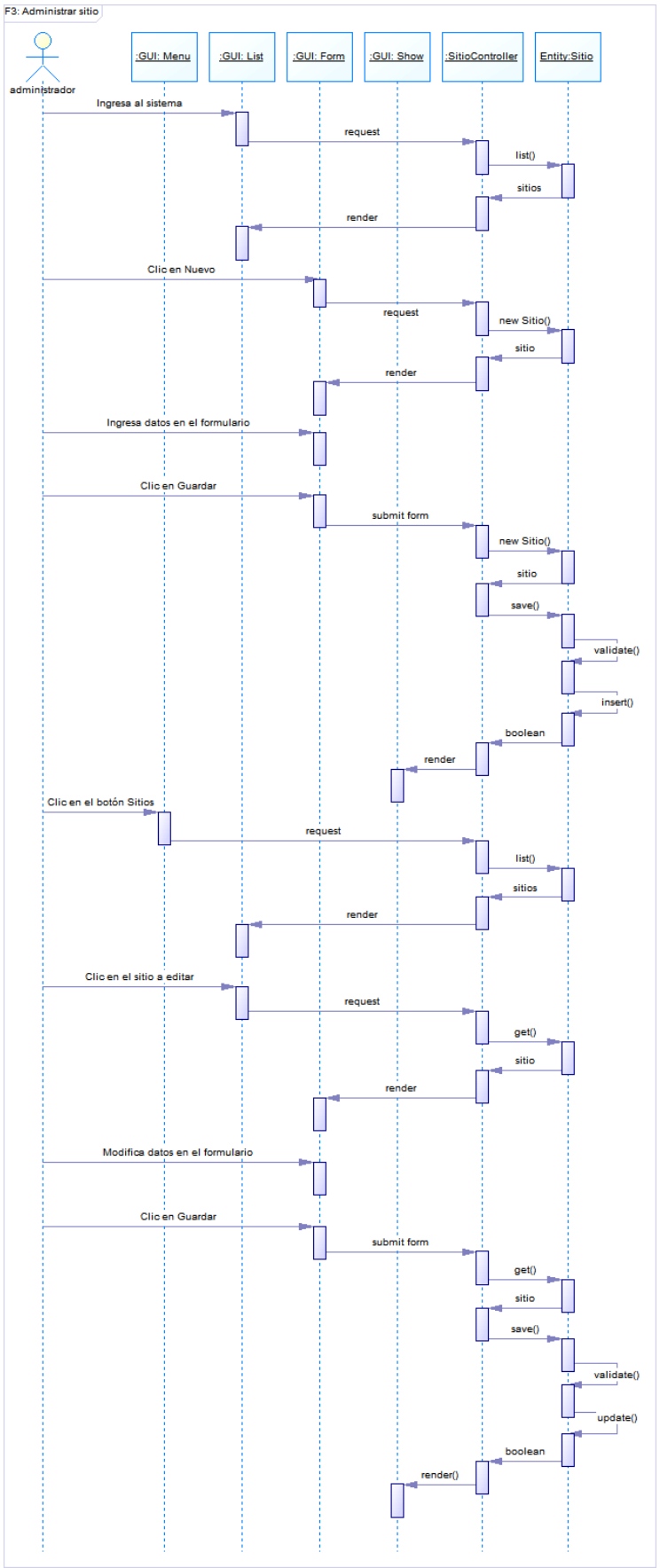
4.4.1.1. F1: Administrar usuario



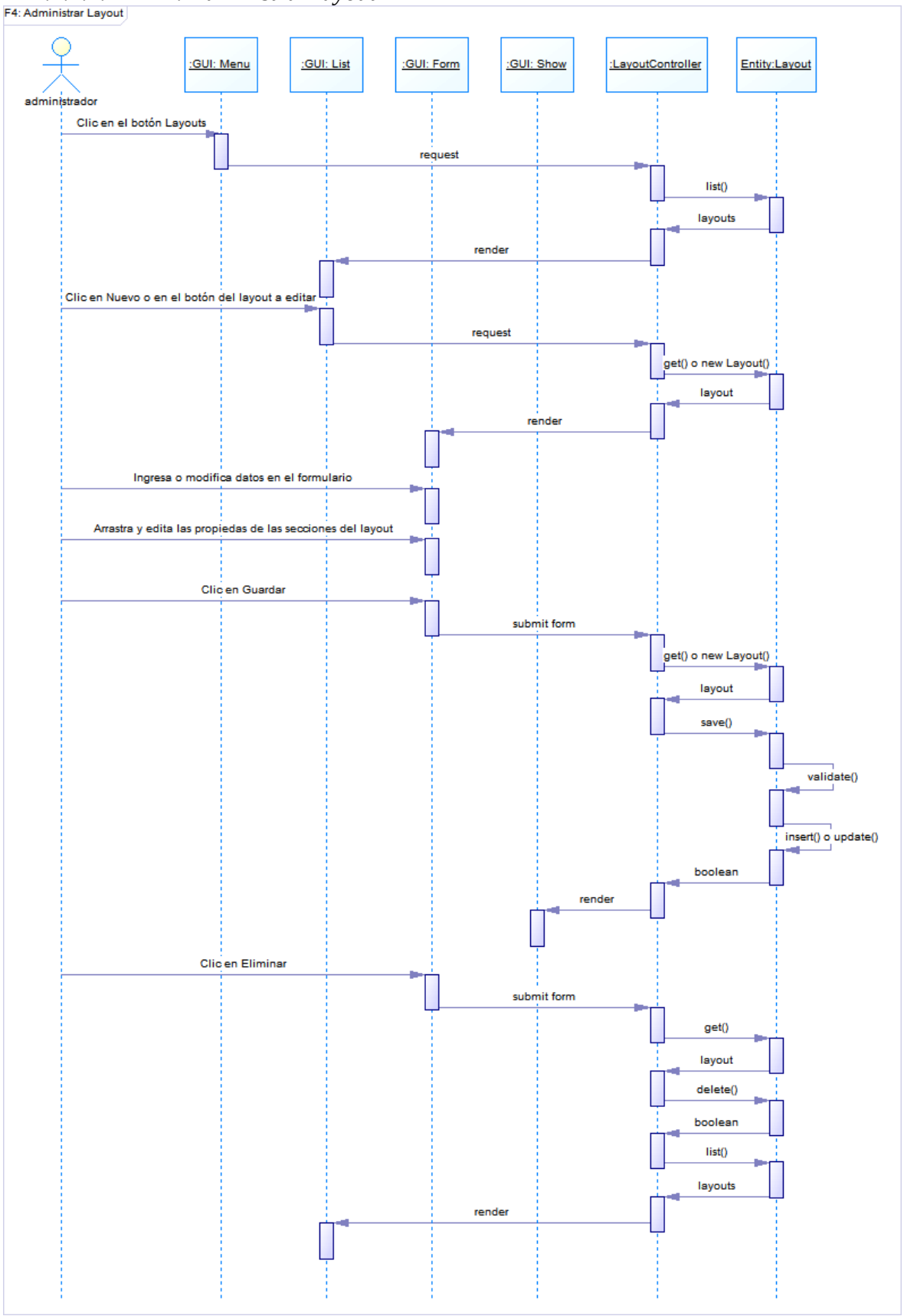
4.4.1.2. F2: Administrar idioma



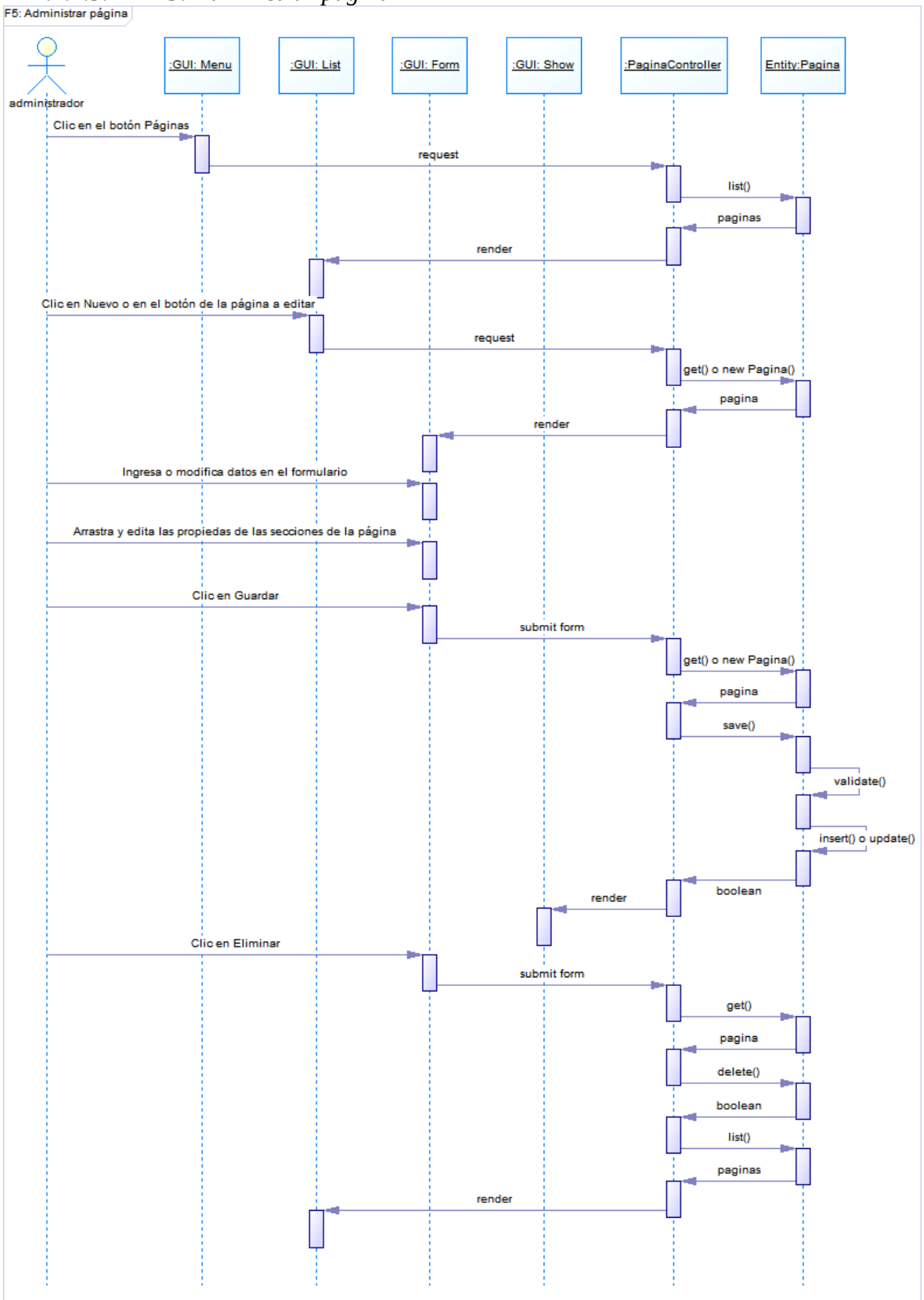
4.4.1.3. F3: Administrar sitio



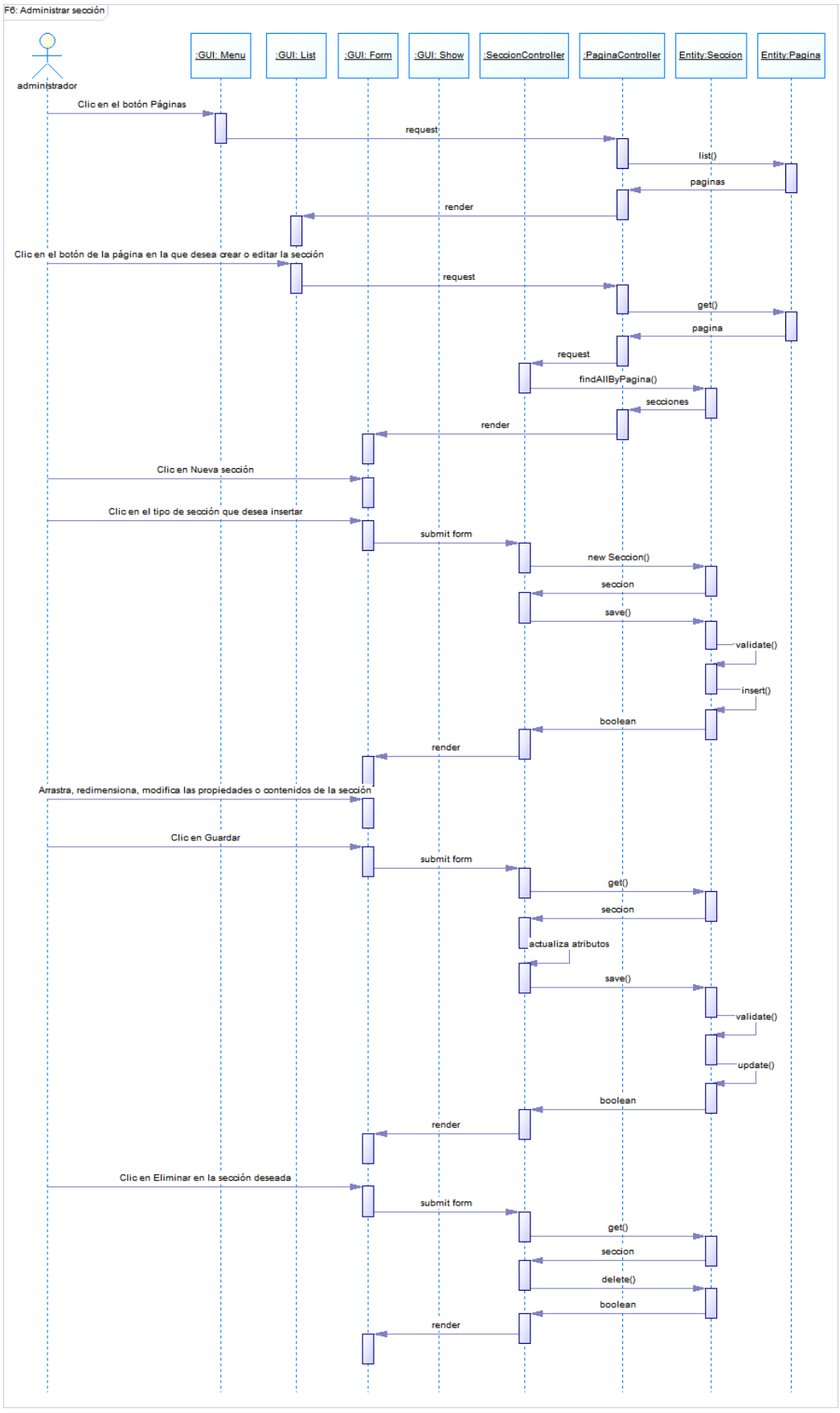
4.4.1.4. F4: Administrar layout



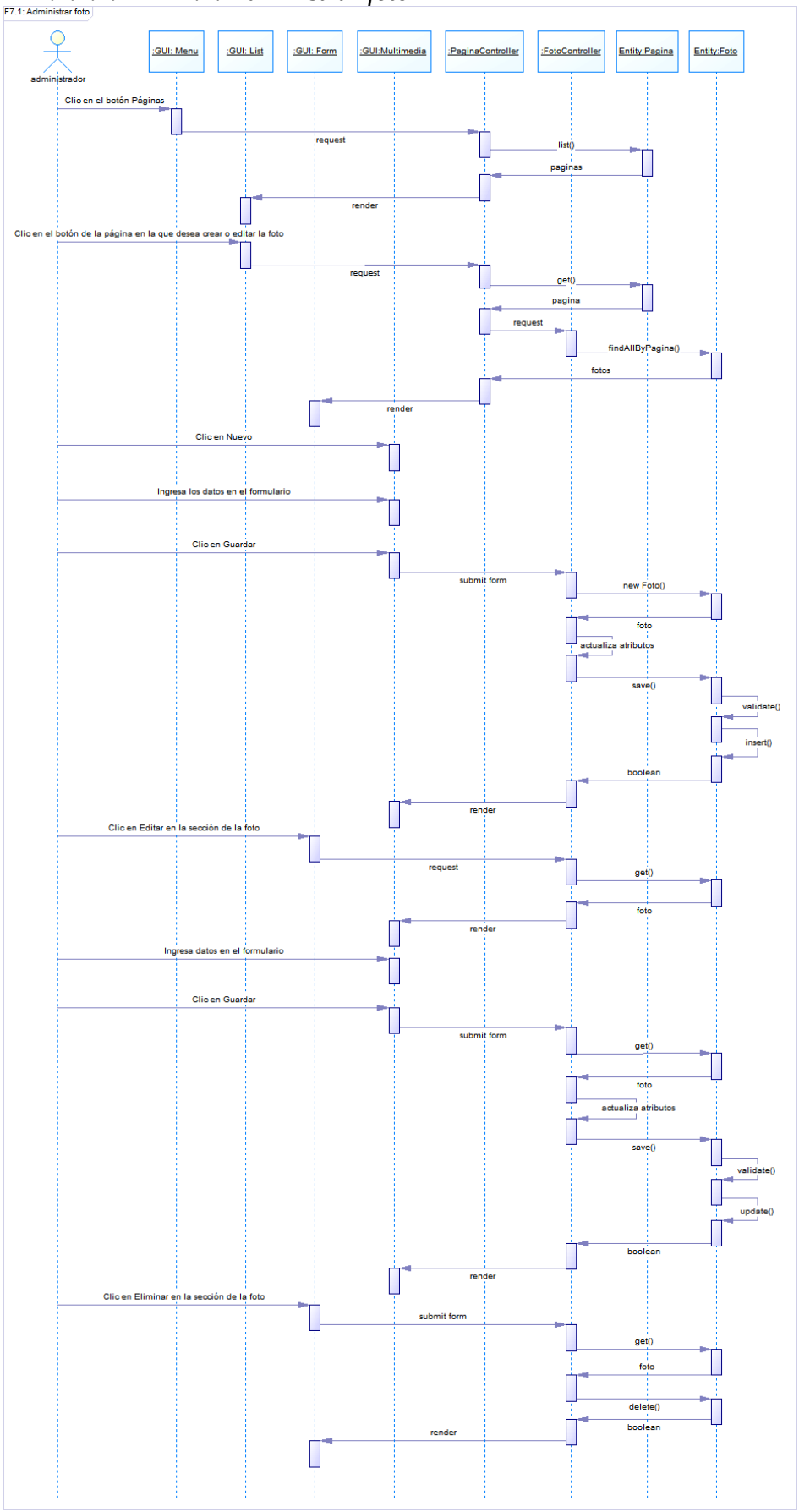
4.4.1.5. F5: Administrar página



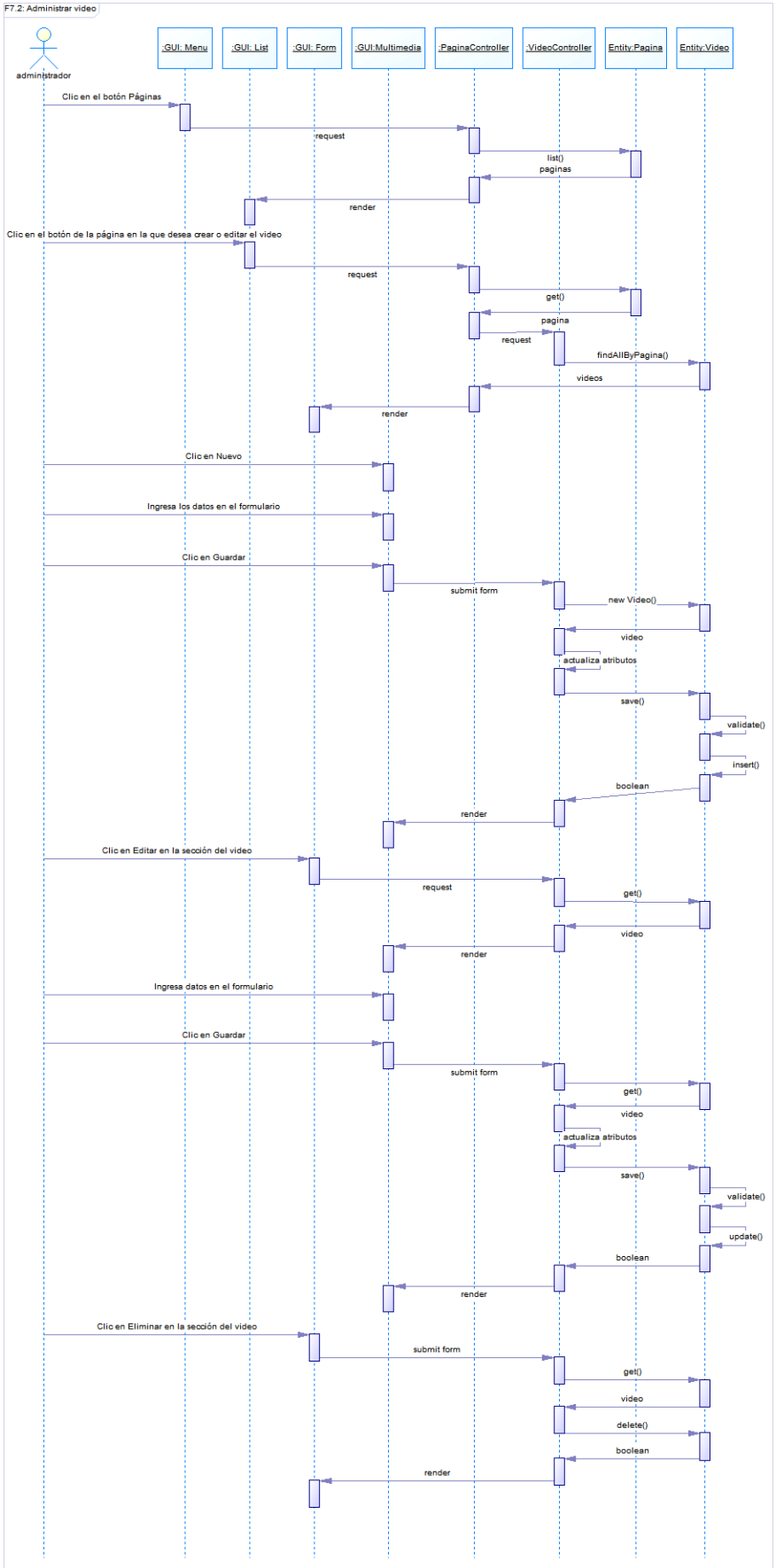
4.4.1.6. F6: Administrar sección



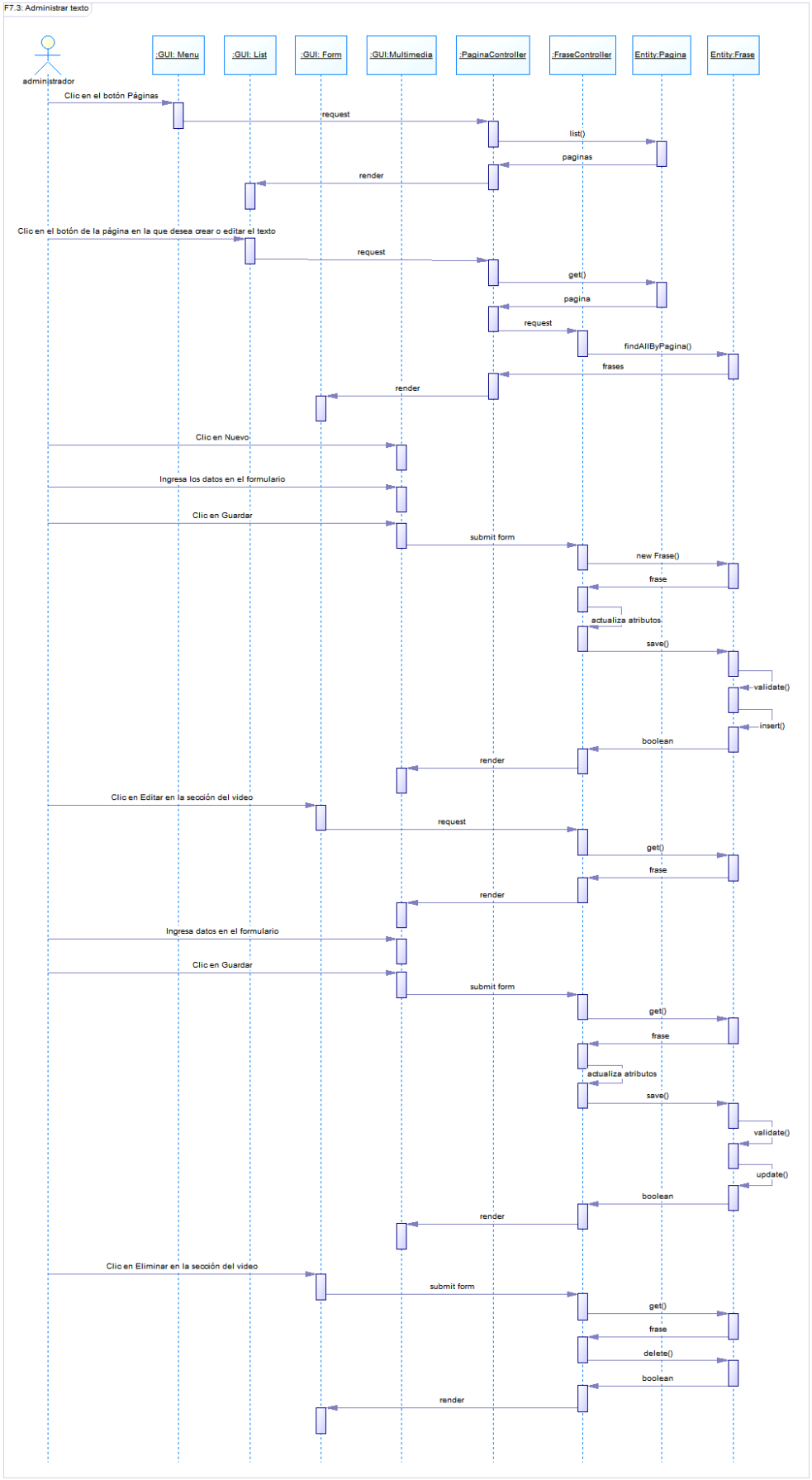
4.4.1.7. F7.1: Administrar foto



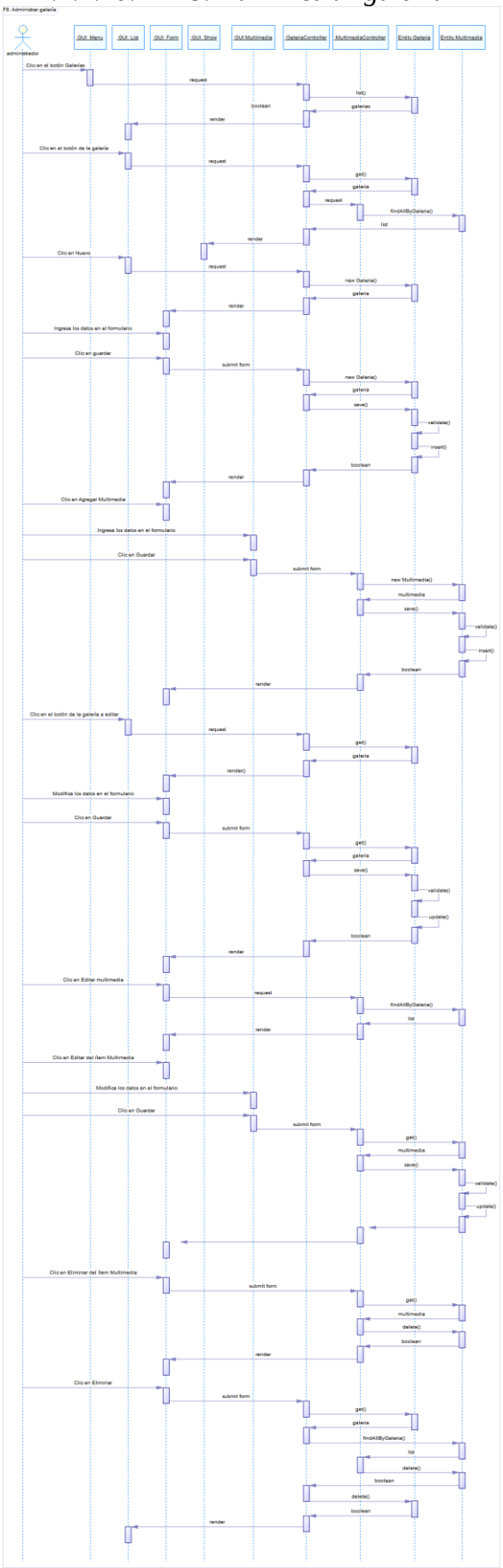
4.4.1.8. F7.2: Administrar video



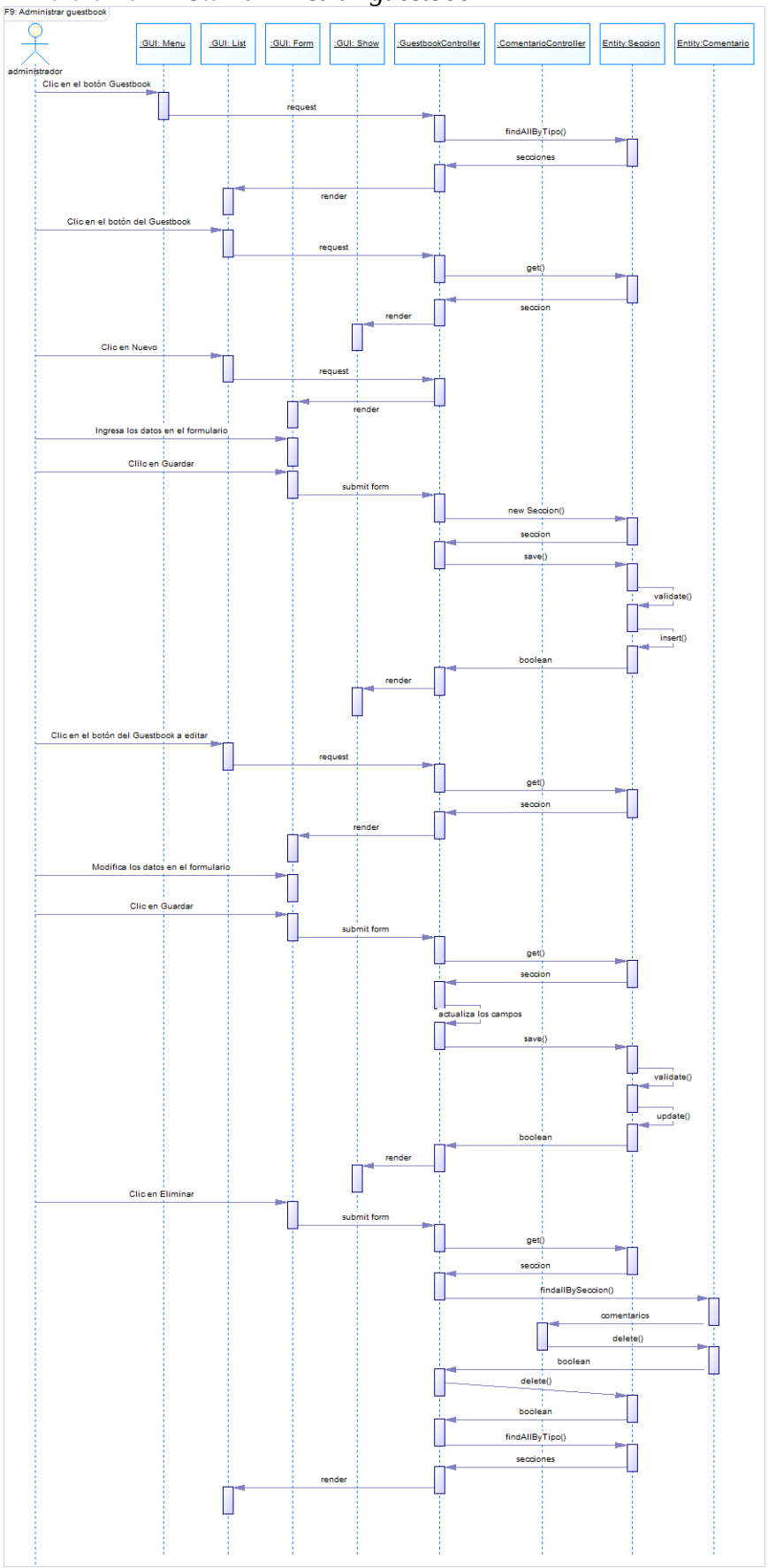
4.4.1.9. F7.3: Administrar texto



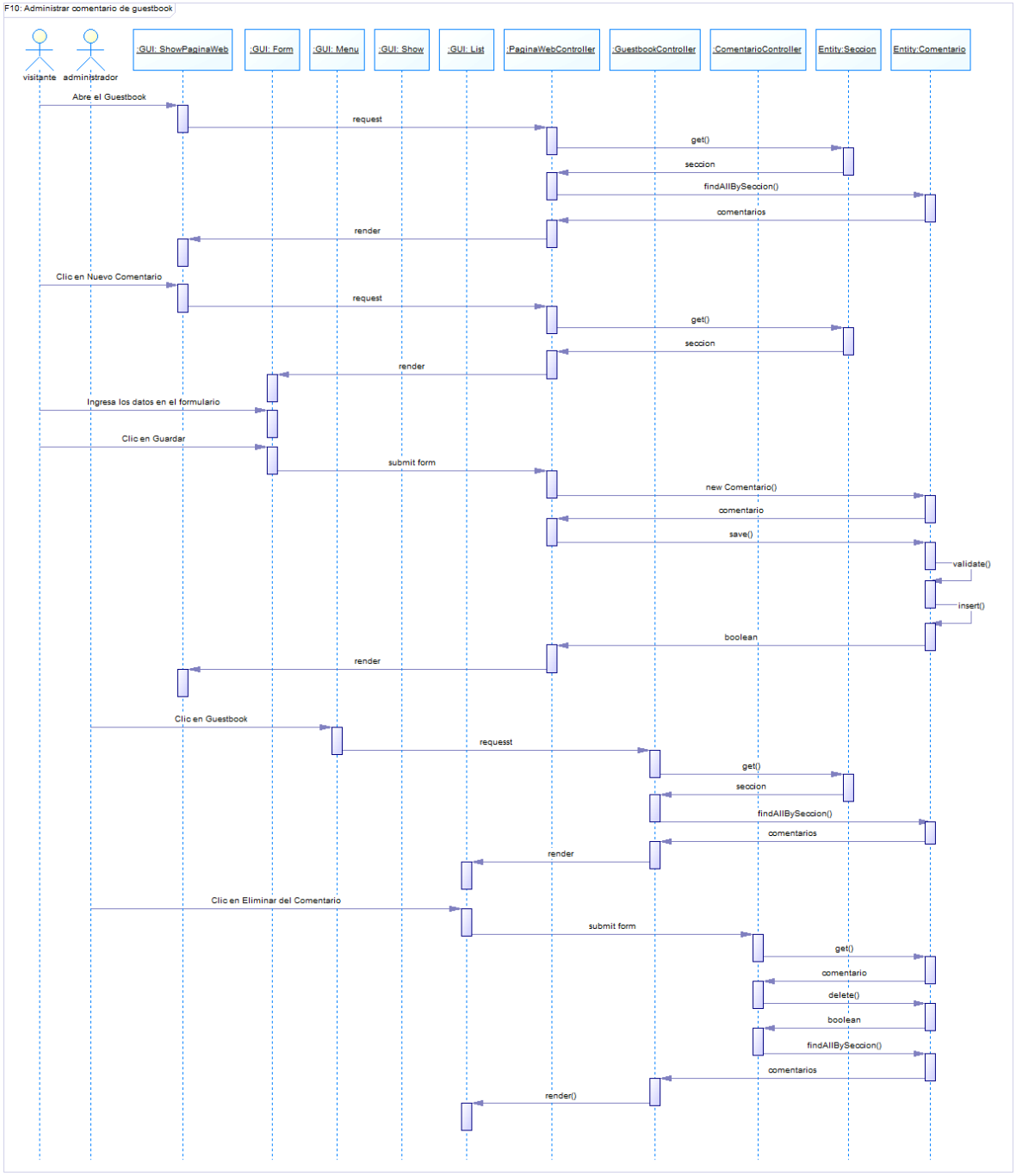
4.4.1.10. F8: Administrar galería



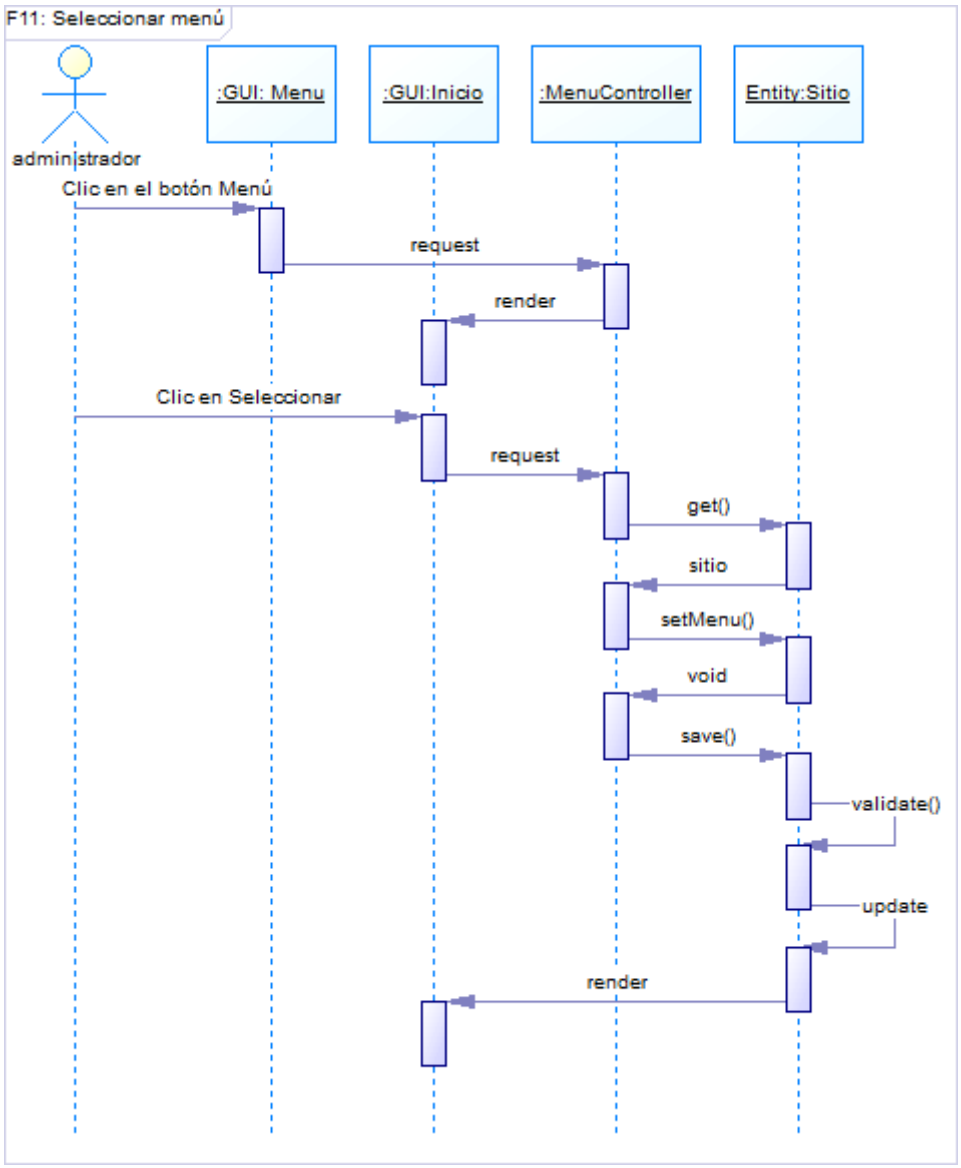
4.4.1.11. F9.: Administrar guestbook



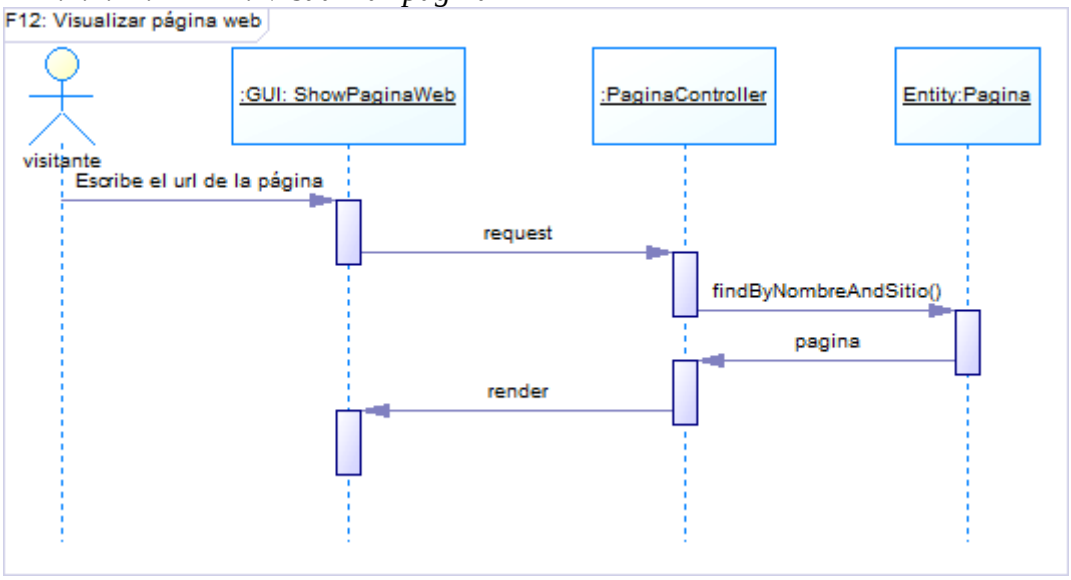
4.4.1.12. F10: Administrar comentario



4.4.1.13. F11: Seleccionar menú



4.4.1.14. F12: Visualizar página



4.4.2. Diseño de clases

4.4.2.1. Diseño de clases UI

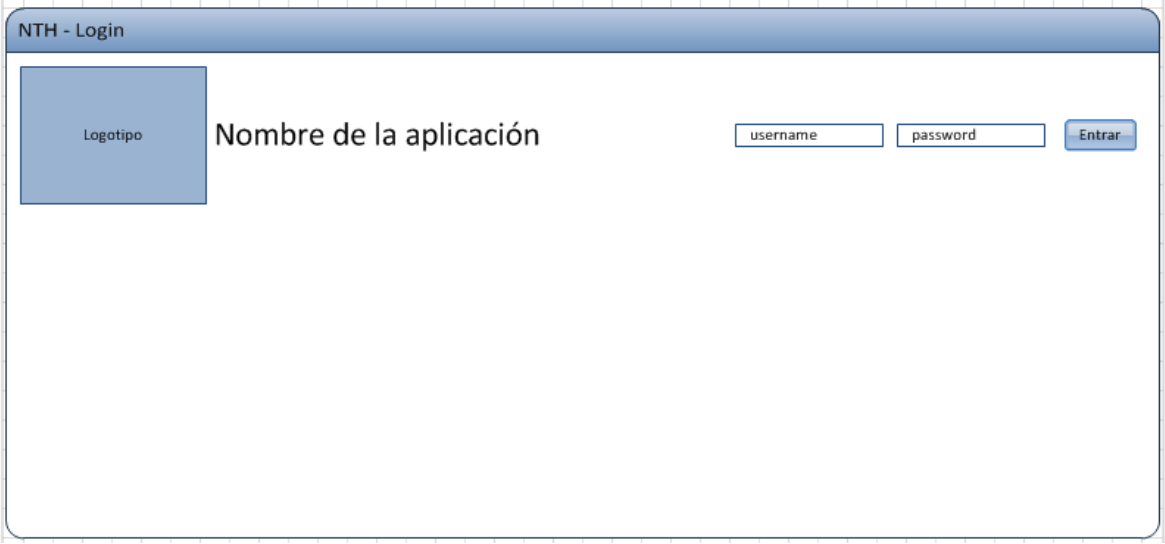
En el presente subcapítulo detallamos clases de interfaz gráfica.

Código	Nombre	Descripción
F0 – GUI – 01	Login	Permite ingresar al sistema
F0 – GUI – 02	Pantalla principal	Pantalla principal del sistema con únicamente el menú principal y el menú flotante
F1 – GUI – 01	Administrar usuarios	Pantalla que permite ver los usuarios ingresados, crear nuevos, modificar y eliminar
F2 – GUI – 01	Administrar idiomas	Pantalla que permite ver los usuarios ingresados, crear nuevos, modificar y eliminar
F3 – GUI – 01	Administrar sitios	Pantalla que permite ver los usuarios ingresados y crear nuevos
F4 – GUI – 01	Administrar layouts	Pantalla que permite ver los layouts ingresados, crear nuevos, modificar y eliminar
F5-F6 – GUI – 01	Administrar páginas y secciones	Pantalla que permite ver las páginas ingresadas, crear nuevas, modificar y eliminar. Además permite editar las secciones de la página representadas por rectángulos que pueden arrastrarse para cambiar su posición y tamaño
F7.1 – GUI – 01	Administrar multimedia – fotos	Pantalla que permite subir una foto al servidor, seleccionando el archivo, o modificar las propiedades de una sección de tipo foto.
F7.2 – GUI – 01	Administrar multimedia – videos – subir video	Pantalla que permite subir un video a una cuenta de YouTube
F7.2 – GUI – 02	Administrar multimedia – videos – seleccionar video	Pantalla que permite seleccionar un video de una cuenta de YouTube para ubicarlo en la página

F7.3 – GUI – 01	Administrar multimedia – texto – Texto	Pantalla que permite editar el contenido de una sección de texto con la ayuda de un editor de texto con formato
F7.3 – GUI – 02	Administrar multimedia – texto – Link	Pantalla que permite editar el contenido de una sección de texto con la ayuda de un editor de texto con formato, además de seleccionar una página existente para crear un link
F7.3 – GUI – 03	Administrar multimedia – texto – Descarga	Pantalla que permite editar el contenido de una sección de texto con la ayuda de un editor de texto con formato, además de seleccionar un archivo para subir al servidor y crear un link de descarga
F7.3 – GUI – 04	Administrar multimedia – texto – Contenido	Pantalla que permite seleccionar una página existente para ubicar todo el contenido de ésta en una sección de la página
F7.3 – GUI – 05	Administrar multimedia – texto – Contacto	Pantalla que permite crear un formulario para que el visitante envíe un e-mail al administrador de la página
F8 – GUI – 01	Administrar galerías – Selección de tipo	Pantalla de selección del tipo de galería que se va a crear. Puede ser texto o fotos
F8 – GUI – 02	Administrar galerías – Edición de multimedia	Pantalla que permite seleccionar el multimedia que irá en la galería.
F8 – GUI – 03	Administrar galerías – modificar propiedades	Pantalla que permite modificar las propiedades de la galería. Esto permite modificar la manera en la que se muestra la galería a los visitantes
F8 – GUI – 04	Seleccionar Galería	Pantalla que permite seleccionar una galería creada previamente para insertarla en una sección de la página.
F9-F10 – GUI – 01	Administrar guestbook y comentarios	Pantalla que permite editar las etiquetas que ve el usuario de la página web cuando va a crear un nuevo comentario. También se .pueden eliminar comentarios
F10 – GUI – 02	Nuevo comentario	Pantalla vista por el visitante de la página web y que permite dejar un comentario, una queja, una sugerencia, etc. en un guestbook o libro de visitas previamente configurado en el sitio.
F11 – GUI – 01	Seleccionar menú	Pantalla que permite cambiar la manera en que se muestra en menú
F12 – GUI – 01	Visualizar página web	Esta es la pantalla que visualiza el usuario de la página web, es decir, quien entra a ver, sin opción a editar nada

4.4.2.1.1 F0 – GUI – 01 – Login

Permite ingresar al sistema. Muestra el formulario de ingreso de credenciales. Al hacer clic en el botón Entrar se valida la información ingresada y, de estar correcta, ingresa al sistema.

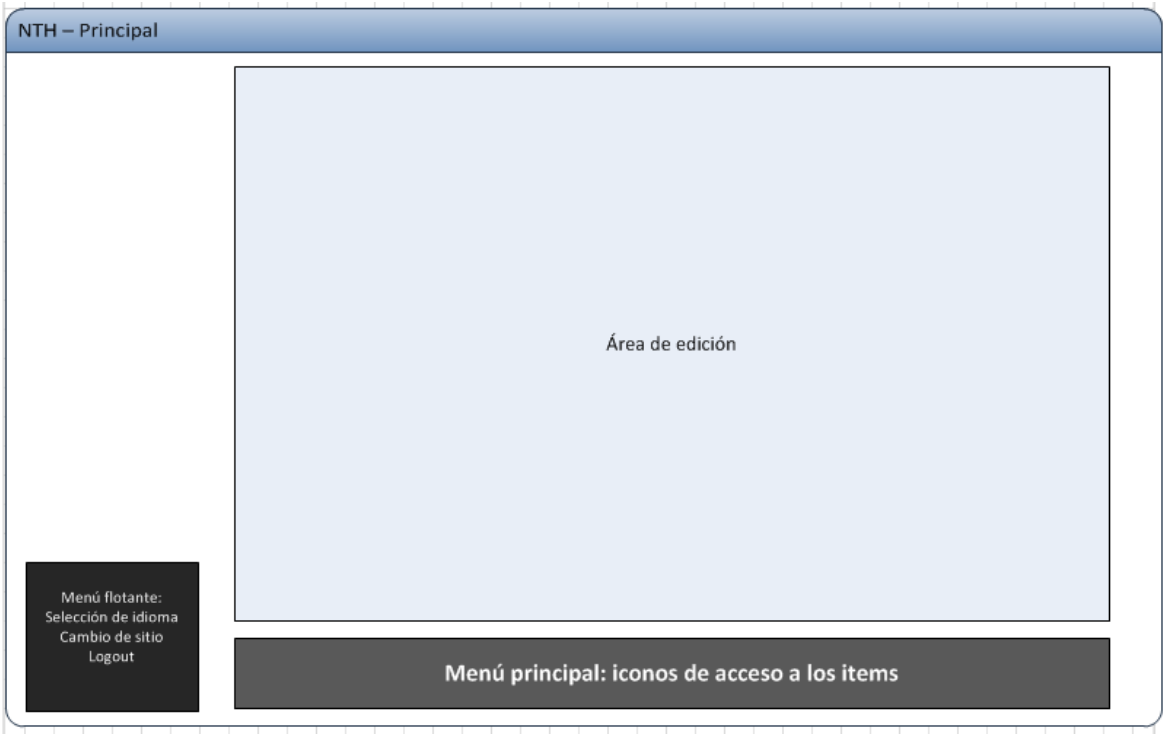


Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Entrar	Hace submit del formulario, valida las credenciales, ingresa al sistema y muestra la pantalla principal

4.4.2.1.2 F0 – GUI – 02 – Pantalla principal

Es la pantalla principal del sistema. Contiene únicamente el menú principal y el menú flotante con opciones de selección de idioma, de cambio de sitio y botón para salir del sistema.

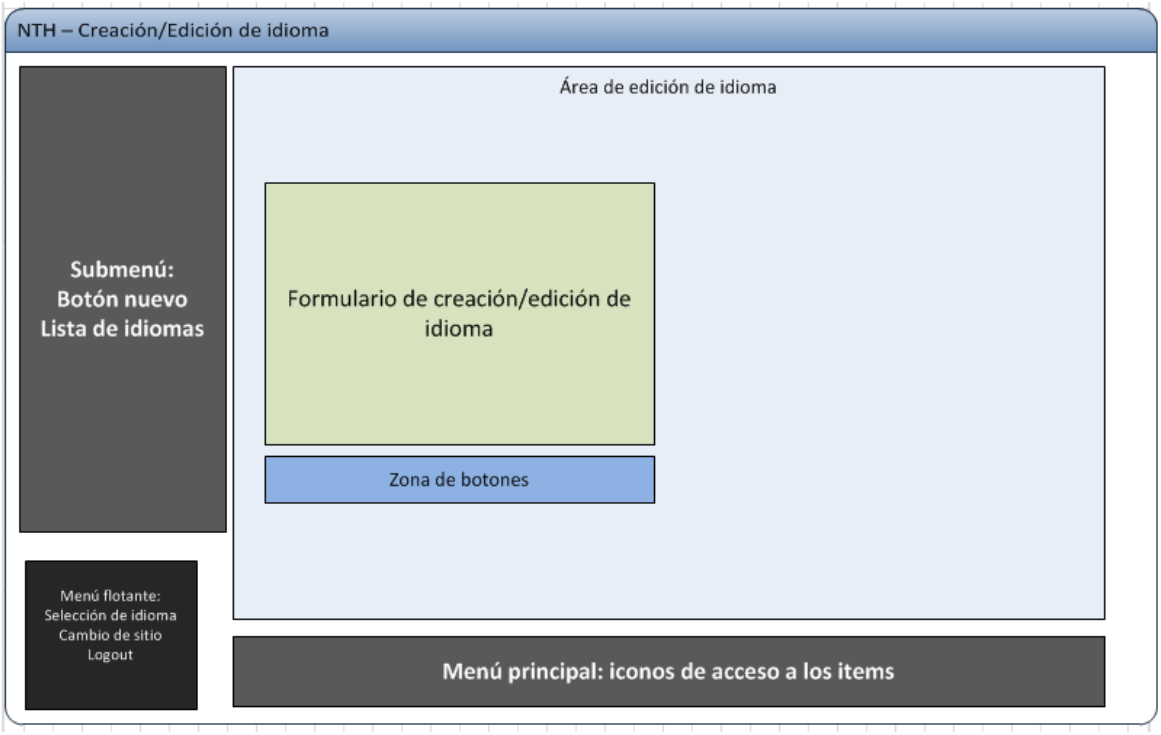


Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Salir	Sale del sistema
	Cambiar	Muestra la pantalla de selección de sitio
	Menú principal	Botones para acceder a las funcionalidades del sistema: Layouts, Páginas, Galerías, Idiomas, Menú, Cuenta y Usuarios

4.4.2.1.3 F1 – GUI – 01 – Administrar usuarios

Muestra la pantalla de administración de usuarios. En el submenú se cargan los usuarios existentes en forma de botones. Al hacer clic en uno de estos botones se muestra el formulario con los datos del usuario precargados.

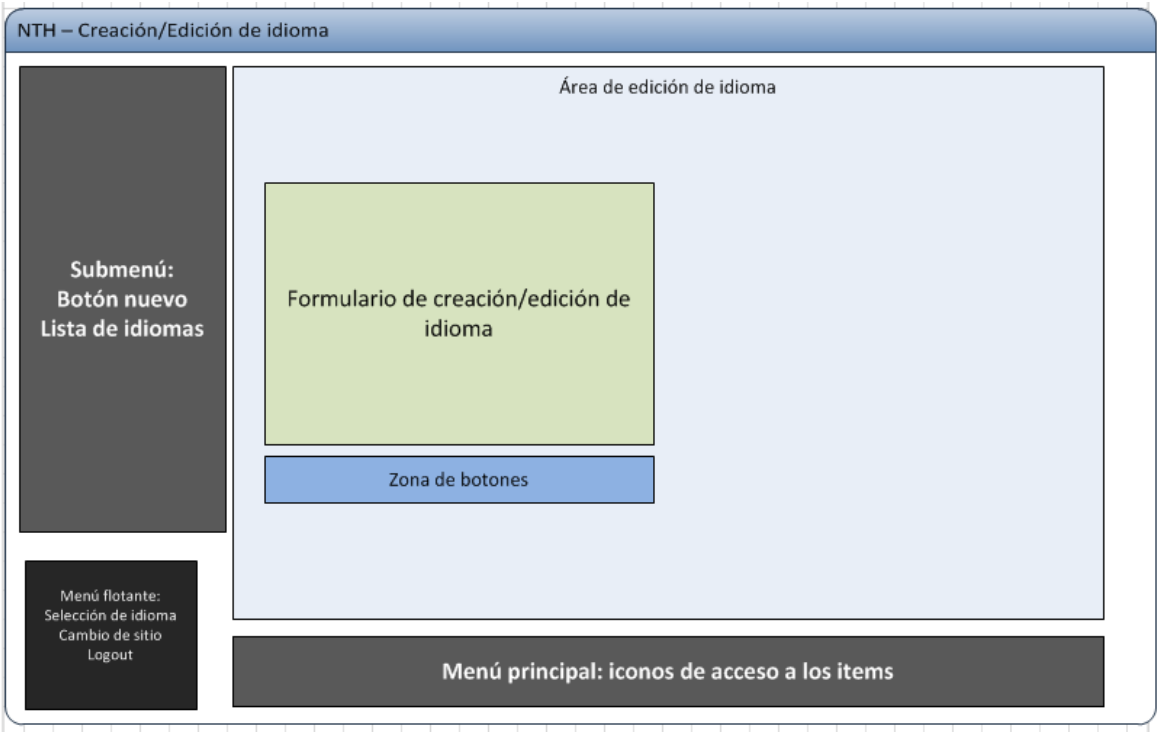


Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Salir	Sale del sistema
	Cambiar	Muestra la pantalla de selección de sitio
	Menú principal	Botones para acceder a las funcionalidades del sistema: Layouts, Páginas, Galerías, Idiomas, Menú, Cuenta y Usuarios
	Nuevo	Muestra el formulario de creación/edición vacío para crear registro
	Botones de lista	Un botón por cada registro en la base de datos. Muestra el formulario de creación/edición con los datos del registro seleccionado
	Guardar	Guarda el registro, ya sea creando uno nuevo o actualizando uno existente
	Eliminar	Elimina el registro de la base de datos

4.4.2.1.4 F2 – GUI – 01 – Administrar idiomas

Muestra la pantalla de administración de idiomas. En el submenú se cargan los idiomas existentes en forma de botones. Al hacer clic en uno de estos botones se muestra el formulario con los datos del idioma precargados.

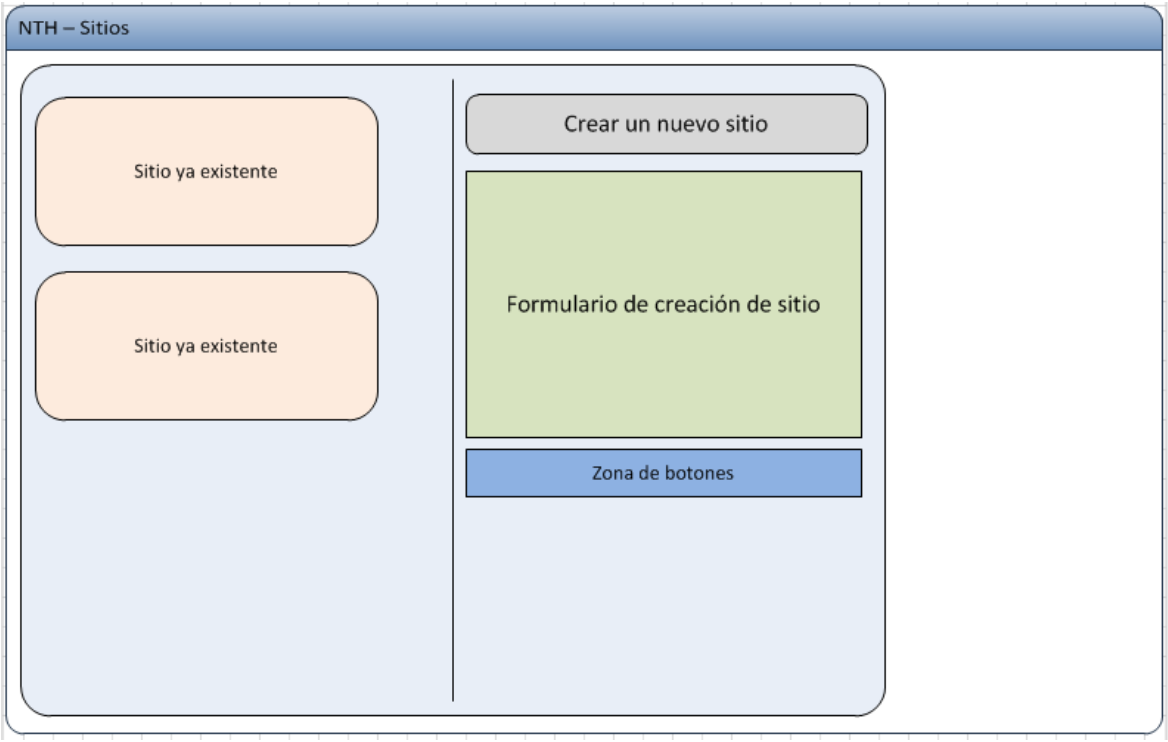


Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Salir	Sale del sistema
	Cambiar	Muestra la pantalla de selección de sitio
	Menú principal	Botones para acceder a las funcionalidades del sistema: Layouts, Páginas, Galerías, Idiomas, Menú, Cuenta y Usuarios
	Nuevo	Muestra el formulario de creación/edición vacío para crear registro
	Botones de lista	Un botón por cada registro en la base de datos. Muestra el formulario de creación/edición con los datos del registro seleccionado
	Guardar	Guarda el registro, ya sea creando uno nuevo o actualizando uno existente
	Eliminar	Elimina el registro de la base de datos

4.4.2.1.5 F3 – GUI – 01 – Administrar sitios

Muestra la pantalla de administración de usuarios. En el área izquierda se cargan los sitios existentes en forma de botones, en el área derecha se muestra un formulario para crear un sitio nuevo. Al hacer clic en uno de los botones de sitio existente se muestra la pantalla principal del sistema.

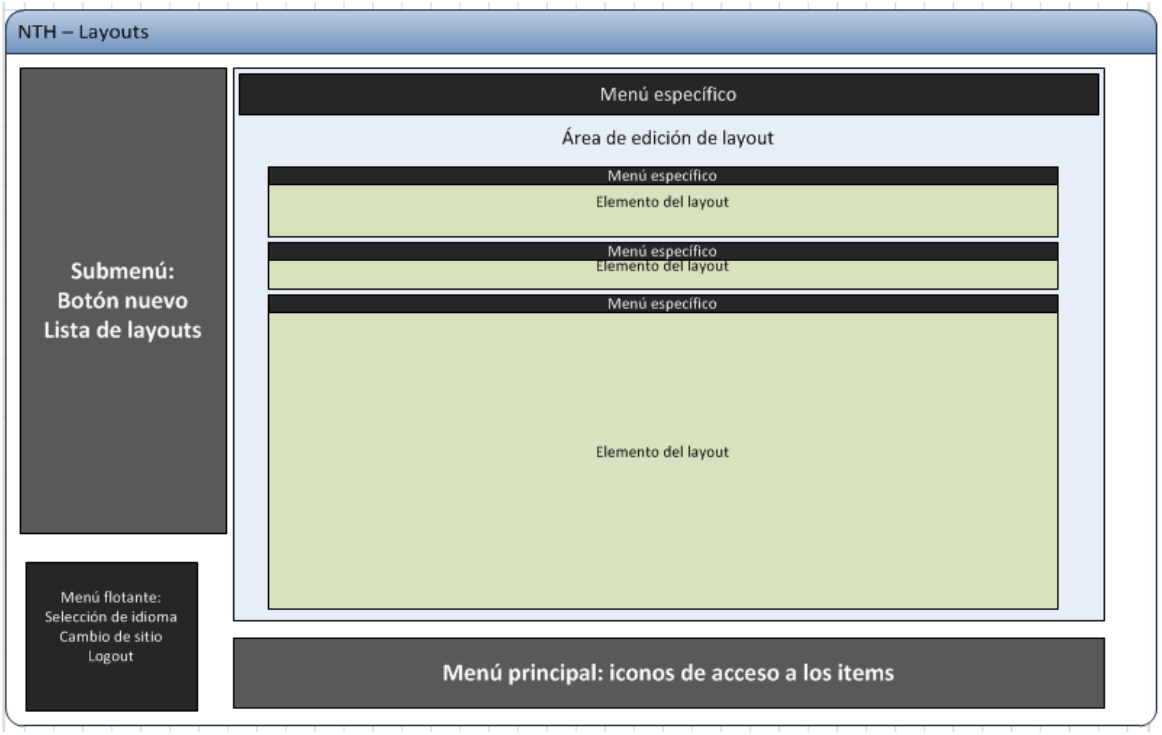


Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Nuevo	Muestra el formulario de creación/edición vacío para crear registro
	Botones de lista	Un botón por cada registro en la base de datos. Muestra el formulario de creación/edición con los datos del registro seleccionado
	Guardar	Guarda el registro, ya sea creando uno nuevo o actualizando uno existente

4.4.2.1.6 F4 – GUI – 01 – Administrar layouts

Muestra la pantalla de edición de layouts. Los layouts existentes se cargan en el submenú en forma de botones. En el área de edición se muestran las diferentes secciones (banner, menú, selección de idioma, contenido principal). Estas secciones son representadas por rectángulos que pueden arrastrarse para cambiar su posición y tamaño.

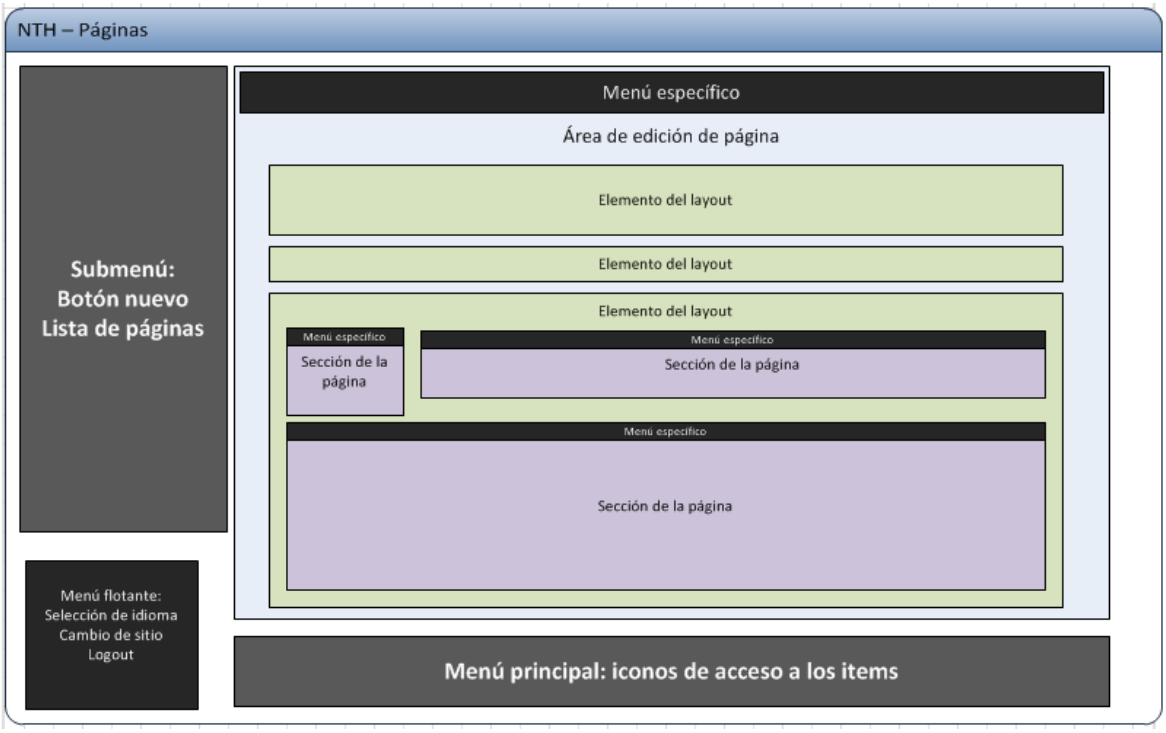


Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Salir	Sale del sistema
	Cambiar	Muestra la pantalla de selección de sitio
	Menú principal	Botones para acceder a las funcionalidades del sistema: Layouts, Páginas, Galerías, Idiomas, Menú, Cuenta y Usuarios
	Nuevo	Muestra la pantalla de creación/edición de layouts
	Botones de lista	Un botón por cada registro en la base de datos. Muestra la pantalla de creación/edición de layouts
	Menú específico	Botones para acceder a las funciones específicas de creación y edición de layout: • **Guardar:** Guarda el layout, ya sea creando un nuevo registro o modificando uno existente • **Eliminar:** Elimina el layout • **Reset:** Reinicializa las áreas a su estado de defecto • **Propiedades:** Abre una lista de propiedades que se pueden modificar • **Crear página:** Crea una página con el layout seleccionado

4.4.2.1.7 F5-F6 – GUI – 01 – Administrar páginas y secciones

Muestra la pantalla de edición de páginas. Las páginas existentes se cargan en el submenú en forma de botones. En el área de edición se muestran las diferentes secciones del layout y de la página (banner, menú, selección de idioma, contenido principal en el caso de layout, fotos, videos, textos). Estas secciones son representadas por rectángulos que pueden arrastrarse para cambiar su posición y tamaño. En esta pantalla, las secciones del layout se muestran para tener una referencia pero no pueden ser modificadas, pero se puede agregar secciones de los diferentes tipos a la página, así como editarlas y eliminarlas.



Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Salir	Sale del sistema
	Cambiar	Muestra la pantalla de selección de sitio
	Menú principal	Botones para acceder a las funcionalidades del sistema: Layouts, Páginas, Galerías, Idiomas, Menú, Cuenta y Usuarios
	Nuevo	Muestra la pantalla de creación/edición de página
	Botones de lista	Un botón por cada registro en la base de datos. Muestra la pantalla de creación/edición de página
	Menú específico	Botones para acceder a las funciones específicas de

	de página	creación y edición de página: • **Guardar:** Guarda la página, ya sea creando un nuevo registro o modificando uno existente • **Ver página:** Muestra la página que se está editando en una nueva ventana del navegador • **Insertar sección:** Permite seleccionar un tipo de sección para insertarla en la página • **Eliminar:** Elimina a página • **Editar diseño:** Muestra la pantalla de edición de layout con el layout de la página actual precargado • **Menú:** Permite editar el texto que muestra el menú para acceder a la página y el orden.
	Menú específico de sección	Botones para acceder a las funciones específicas de edición de sección: • **Propiedades:** Abre una lista de propiedades que se pueden modificar • **Editar:** Muestra una pantalla específica según el tipo de sección para modificar su contenido (ver pantallas de foto, video, texto, galería, guestbook) • **Eliminar:** Elimina la sección y su contenido (es decir, elimina también foto, video, texto según sea el caso)

4.4.2.1.8 F7.1 – GUI – 01 – Administrar multimedia – fotos

La administración de fotos se abre en una pantalla aparte, desde otra pantalla, como la de galería o sección. En esta pantalla se puede subir una foto al servidor, seleccionando el archivo, o modificar las propiedades de una sección de tipo foto.

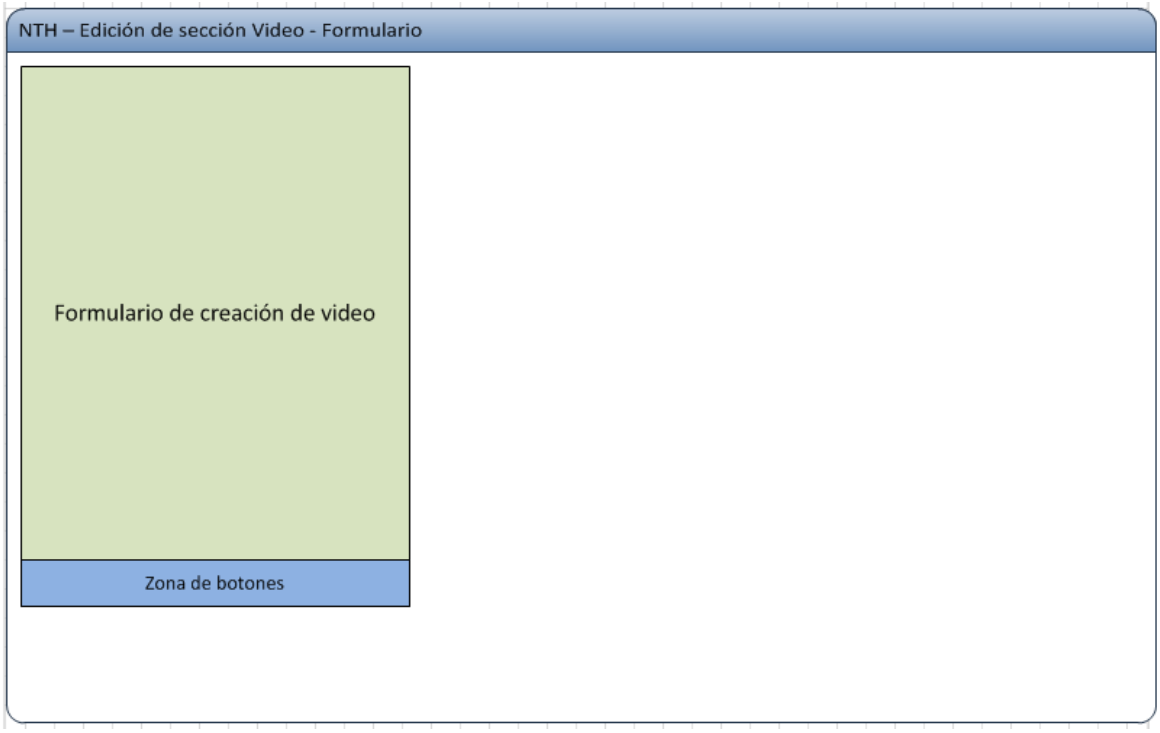


Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Guardar	Guarda la foto en la base de datos, actualiza la sección en la base de datos y refresca la página para mostrar la foto en la sección correspondiente. Cierra la ventana de edición.

4.4.2.1.9 F7.2 – GUI – 01 – Administrar multimedia – videos – subir video

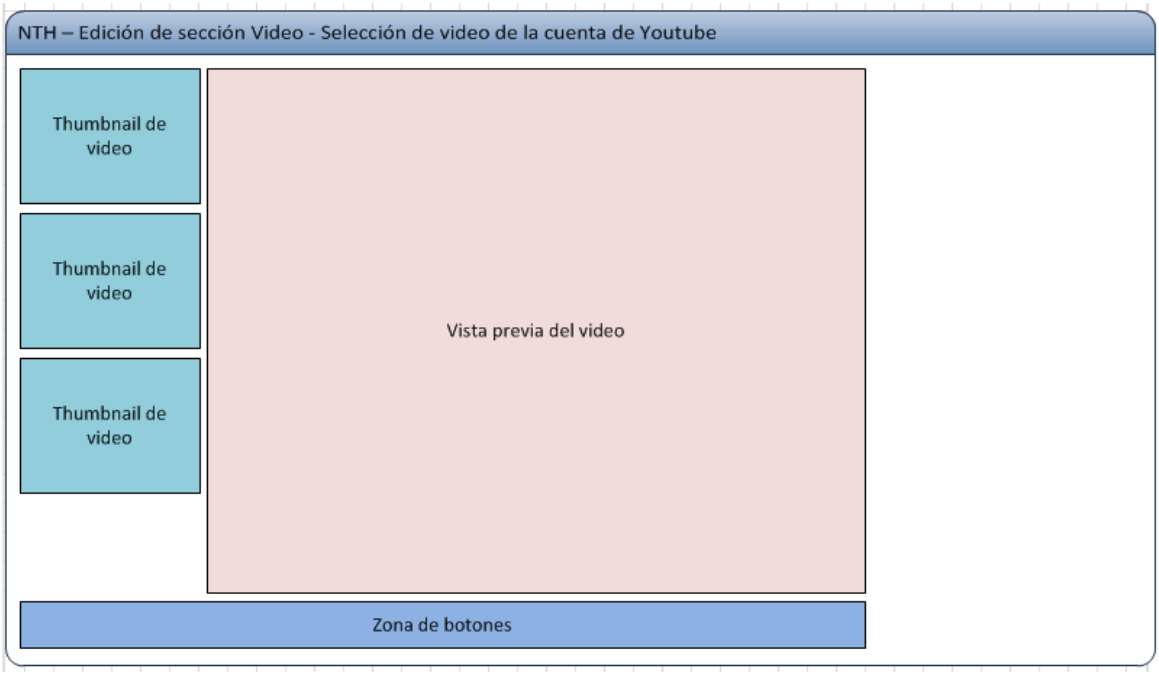
La administración de videos se abre en una pantalla aparte, desde otra pantalla, como la de galería o sección. En esta pantalla se puede subir un video a una cuenta de YouTube, o seleccionar un video de esta cuenta para ubicarlo en la página.



Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Guardar	Guarda el video y su información en línea utilizando una cuenta de Youtube y muestra la pantalla de selección de video

4.4.2.1.10 F7.2 – GUI – 02 – Administrar multimedia – videos – seleccionar video



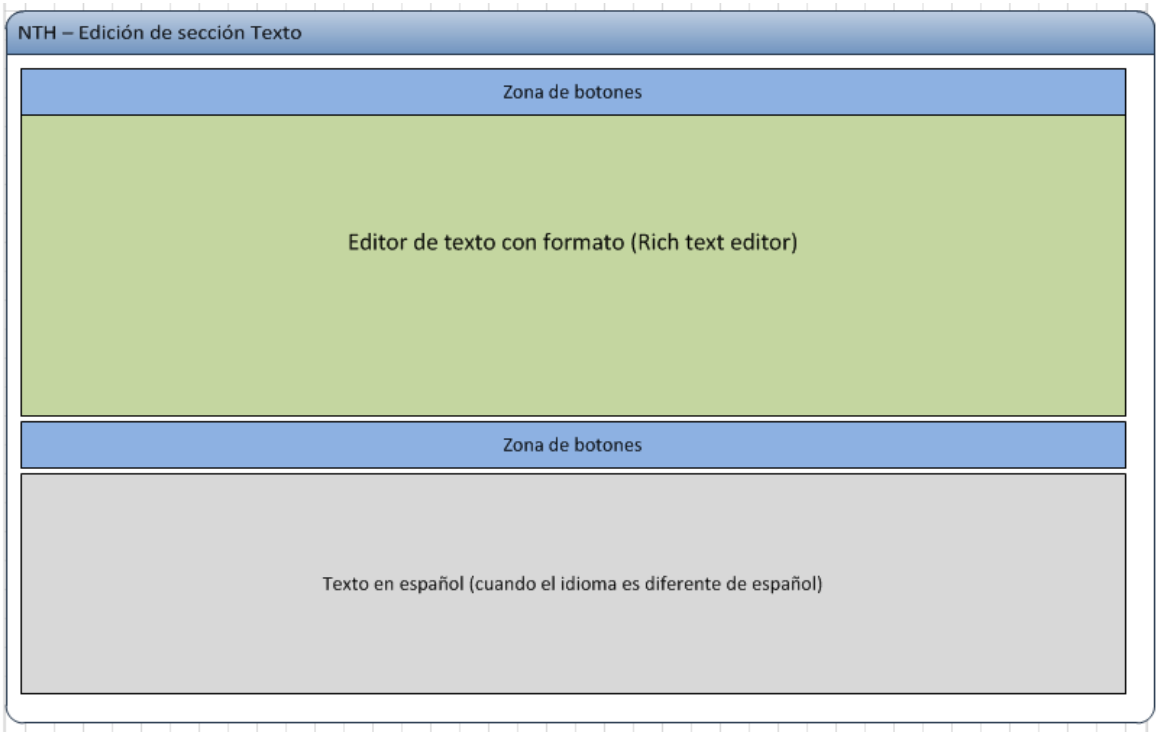
Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Botones de lista	Un botón por cada registro en la base de datos. Muestra el video en el área de previsualización
	Seleccionar	Selecciona el video. Actualiza la sección en la base de datos y refresca la página para mostrar el video seleccionado en la sección correspondiente. Cierra la ventana de edición.

4.4.2.1.11 F7.3 – GUI – 01 – Administrar multimedia – texto – Texto

La administración de texto se abre en una pantalla aparte, desde otra pantalla, como la de sección. En esta pantalla se puede editar el contenido de una sección de texto con la ayuda de un editor de texto con formato. Este editor contiene botones para dar diferentes formatos al texto, como subrayado, negritas, itálicas, etc. y también para modificar su tamaño, tipo de letra, color, etc. La edición de texto incluye también la edición de links, links de descarga, contenido (inclusión de una página ya existente), contacto (formulario que permite al usuario de la página web comunicarse a través de un e-mail con el administrador). En todos los casos, cuando el idioma es diferente a español, se muestra en el área inferior de la pantalla el contenido en español para facilitar la traducción.

La edición de texto incluye únicamente el editor de texto con formato, el botón de Guardar y el texto en español en el caso en que se esté editando el contenido en otro idioma.

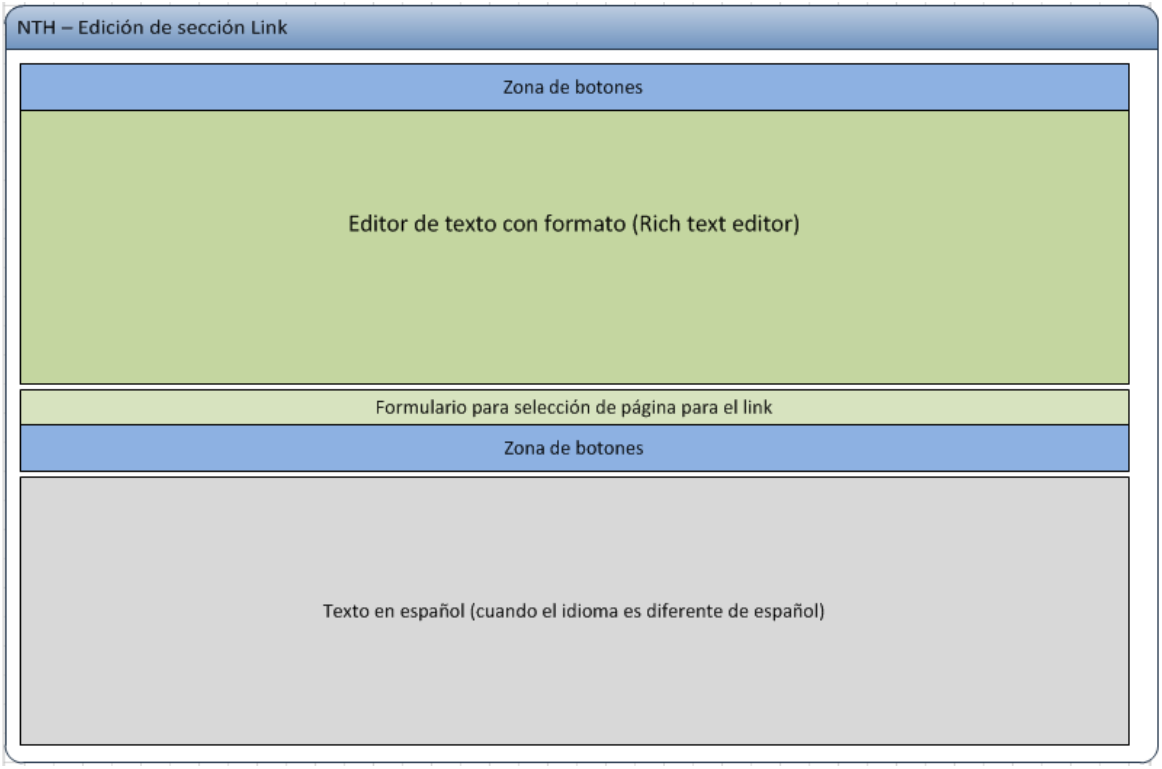


Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Botones de edición de texto	Botones que simulan la interfaz de un editor de texto, contiene entonces todas las funciones propias de uno, como negrillas, itálicas, subrayar, etc.
	Guardar	Guarda el texto en la base de datos, actualiza la sección y refresca la página para mostrar el texto en la sección correspondiente. Cierra la ventana de edición.

4.4.2.1.12 F7.3 – GUI – 02 – Administrar multimedia – texto – Link

La edición de link incluye el editor de texto con formato, el botón de Guardar, el texto en español en el caso en que se esté editando el contenido en otro idioma, además de un selector de páginas ya existentes, a la cual redireccionará el link.

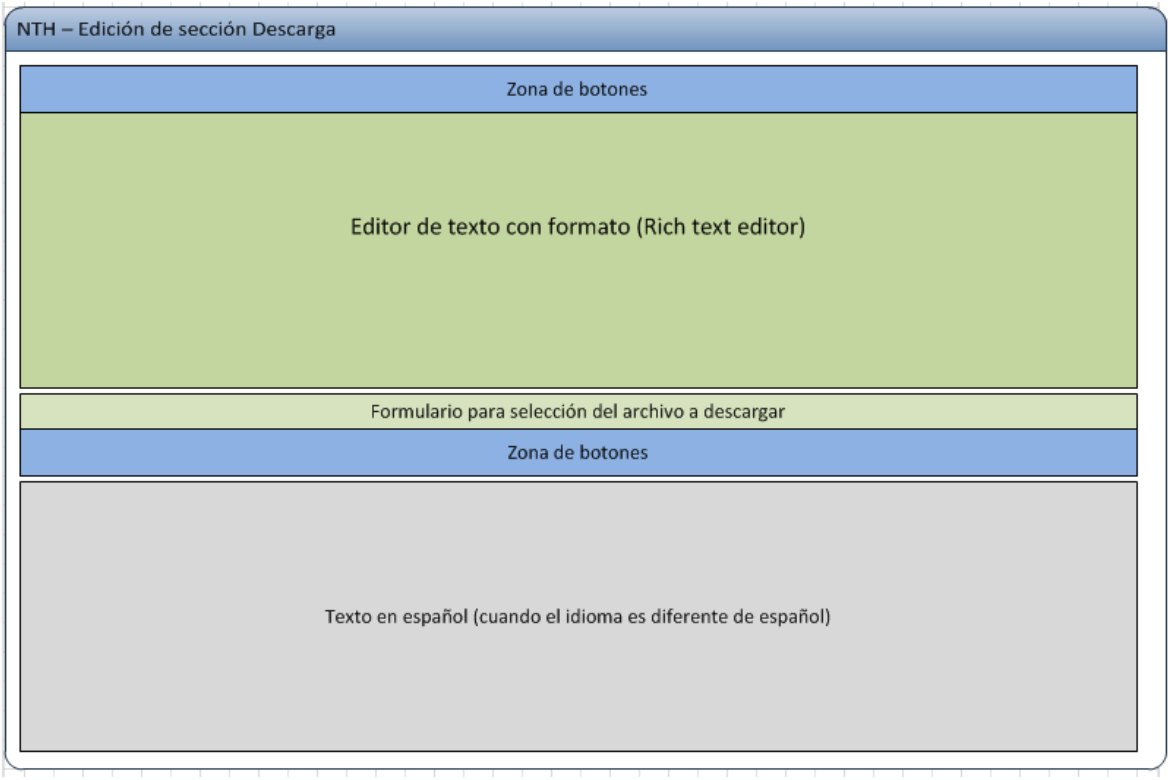


Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Botones de edición de texto	Botones que simulan la interfaz de un editor de texto, contiene entonces todas las funciones propias de uno, como negrillas, itálicas, subrayar, etc.
	Guardar	Guarda el texto en la base de datos junto con la página seleccionada para el link actualiza la sección y refresca la página para mostrar el texto en la sección correspondiente. Cierra la ventana de edición.

4.4.2.1.13 F7.3 – GUI – 03 – Administrar multimedia – texto – Descarga

La edición de link incluye el editor de texto con formato, el botón de Guardar, el texto en español en el caso en que se esté editando el contenido en otro idioma, además de un selector de archivos, el cual redireccionará se podrá descargar.

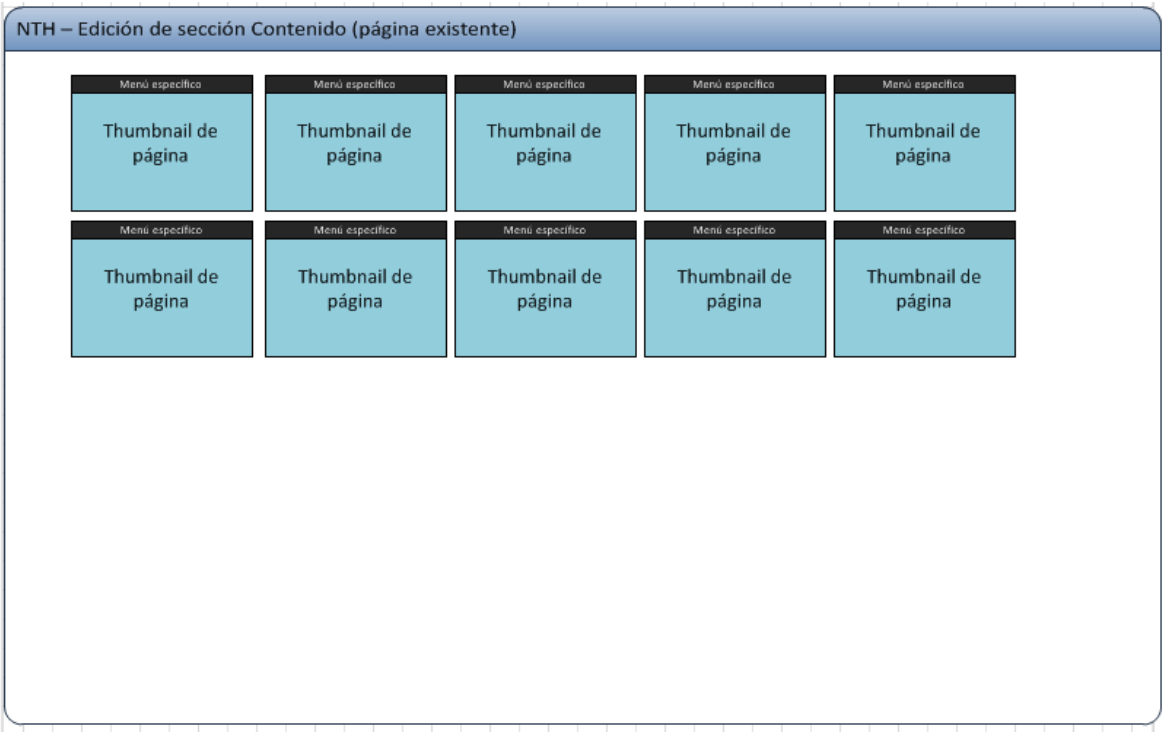


Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Botones de edición de texto	Botones que simulan la interfaz de un editor de texto, contiene entonces todas las funciones propias de uno, como negrillas, itálicas, subrayar, etc.
	Guardar	Guarda el texto en la base de datos y el archivo en la carpeta dedicada a esto, actualiza la sección y refresca la página para mostrar el texto en la sección correspondiente. Cierra la ventana de edición.

4.4.2.1.14 F7.3 – GUI – 04 – Administrar multimedia – texto – Contenido

La edición contenido difiere de las otras pantallas pues no se muestra el editor de texto. En esta pantalla se muestran miniaturas de páginas ya existentes para insertarlas en una sección.

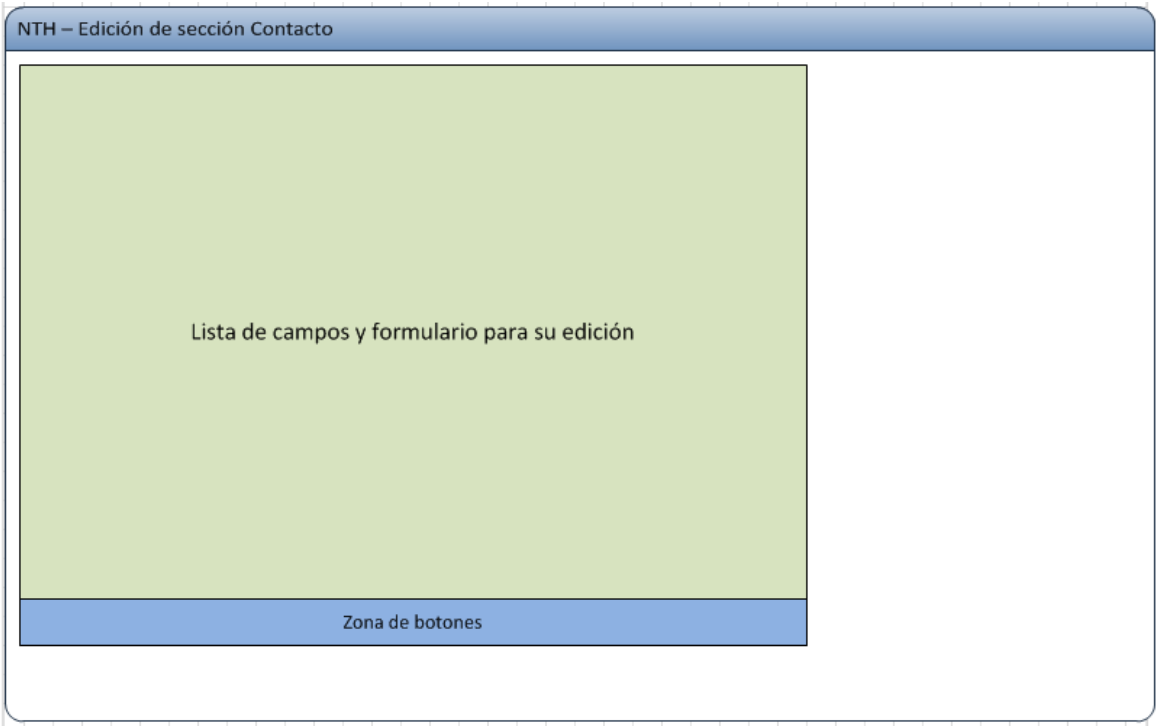


Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Menú específico	Botones que dan funcionalidad específica a las imágenes de cada página: • **Seleccionar:** Selecciona la página para mostrar en la sección, guarda en la base de datos y refresca la página para mostrar el contenido en la sección correspondiente. Cierra la ventana de edición.

4.4.2.1.15 F7.3 – GUI – 05 – Administrar multimedia – texto – Contacto

La edición de sección de contacto permite ingresar un formulario que enviará un e-mail con los datos ingresados por el usuario de la página web.

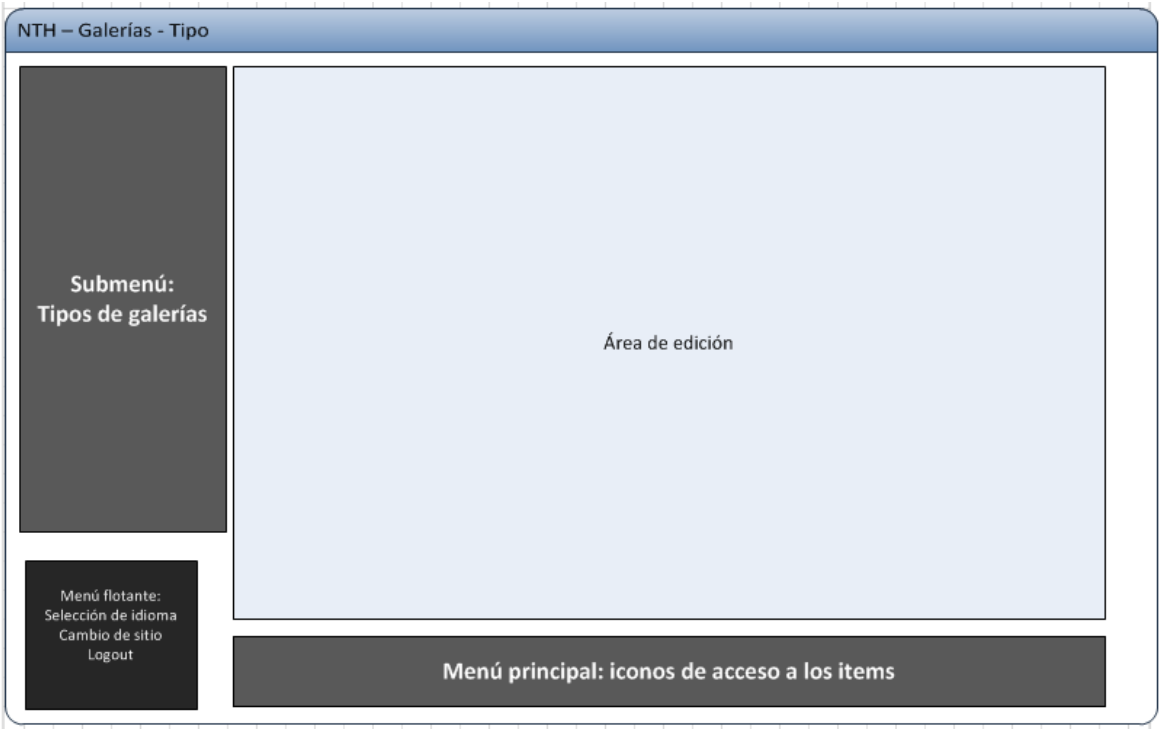


Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Guardar	Guarda los textos de los campos en la base de datos, las asocia a la sección y refresca la página para mostrar el formulario en la sección correspondiente. Cierra la ventana de edición.

4.4.2.1.16 F8 – GUI – 01 – Administrar galerías – Selección de tipo

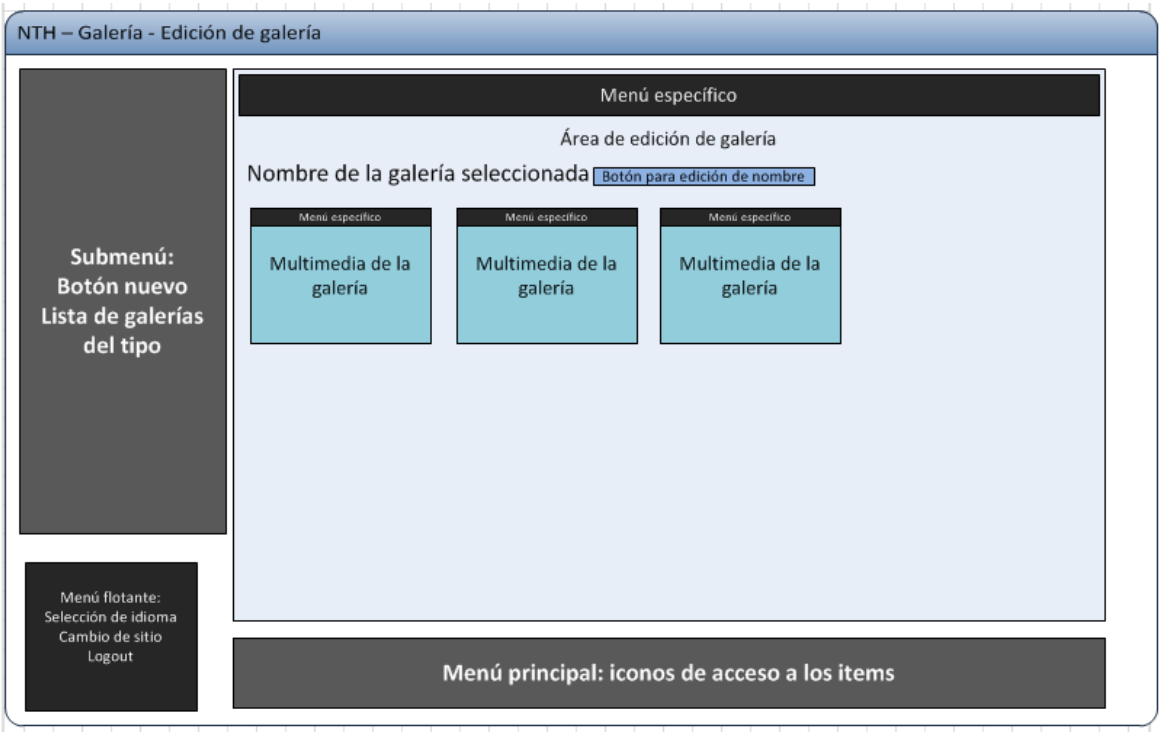
Muestra la pantalla de edición de galerías. En primer lugar se muestran los tipos de galerías que se pueden crear o editar (Texto y Fotos). Una vez seleccionado el tipo, las galerías existentes se cargan en el submenú en forma de botones. En el área de edición se muestra inicialmente el formulario de propiedades donde se puede el modelo la galería. En segundo lugar se muestra el multimedia que va en la galería, puede ser contenido o fotos. En tercer lugar se muestra una previsualización de la galería y un área de edición de las propiedades. Estas propiedades son específicas para cada tipo y modelo de galería.



Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Salir	Sale del sistema
	Cambiar	Muestra la pantalla de selección de sitio
	Menú principal	Botones para acceder a las funcionalidades del sistema: Layouts, Páginas, Galerías, Idiomas, Menú, Cuenta y Usuarios
	Nuevo	Muestra la pantalla de creación/edición de galería
	Botones de lista	Un botón por cada tipo de galería (Foto o contenido). Muestra el submenú del tipo correspondiente.

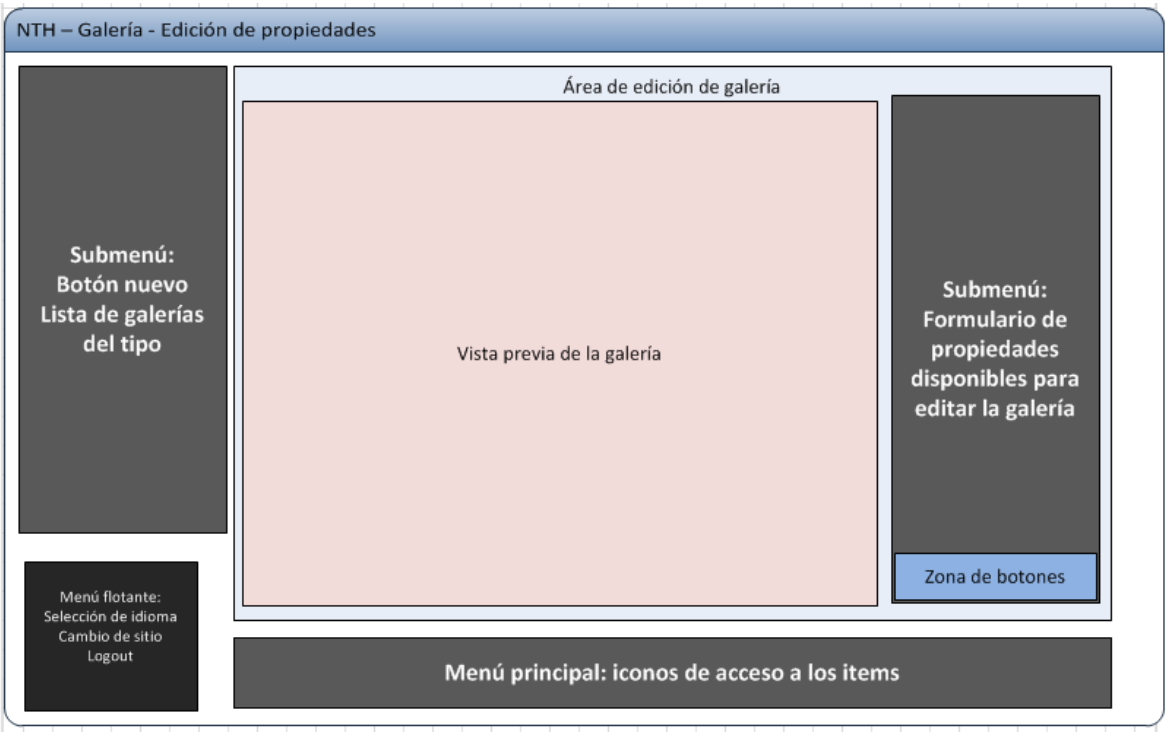
4.4.2.1.17 F8 – GUI – 02 – Administrar galerías – Edición de multimedia



Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Salir	Sale del sistema
	Cambiar	Muestra la pantalla de selección de sitio
	Menú principal	Botones para acceder a las funcionalidades del sistema: Layouts, Páginas, Galerías, Idiomas, Menú, Cuenta y Usuarios
	Nuevo	Muestra la pantalla de creación/edición de galería
	Botones de lista	Un botón por cada registro de la base de datos. Muestra la pantalla de edición de galería
	Menú específico de la galería	Botones para la funcionalidad específica: • **Agregar multimedia:** permite agregar multimedia a la galería, según su tipo • **Continuar:** Muestra la pantalla de edición de propiedades de la galería • **Eliminar:** Elimina la galería
	Menú específico de multimedia	Botones para la funcionalidad específica: • **Editar:** Abre la pantalla de edición de multimedia específico • **Eliminar:** Elimina el ítem de multimedia de la galería

4.4.2.1.18 F8 – GUI – 03 – Administrar galerías – Modificar propiedades



Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Salir	Sale del sistema
	Cambiar	Muestra la pantalla de selección de sitio
	Menú principal	Botones para acceder a las funcionalidades del sistema: Layouts, Páginas, Galerías, Idiomas, Menú, Cuenta y Usuarios
	Botones de lista	Un botón por cada registro de la base de datos. Muestra la pantalla de edición de galería
	Recargar	Guarda las propiedades modificadas y refresca la vista previa de la galería

4.4.2.1.19 F8 – GUI – 04 – Seleccionar Galería

Esta pantalla permite seleccionar una galería creada previamente para insertarla en una sección de la página.

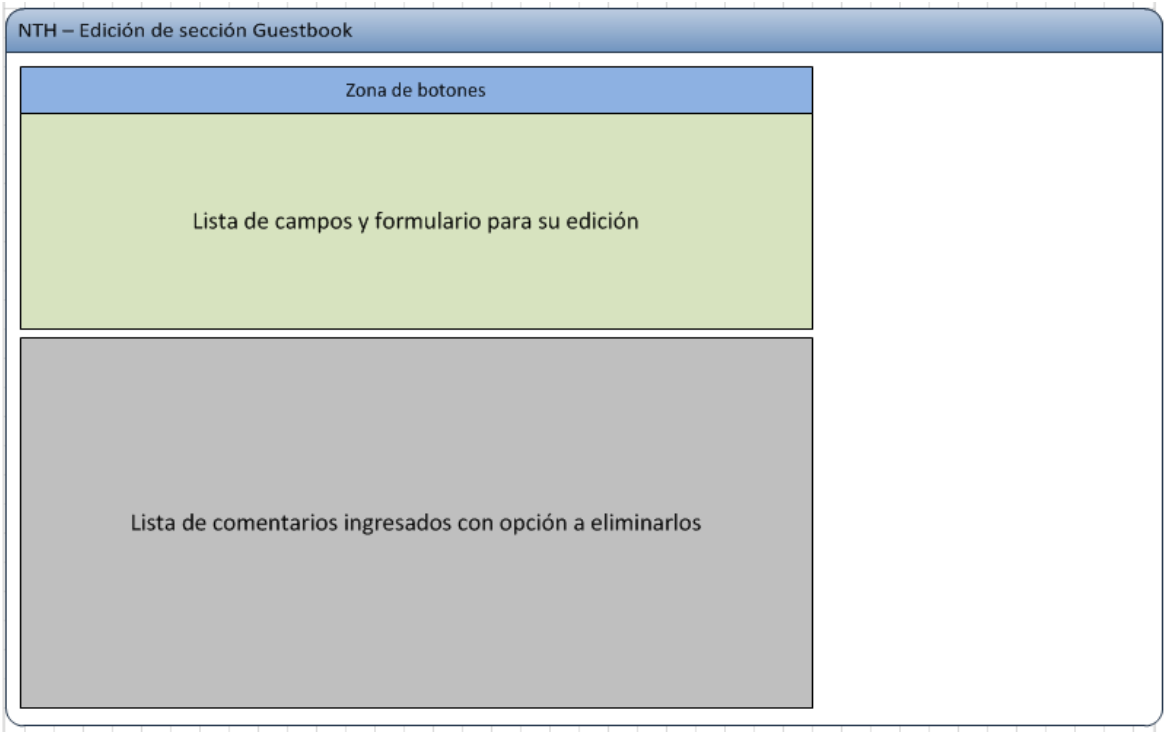


Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Botones de selección de tipo de galería	Un botón por cada tipo de galería. Muestra una lista de las galerías disponibles para selección
	Guardar	Guarda la selección de galería en la sección. Refresca la página para mostrar la galería en la sección correspondiente. Cierra la ventana de edición.

4.4.2.1.20 F9-F10 – GUI – 01 – Administrar guestbook y comentarios

La administración de guestbook se abre en una ventana aparte a partir de la sección correspondiente Aquí se puede editar las etiquetas que ve el usuario de la página web cuando va a crear un nuevo comentario. También se .pueden eliminar comentarios.



Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Guardar	Guarda los textos de los campos en el guestbook
	Eliminar	Elimina el comentario correspondiente del guestbook

4.4.2.1.21 F10 – GUI – 02 – Nuevo comentario

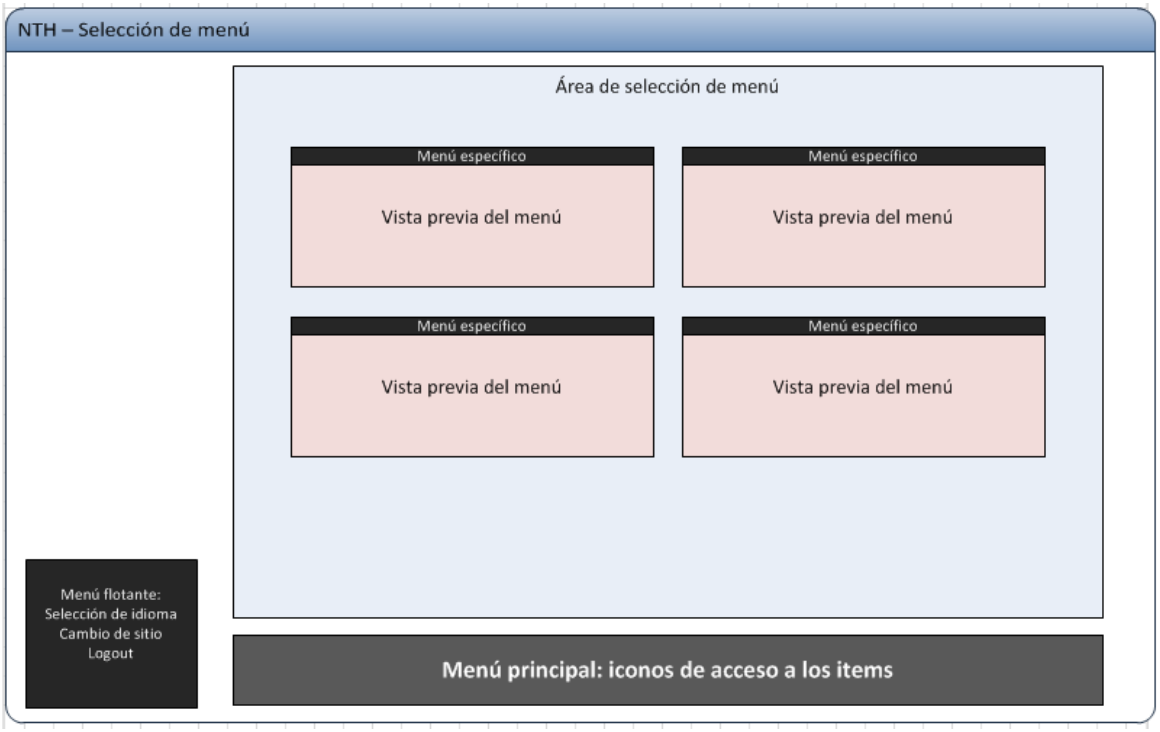
Muestra la pantalla de creación de comentario. Esta pantalla es vista por el visitante de la página web y le permite dejar un comentario, una queja, una sugerencia, etc. en un guestbook o libro de visitas previamente configurado en el sitio.

Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Guardar	Guarda el comentario y los demás datos ingresados
	Cancelar	Cierra el diálogo de creación de comentario sin crear nada.

4.4.2.1.22 F11 – GUI – 01 – Seleccionar menú

Muestra la pantalla de selección de menú. Permite cambiar la manera en que se muestra en menú. Esta selección es por sitio, todas las páginas de un mismo sitio muestran el menú de la misma manera.

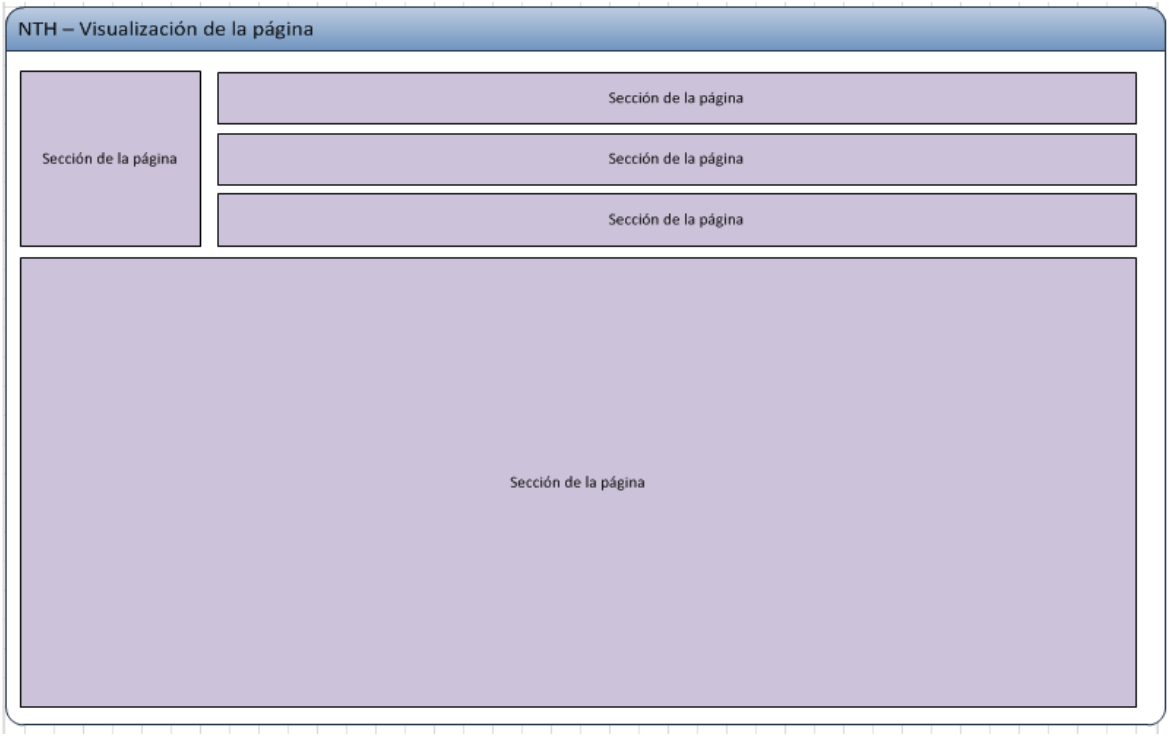


Especificaciones:

Color	Blanco	
Fuente	Arial 12pt	
Color fuente	Negro	
Botones	**Botón**	**Acción**
	Salir	Sale del sistema
	Cambiar	Muestra la pantalla de selección de sitio
	Menú principal	Botones para acceder a las funcionalidades del sistema: Layouts, Páginas, Galerías, Idiomas, Menú, Cuenta y Usuarios
	Seleccionar	Guarda el menú en el sitio para mostrarlo cada vez que se abre una página con sección menú

4.4.2.1.23 F12 – GUI – 01 – Visualizar página web

Esta es la pantalla que visualiza el usuario de la página web, es decir, quien entra a ver, sin opción a editar nada.



Especificaciones:

Color	Blanco
Fuente	Arial 12pt
Color fuente	Negro
Botones	Los botones aquí dependen del diseño realizado

4.4.2.2. Diseño de clases controlador

Caso de uso			Controlador	Descripción
F0	F0	Ingesar al sistema	LoginController	Se encarga de controlar el acceso de los usuarios al sistema. Acciones: • login: valida las credenciales ingresadas por el usuario • logout: limpia las variables de sesión del servidor y redirecciona a la pantalla de login
F1	F1.1	Crear usuario	UsuarioController	Se encarga de controlar las acciones para la administración de usuarios. Acciones: • list: lista de usuarios • show: muestra los datos de un usuario • form: formulario para el ingreso de datos • save: acción para guardar los datos • delete: acción para borrar un registro
	F1.2	Ver usuario		
	F1.3	Editar usuario		
	F1.4	Eliminar usuario		
F2	F2.1	Crear idioma	IdiomaController	Se encarga de controlar las acciones para la

	F2.2	Ver idioma		administración de idiomas. Acciones: • list: lista de idiomas • show: muestra los datos de un idioma • form: formulario para el ingreso de datos • save: acción para guardar los datos • delete: acción para borrar un registro
	F2.3	Editar idioma		
	F2.4	Eliminar idioma		
F3	F3.1	Crear sitio	SitioController	Se encarga de controlar las acciones para la administración de sitios. Acciones: • list: lista de sitios • show: muestra los datos de un sitio • form: formulario para el ingreso de datos • save: acción para guardar los datos • delete: acción para borrar un registro
	F3.2	Ver sitio		
	F3.3	Editar sitio		
F4	F4.1	Crear layout	LayoutController	Se encarga de controlar las acciones para la administración de layouts. Acciones: • list: lista de layouts • show: muestra los datos de un layout • form: formulario para el ingreso de datos • save: acción para guardar los datos • delete: acción para borrar un registro
	F4.2	Ver layout		
	F4.3	Editar layout		
	F4.4	Eliminar layout		
F5	F5.1	Crear página	PaginaController	Se encarga de controlar las acciones para la administración de páginas. Acciones: • list: lista de página • show: muestra los datos de un página • form: formulario para el ingreso de datos • save: acción para guardar los datos • delete: acción para borrar un registro
	F5.2	Ver página		
	F5.3	Editar página	EditorController	Se encarga de controlar las acciones para la edición de los elementos contenidos dentro de una página. Acciones: • nuevaSeccion • deleteSeccion • editarSeccion • setSitio • cambiarIdioma • propiedades • guardarPagina • subMenu • editarBanner
	F5.4	Eliminar página		
F6	F6.1	Crear sección	SeccionController	Se encarga de controlar las acciones para la administración de secciones.
	F6.2	Ver sección		

	F6.3	Editar sección		Acciones: • list: lista de secciones • show: muestra los datos de un sección • form: formulario para el ingreso de datos • save: acción para guardar los datos • delete: acción para borrar un registro
	F6.4	Eliminar sección		
F7	F7.1	Administrar fotos	FotoController	Se encarga de controlar las acciones para la administración de fotos. Acciones: • list: lista de foto • show: muestra los datos de un foto • form: formulario para el ingreso de datos • save: acción para guardar los datos • delete: acción para borrar un registro • upload: sube una fotografía al servidor • render: muestra una fotografía de la base de datos
	F7.2	Administrar videos	VideoController	Se encarga de controlar las acciones para la administración de videos. Acciones: • verVideos: lista de videos • verVideos: muestra los datos de un video • guardarVideo: acción para guardar los datos • login: acción para la autentificación con el sistema de Youtube • selectVideo: acción para seleccionar un video del servicio de youtube
	F7.3	Administrar texto	FraseController	Se encarga de controlar las acciones para la administración de textos. Acciones: • list: lista de textos • show: muestra los datos de un texto • form: formulario para el ingreso de datos • save: acción para guardar los datos • delete: acción para borrar un registro
F8	F8.1	Crear galería	GaleriaController	Se encarga de controlar las acciones para la administración de galerías. Acciones: • list: lista de galerías • show: muestra los datos de un galería • form: formulario para el ingreso de datos • save: acción para guardar los datos • delete: acción para borrar un registro
	F8.2	Ver galería		
	F8.3	Editar galería	EditorGaleriaController	Se encarga de las acciones de edición de las propiedades de una galería. Acciones: • menuFotos • deleteFoto
	F8.4	Eliminar galería		

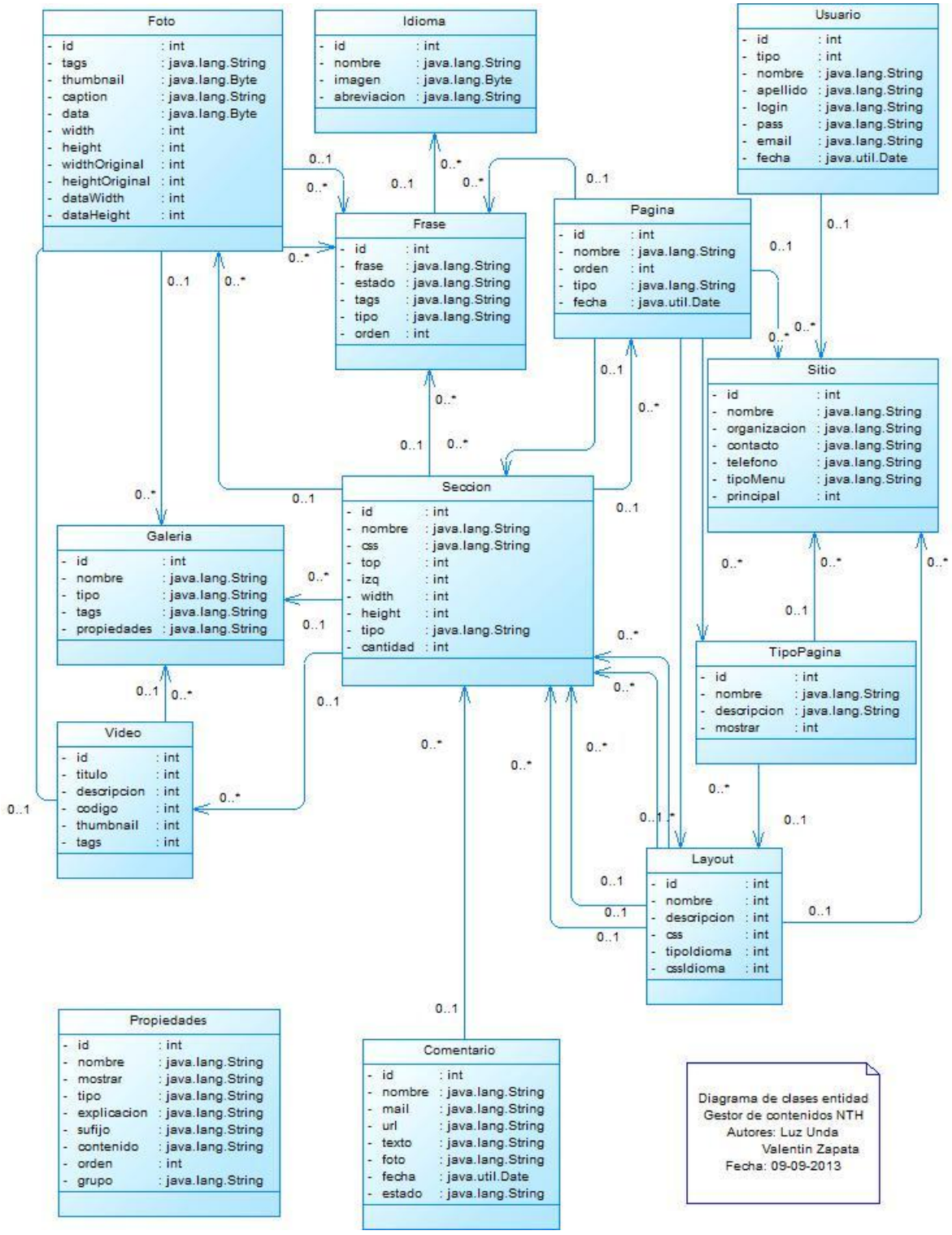
				• guardarGaleria • customizeGal • customizeSquares • customizeJCycle • customizeCloudCarousel • renderGal • renameGal • savePagToGal
F9	F9.1	Crear publicación	PublicacionController	Se encarga de controlar las acciones para la administración de publicaciones. Acciones: • list: lista de publicaciones • show: muestra los datos de un publicación • form: formulario para el ingreso de datos • save: acción para guardar los datos • delete: acción para borrar un registro
	F9.2	Ver publicación		
	F9.3	Editar publicación		
	F9.4	Eliminar publicación		
F10	F10.1	Crear guestkook	GuestbookController	Se encarga de controlar las acciones para la administración de guestbooks. Acciones: • list: lista de guestbooks • show: muestra los datos de un guestbook • form: formulario para el ingreso de datos • save: acción para guardar los datos • delete: acción para borrar un registro
	F10.2	Ver guestkook		
	F10.3	Editar guestkook		
	F10.4	Eliminar guestkook		
F11	F11.1	Crear comentario	PaginaWebController	Se encarga de controlar las interacciones de los visitantes con las páginas web registradas en el gestor de contenidos. Acciones: • guardarComentario: guardar un comentario • verComentario: muestra los datos de un comentario • borraComentario: eliminar un comentario
	F11.2	Ver comentario		
	F11.3	Eliminar comentario		
F12	F12	Seleccionar menú	MenuController	Se encargar de presentar los estilos de menú disponibles para el sitio. Acciones: • mostrarTodos: muestra los estilos de menú disponibles • mostrarMenu: muestra el estilo de menú seleccionado • guardarTipo: guardar el estilo del menú seleccionado
F13	F13	Visualizar página web	PaginaWebController	Se encarga de controlar las interacciones de los visitantes con las páginas web registradas en el gestor de contenidos.

				Acciones: • showPagina: muestra una página web registrada en el gestor de contenidos

4.4.2.3. Diseño de clases entidad

4.4.2.3.1 Diagrama de clases entidad

El diagrama de clases entidad se podrá encontrar en tamaño real en el Anexo 3: Diagrama de clases entidad.



4.4.2.3.2 Diccionario de clases entidad

Caso de uso	Nombre	Descripción
F11, F13	`Comentario`	Es el comentario que deja un visitante de la página web a través del guestbook (libro de visitas). Se representa como una

		cadena de texto.
F7.1, F13	Foto	Es una foto que puede ingresarse en una página o galería. Se representa como un arreglo de bytes para el almacenamiento de la imagen. Las propiedades como el nombre y el tamaño se almacenan como texto y enteros.
F7.3, F13	Frase	Es una cadena de texto asociada a un idioma.
F8, F13	Galeria	Es una forma de agrupar fotos o contenidos (páginas existentes) con un modelo de presentación generalmente animado y editable por el administrador del sitio.
F2, F13	Idioma	Es el idioma en el cual se muestra la página web.
F4, F13	Layout	Es una plantilla que sirve para la creación de una página.
F5, F13	Pagina	Es una página del sitio web que puede contener secciones de diferentes tipos.
F4, F5, F6, F8	Propiedades	Es un catálogo de propiedades CSS y de Javascript para modificar la apariencia de las páginas, secciones y galerías del sitio web.
F6, F13	Seccion	Es un contenedor que representa un elemento tipo div. Puede ser de diferentes tipos, como foto, video, texto, etc.
F3, F12, F13	Sitio	Es una forma de agrupar las páginas. Tiene un nombre, que es el que se escribe en la barra de direcciones para acceder a las páginas web.
F5	TipoPagina	Permite clasificar las páginas de acuerdo a su utilidad dentro del gestor de contenidos.
F0, F1	Usuario	Permite ingresar al sistema como administrador para efectuar modificaciones.

F7.2, F13	`Video`	Es una agrupación de propiedades que permiten recuperar un video del servicio de YouTube.

4.4.3. Diseño de pruebas

4.4.3.1. Diseño de pruebas de desarrollo

Para cada parte del sistema desarrollada se completará un formulario como el siguiente:

Nombre: _____ **Fecha:** _____

Módulo: _____ **Caso de uso:** _____

Prueba: _____

_____

Precondición: _____

_____

Resultado esperado: _____

_____

Prueba 1
Resultado obtenido: _____ _____ **¿Pasa la prueba?** SI NO **Errores encontrados:** _____ _____ _____ _____ **Solución:** _____ _____ **Comentarios, sugerencias, mejoras:** _____ _____ _____ _____

Prueba 2

Resultado obtenido: _____
_____

¿Pasa la prueba? SI NO

Errores encontrados: _____
_____
_____
_____

Solución: _____
_____

Comentarios, sugerencias, mejoras: _____
_____
_____
_____

Prueba 3

Resultado obtenido: _____
_____

¿Pasa la prueba? SI NO

Errores encontrados: _____
_____
_____
_____

Solución: _____
_____

Comentarios, sugerencias, mejoras: _____
_____
_____
_____

4.4.3.2. Diseño de prueba de usuario

Al usuario final se le entregará el siguiente formulario, el cual deberá completar y devolver a los desarrolladores para que se puedan corregir los errores encontrados, y tomar en cuenta las sugerencias.

Nombre: _____ **Fecha:** _____

1. ¿Pudo ingresar al sistema utilizando sus credenciales?
   SI NO
   Errores o problemas: _____
   _____
   Comentarios o sugerencias: _____
   _____

1. ¿Pudo acceder correctamente a las pantallas principales del sistema utilizando los botones del menú principal?
   SI NO
   Errores o problemas: _____
   _____

   Comentarios o sugerencias: _____
   _____

2. ¿Pudo insertar exitosamente elementos nuevos al sistema?

Elemento	¿Se insertó correctamente?	Errores o problemas encontrados	Sugerencias o comentarios
Usuario	SI NO		
Sitio	SI NO		
Layout	SI NO		
Página	SI NO		
Sección	SI NO		
Foto	SI NO		
Video	SI NO		
Texto	SI NO		
Galería de fotos	SI NO		
Galería de contenidos	SI NO		

Guestbook	SI NO		
Comentario de guestbook	SI NO		

3. ¿Pudo visualizar correctamente la información de los elementos existentes?

Elemento	¿Se visualizó correctamente?	Errores o problemas encontrados	Sugerencias o comentarios
Usuario	SI NO		
Sitio	SI NO		
Layout	SI NO		
Página	SI NO		
Sección	SI NO		
Foto	SI NO		
Video	SI NO		
Texto	SI NO		
Galería de fotos	SI NO		
Galería de contenidos	SI NO		
Guestbook	SI NO		
Comentario de guestbook	SI NO		
Menú	SI NO		
Página web	SI NO		

4. ¿Pudo modificar exitosamente los campos de un elemento existente?

Elemento	¿Se modificó correctamente?	Errores o problemas encontrados	Sugerencias o comentarios
Usuario	SI NO		
Sitio	SI NO		
Layout	SI NO		

Página	SI NO		
Sección	SI NO		
Foto	SI NO		
Video	SI NO		
Texto	SI NO		
Galería de fotos	SI NO		
Galería de contenidos	SI NO		
Guestbook	SI NO		
Menú	SI NO		

5. ¿Pudo eliminar correctamente un elemento existente?

Elemento	¿Se eliminó correctamente?	Errores o problemas encontrados	Sugerencias o comentarios
Usuario	SI NO		
Sitio	SI NO		
Layout	SI NO		
Página	SI NO		
Sección	SI NO		
Foto	SI NO		
Video	SI NO		
Texto	SI NO		
Galería de fotos	SI NO		
Galería de contenidos	SI NO		
Guestbook	SI NO		
Comentario de guestbook	SI NO		
Menú	SI NO		

Página web	SI NO		

6. Califique la dificultad de utilización de cada una de las partes principales del sistema. La calificación es del 1 al 5, donde 1 es Extremadamente fácil y 5 Extremadamente difícil.

Sección del sistema	Nivel de dificultad de uso	Errores o problemas encontrados	Sugerencias o comentarios
Usuario			
Sitio			
Layout			
Página			
Sección			
Foto			
Video			
Texto			
Galería de fotos			
Galería de contenidos			
Guestbook			
Comentario de guestbook			
Menú			
Página web			

4.4.3.3. Pruebas y condiciones de aceptación

A continuación presentamos para cada caso de uso un ejemplo de prueba, con sus respectivos resultados esperados y condición de aceptación.

Caso de Prueba: F0 Ingreso al sistema
Precondición: ninguna

Caso de	Entradas	Resultados esperados	Condición de aceptación

uso			
F0	❖ El usuario ingresa sus credenciales, por defecto los valores iniciales del usuario administrador son: • Nombre de usuario: admin • Contraseña: admin	❖ Se despliega la página principal del sistema.	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones

Caso de prueba: F1 administrar usuarios
Precondición: ninguna

Caso de uso	**Entradas**	**Resultados esperados**	**Condición de aceptación**
F1	❖ Clic en el botón "usuarios" en el menú principal	❖ Muestra la página de usuarios	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F1.1	❖ Ingresa datos de prueba: • Nombre: Juan • Apellido: Rosales • Login: jRosales • Password: 123456 • Email:juan@rosales.com	❖ El sistema guarda el nuevo usuario en la base de datos y muestra la página de lista de usuarios, en la cual se encuentra el usuario ingresado.	Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información
F1.2	❖ El usuario da clic sobre el nombre del usuario que desee ver.	❖ Muestra una página donde se muestran todos los datos asociados con el usuario seleccionado	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F1.3	❖ El usuario da clic en el botón "editar". ❖ Modifica los datos del usuario con la siguiente información: • Login: jRosalesUno • Password: 1234567 • Email:juanR@rosales.com	❖ El sistema guarda los datos modificados en la base de datos y muestra la página de lista de usuarios, en la cual se encuentra el usuario modificado con los nuevos datos.	Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información
F1.4	❖ El usuario da clic sobre el nombre del usuario que desee ver. ❖ Clic en el botón "eliminar"	❖ Modifica el estado del usuario	Se despliega la pantalla correcta y permite hacer

	❖ Clic en el botón “aceptar”	seleccionado y muestra la lista de usuarios, en la cual no se encuentra el usuario eliminado.	clic en vínculos y botones

Caso de prueba: F3 administrar sitios
Precondición: un usuario registrado en el sistema

Caso de uso	Entradas	Resultados esperados	Condición de aceptación
F0	❖ El usuario ingresa en el sistema	❖ Muestra la pantalla de selección y creación de sitios	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F3.1	❖ Ingresa datos del sitio en el formulario: • Nombre: prueba • Organización: Acción social • Contacto: Juan Rosales • Teléfono: 1234567	❖ El sistema guarda el nuevo sitio en la base de datos y muestra la página principal del sistema.	Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información
F3.2	❖ El usuario da clic en el botón “administración” en el menú principal. ❖ Clic en el botón “sitio” ❖ Clic en el nombre del sitio que se desea visualizar	❖ Muestra una página donde se muestran todos los datos asociados con el sitio seleccionado	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F3.3	❖ El usuario da clic en el botón “editar”. ❖ Modifica los datos del sitio con la siguiente información: • Nombre: prueba2 • Teléfono: 7654321	❖ El sistema guarda los datos modificados en la base de datos y muestra la página de lista de sitios, en la cual se encuentra el sitio modificado con los nuevos datos.	Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información

Caso de prueba: F4.Administrar layouts
Precondición: un usuario y un sitio registrados en el sistema

Caso de uso	Entradas	Resultados esperados	Condición de aceptación
F4	❖ Clic en el botón “layouts” en el menú principal	❖ Se despliega la pantalla de administración de layouts	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones

F4.1	❖ Clic en el botón “Crear nuevo” ❖ Arrastra y/o redimensiona las diferentes áreas disponibles. ❖ Clic en el botón “Guardar diseño”. ❖ Ingresa los datos en el formulario: • Nombre del diseño: prueba • Descripción: prueba ❖ Clic en el botón “Aceptar”	❖ El sistema guarda el nuevo layout en la base de datos y se agrega un botón en el área de administración	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones. Además permite mover las secciones arrastrándolas y redimensionarlas. Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información
F4.2	❖ Clic en el botón con el nombre del layout (“prueba”)	❖ El sistema muestra el layout	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F4.3	❖ Clic en el botón con el nombre del layout (“prueba”) ❖ Arrastra y/o redimensiona las diferentes áreas disponibles. Clic en el botón “Guardar diseño” ❖ En caso de ser necesario, modifica los datos del formulario ❖ Clic en el botón “Aceptar”	❖ El sistema guarda los cambios en la base de datos y muestra la pantalla de edición.	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones. Permite mover las secciones arrastrándolas y redimensionarlas. Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información
F4.4	❖ Clic en el botón con el nombre del layout (“prueba”) ❖ Clic en el botón “Borrar diseño” ❖ Clic en el botón “OK”	❖ El sistema elimina el layout de la base de datos y elimina el botón del área de administración	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones. Al hacer clic en Eliminar se elimina del sistema el elemento deseado.

Caso de prueba: F5 administrar páginas
Precondición: un usuario, sitio y layout registrado en el sistema.

Caso de uso	**Entradas**	**Resultados esperados**	**Condición de aceptación**
F5	❖ Clic en el botón “páginas”	Muestra la pantalla de administración de páginas	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F5.1	❖ Clic en el botón “Crear	❖ El sistema guarda la	Muestra el formulario

	nueva” • Selecciona el layout: prueba ❖ Clic en el botón crear página Ingresa datos de la página en el formulario: • Nombre: inicio • Menú: Home • Orden: 1	nueva página en la base de datos, agrega el botón en el menú de páginas, muestra la pantalla de edición de contenidos.	y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información
F5.2	❖ El usuario da clic en el botón con el nombre de la página que desea visualizar en el menú de páginas.	❖ Muestra una página donde se muestran todos los datos asociados con la página seleccionada.	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F5.3	❖ El usuario da clic en el botón “Menú”. ❖ Modifica los datos la página con la siguiente información: • Menú: Inicio • Orden: 2	❖ El sistema guarda los datos modificados en la base de datos y muestra la pantalla de edición de contenido con los datos modificados.	Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información
F5.4	❖ Clic en el botón “Borrar” ❖ Clic en el botón “Aceptar”	❖ Borrar la página de la base de datos, muestra la pantalla de administración de páginas.	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones. Al hacer clic en Eliminar se elimina del sistema el elemento deseado.

Caso de prueba: F6 administrar secciones
Precondición: un usuario, sitio, layout y página registrados en el sistema.

Caso de uso	**Entradas**	**Resultados esperados**	**Condición de aceptación**
F6.1	❖ Clic en el botón “Insertar sección” • Selecciona el tipo “Texto”	❖ El sistema guarda la nueva sección en la base de datos, agrega el recuadro que representa gráficamente a la sección dentro de la página.	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F6.2	❖ Clic en el botón “propiedades” ubicado en la barra de administración de la sección.	❖ Muestra una ventana con las propiedades de la sección.	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones

F6.3	❖ Edita las propiedades de la sección con los siguientes datos: • Ancho: 300 • Alto: 200 ❖ Clic en el botón "Aplicar". ❖ Clic en el botón "Guardar"	❖ El sistema guarda los datos modificados en la base de datos y aplica los cambios efectuados en el recuadro que representa a la sección.	Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información
F6.4	❖ Clic en el botón "eliminar" ubicado en la barra de administración de la sección.	❖ Borrar la sección de la base de datos, borra el recuadro que representa a la sección.	Al hacer clic en Eliminar se elimina del sistema el elemento deseado.

Caso de prueba: F7.1 Administrar fotos
Precondición: un usuario, sitio, layout, página y sección registrados en el sistema.

Caso de uso	**Entradas**	**Resultados esperados**	**Condición de aceptación**
F7.1.1	❖ Clic en el botón "Editar"	❖ El sistema abre una ventana en el navegador con formulario de ingreso de datos.	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F7.1.1 F7.1.2	❖ Ingresa los siguientes datos de prueba: • Nombre: foto1 • Ancho: 300 • Alto: 300 • Imagen: selecciona una imagen de prueba. Clic en el botón "guardar"	❖ Guardar la foto y sus datos, recarga el recuadro de la sección con la fotografía seleccionada.	Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información
F7.1.3	❖ Clic en el botón "Editar" ❖ Edita las propiedades de la foto con los siguientes datos: • Ancho: 200 • Alto: 200 ❖ Clic en el botón "Guardar"	❖ El sistema redimensiona la fotografía, guarda los datos modificados en la base de datos y aplica los cambios efectuados en el recuadro que representa a la sección.	Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información. Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F7.1.4	❖ Clic en el botón "eliminar" ubicado en la barra de administración de la	❖ Borrar la sección de la base de datos y todos sus contenidos.	Al hacer clic en Eliminar se elimina del sistema el elemento deseado.

	sección.		

Caso de prueba: F7.2 Administrar videos
Precondición: un usuario, sitio, layout, página y sección registrados en el sistema.

Caso de uso	Entradas	Resultados esperados	Condición de aceptación
F7.2.1	❖ Clic en el botón "Editar"	❖ El sistema abre una ventana en el navegador con formulario de ingreso de datos.	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F7.2.1 F7.2.2	❖ Clic en el link "Clic aquí para continuar" ❖ Clic en el botón "Grant acces" ❖ Clic en "subir video" ❖ Ingresa los siguientes datos de prueba: • Título: video de prueba • Categoría: Educación • Descripción: video de prueba ❖ Clic en continuar • Selecciona el archivo del video	❖ Sube el video seleccionado a YouTube, guarda el URL en la base de datos y actualiza la sección para reflejar los cambios.	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones. Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información
F7.2.4	❖ Clic en el botón "eliminar" ubicado en la barra de administración de la sección.	❖ Borrar la sección de la base de datos y todos sus contenidos.	Al hacer clic en Eliminar se elimina del sistema el elemento deseado.

Caso de prueba: F7.3 Administrar texto
Precondición: un usuario, sitio, layout, página y sección registrados en el sistema.

Caso de uso	Entradas	Resultados esperados	Condición de aceptación
F7.3.1	❖ Clic en el botón "Editar"	❖ El sistema abre una ventana en el navegador con formulario de ingreso de datos.	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F7.3.1 F7.3.2	❖ Ingresa el siguientes texto de prueba: • Texto: Texto de prueba con formato • Edita el texto con cualquiera de las opciones del	❖ Guarda el texto en la base de datos y actualiza la sección para reflejar los cambios.	Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información

	editor de texto ❖ Clic en el botón "guardar"		
F7.3.3	❖ Clic en el botón "Editar" Edita el texto: • Texto: texto editado ❖ Clic en el botón "Guardar"	❖ El sistema guarda los datos modificados en la base de datos y aplica los cambios efectuados en el recuadro que representa a la sección.	Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información
F7.3.4	❖ Clic en el botón "eliminar" ubicado en la barra de administración de la sección.	❖ Borrar la sección de la base de datos y todos sus contenidos.	Al hacer clic en Eliminar se elimina del sistema el elemento deseado.

Caso de prueba: F8.Administrar galerías de fotos
Precondición: un usuario y un sitio registrados en el sistema

Caso de uso	**Entradas**	**Resultados esperados**	**Condición de aceptación**
F8	❖ Clic en el botón "Galería" en el menú principal	❖ Se despliega la pantalla de administración de Galerías	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F8.1	❖ Clic en el botón "Galería" en el menú principal ❖ Clic en el botón del tipo de galería, en este caso, "Fotos" ❖ Clic en el botón "Crear nueva" ❖ Clic en el diseño deseado ❖ Llena el nombre de la galería, y los tags • Nombre: Prueba • Tags: Galería prueba ❖ Clic en el botón "Guardar" ❖ Agrega fotos (Ver F7.1.1) ❖ Clic en el botón "Continuar" ❖ Modifica las propiedades necesarias para que la galería se	❖ El sistema crea una galería con el nombre y los tags ingresados. ❖ El sistema inserta las fotos y las asigna a la galería ❖ El sistema modifica las propiedades de la galería: las guarda en la base de datos y las muestra al usuario.	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones. Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información

	vea como deseado. ❖ Clic en el botón "Recargar" para guardar y ver los cambios realizados en las propiedades		
F8.2	❖ Clic en el botón del tipo de galería, en este caso, "Fotos" ❖ Clic en el botón con el nombre de la galería, en este caso, "Prueba" ❖ Clic en el botón "Continuar"	❖ El sistema muestra una pre visualización de la galería, con las fotos ingresadas y las propiedades establecidas	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F8.3	❖ Clic en el botón del tipo de galería, en este caso, "Fotos" ❖ Clic en el botón con el nombre de la galería, en este caso, "Prueba" ❖ Clic en el recuadro con el nombre de la galería para cambiarlo ❖ Agrega, edita o elimina fotos (Ver F7.1.1, F7.1.3, F7.1.4 respectivamente) ❖ Clic en el botón "Continuar" ❖ Modifica las propiedades necesarias para que la galería se vea como deseado. ❖ Clic en el botón "Recargar" para guardar y ver los cambios realizados en las propiedades	❖ El sistema modifica el nombre de la galería ❖ El sistema inserta las fotos y las asigna a la galería, las modifica o las elimina ❖ El sistema modifica las propiedades de la galería: las guarda en la base de datos y las muestra al usuario.	Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información
F8.4	❖ Clic en el botón "Galería" en el menú principal ❖ Clic en el botón del tipo de galería, en este caso, "Fotos" ❖ Clic en el botón con el nombre de la galería, en este caso, "Prueba" ❖ Clic en el botón "Eliminar"	❖ El sistema elimina la galería y todas sus fotos de la base de datos	Al hacer clic en Eliminar se elimina del sistema el elemento deseado.

	❖ Clic en el botón “OK”		

Caso de prueba: F8.Administrar galerías de contenidos
Precondición: un usuario, un sitio, un layout y páginas creadas con el layout registrados en el sistema

Caso de uso	Entradas	Resultados esperados	Condición de aceptación
F8	❖ Clic en el botón “Galería” en el menú principal	❖ Se despliega la pantalla de administración de Galerías	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F8.1	❖ Clic en el botón “Galería” en el menú principal ❖ Clic en el botón del tipo de galería, en este caso, “Texto” ❖ Clic en el botón “Crear nueva” ❖ Clic en el diseño deseado ❖ Llena el nombre de la galería, los tags y selecciona el layout de las páginas que van a estar en la galería • Nombre: Prueba Texto • Tags: Galería prueba texto • Layout: Galería ❖ Clic en el botón “Guardar” ❖ Arrastra las páginas del área “Páginas disponibles” al área “Páginas seleccionadas” ❖ Clic en el botón “Continuar” ❖ Modifica las propiedades necesarias para que la galería se vea como deseado. ❖ Clic en el botón “Recargar” para guardar y ver los cambios realizados en las propiedades	❖ El sistema crea una galería con el nombre y los tags ingresados. ❖ El sistema asigna las páginas seleccionadas a la galería ❖ El sistema modifica las propiedades de la galería: las guarda en la base de datos y las muestra al usuario.	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones. Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información

F8.2	❖ Clic en el botón del tipo de galería, en este caso, "Texto" ❖ Clic en el botón con el nombre de la galería, en este caso, "Prueba Texto" ❖ Clic en el botón "Continuar"	❖ El sistema muestra una pre visualización de la galería, con las páginas seleccionadas y las propiedades establecidas	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F8.3	❖ Clic en el botón del tipo de galería, en este caso, "Texto" ❖ Clic en el botón con el nombre de la galería, en este caso, "Prueba Texto" ❖ Clic en el recuadro con el nombre de la galería para cambiarlo ❖ Agrega o elimina páginas moviéndolas entre las secciones "Páginas disponibles" y "Páginas seleccionadas" ❖ Clic en el botón "Continuar" ❖ Modifica las propiedades necesarias para que la galería se vea como deseado. ❖ Clic en el botón "Recargar" para guardar y ver los cambios realizados en las propiedades	❖ El sistema modifica el nombre de la galería ❖ El sistema asigna o des asigna las páginas a la galería ❖ El sistema modifica las propiedades de la galería: las guarda en la base de datos y las muestra al usuario.	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones. Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información
F8.4	❖ Clic en el botón "Galería" en el menú principal ❖ Clic en el botón del tipo de galería, en este caso, "Texto" ❖ Clic en el botón con el nombre de la galería, en este caso, "Prueba Texto" ❖ Clic en el botón "Eliminar" ❖ Clic en el botón "OK"	❖ El sistema elimina la galería de la base de datos y des asigna las páginas	Al hacer clic en Eliminar se elimina del sistema el elemento deseado.

Caso de prueba: F9.Administrar guestbook
Precondición: un usuario, un sitio, un layout, una página y una sección tipo "Guestbook" registrados en el sistema

Caso de uso	Entradas	Resultados esperados	Condición de aceptación
F9.1	❖ Crea una sección de tipo "Guestbook" (ver F6.1)	❖ El sistema crea un recuadro que representa la sección de guestbook ❖ El sistema crea registros en la base de datos para las etiquetas y su orden	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F9.2	❖ Abre la página que contiene el guestbook	❖ Muestra la página con todos sus contenidos, incluido el guestbook	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F9.3	❖ Clic en el botón "Editar" de la sección correspondiente ❖ Edita las etiquetas y el orden, de ser necesario ❖ Clic en el botón "Guardar"	❖ El sistema muestra la ventana de edición de Guestbook ❖ El sistema guarda la información modificada en la base de datos	Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información
F9.4	❖ Clic en el botón "Eliminar" de la sección correspondiente	❖ El sistema elimina el guestbook y sus comentarios.	Al hacer clic en Eliminar se elimina del sistema el elemento deseado.

Caso de prueba: F10.Administrar comentarios de guestbook
Precondición: un usuario, un sitio, un layout, una página y una sección tipo "Guestbook" registrados en el sistema

Caso de uso	Entradas	Resultados esperados	Condición de aceptación
F10.1	❖ Abre la página que contiene el guestbook ❖ Clic en el botón "Nuevo comentario" ❖ Completa la información: • Nombre: Prueba • Comentario: Comentario de prueba	❖ El sistema muestra la página con todos sus contenidos, incluido el guestbook ❖ Muestra un diálogo para crear un	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones. Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información

	❖ Clic en el botón "Guardar"	comentario ❖ El sistema guarda el comentario en la base de datos y lo muestra en la página	
F10.2	❖ Abre la página que contiene el guestbook	❖ El sistema muestra la página con todos sus contenidos, incluido el guestbook y sus comentarios	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones
F10.3	❖ Clic en el botón "Páginas" del menú ❖ Clic en el botón con el nombre de la página que contiene el guestbook ❖ Clic en el botón "Editar" de la sección del guestboook ❖ Busca el comentario en el área "Comentarios" ❖ Clic en el link "Eliminar" en la parte superior derecha del comentario ❖ Clic en el botón "OK"	❖ El sistema muestra el menú de Páginas ❖ El sistema muestra la pantalla de edición de la página ❖ El sistema abre la ventana de edición de guestbook ❖ El sistema elimina el comentario de la base de datos	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones. Al hacer clic en Eliminar se elimina del sistema el elemento deseado.

Caso de prueba: F11.Seleccionar menú
Precondición: un usuario y un sitio registrados en el sistema

Caso de uso	**Entradas**	**Resultados esperados**	**Condición de aceptación**
F11	❖ Clic en el botón "Menú" del menú ❖ Clic en el botón "Seleccionar" (la pequeña cruz en la esquina inferior derecha) del menú que se desea seleccionar	❖ El sistema muestra la página de selección de menú ❖ El sistema guarda la selección	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones

Caso de prueba: F12.Visualizar página web
Precondición: un usuario, un sitio y una página registrados en el sistema

Caso de	**Entradas**	**Resultados esperados**	**Condición de aceptación**

uso			
F12	❖ Abre la página que desea visualizar, ya sea escribiendo la dirección en la barra de direcciones del navegador, haciendo clic en el menú (ya sea desde la página de edición o de una página ya abierta) o haciendo clic en un link	❖ El sistema muestra la página necesaria	Se despliega la pantalla correcta y permite hacer clic en vínculos y botones

4.5. Implementación

4.5.1. Herramientas

4.5.1.1. Modelo MVC

El desarrollo del sistema manejador de contenidos fue realizado con Grails, un framework basado en la arquitectura Modelo-Vista-Controlador (MVC). A continuación se explicará en que consiste esta arquitectura.

4.5.1.1.1 Historia

La arquitectura MVC fue implementada por primera vez por Trygve Mikkjel Heyerdahl Reenskaug, informático noruego profesor de la universidad de Oslo, durante una visita al Laboratorio de Investigación de Xerox Palo Alto en 1979. Fue creado como una solución al problema de dar a los usuarios control sobre grandes y complejas cantidades de información que necesita ser visualizada desde múltiples puntos de vista.

Uno de los primeros frameworks de MVS fue la implementación de *smalltalks-80 MVC* en los años 80. Smalltalk es un lenguaje de programación orientado a objetos. El MVC fue diseñado para aplicaciones de escritorio y la idea era separar la presentación de los dominios. Los principales problemas de este framework eran la incomprensibilidad del código, la complejidad de las pantallas, y que no es aplicable a aplicaciones web.

En la actualidad la arquitectura MVC es utilizada en la mayor parte de frameworks incluyendo a Ruby On Rails, Apple Cocoa (framework para desarrollar aplicaciones de Mac OS y de iOS), ASP.Net, Grails, etc.

4.5.1.1.2 Conceptos

El objetivo principal de MVC es desarrollar aplicaciones de forma modular. El input del usuario, el modelo del objeto real y la respuesta visual del sistema están explícitamente separados, y son manejados por tres tipos diferentes de objetos, cada uno especializado en su tarea. Esto permite que la lógica de la aplicación y la interfaz gráfica estén

completamente separadas, pudiendo así implementar diferentes interfaces de usuario (ver Figura 9).

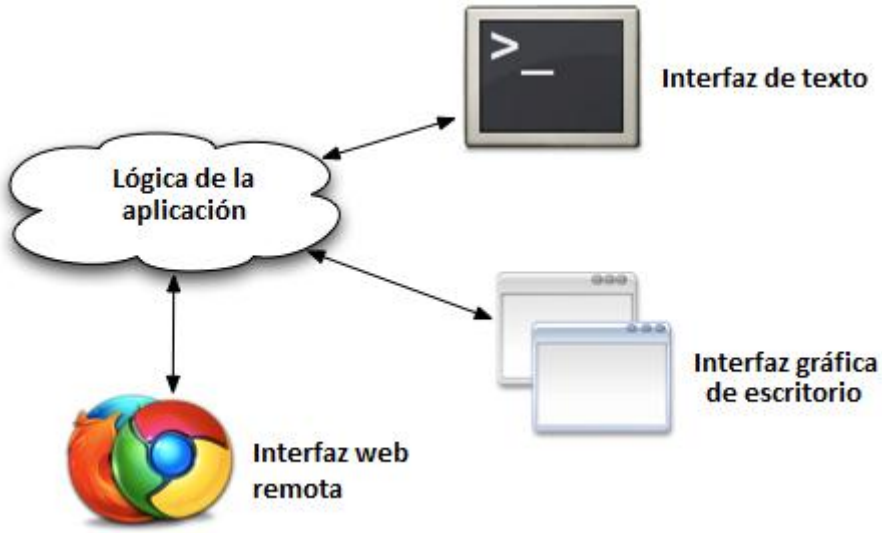


Figura 9: Separación de la lógica de la aplicación de la interfaz gráfica
Autor: Amir Salihefendic
Fuente: http://amix.dk/blog/post/19615

Modelo

El modelo es una representación en forma de datos de computación de una abstracción. Es lo que maneja el comportamiento y los datos de los dominios de la aplicación. Puede ser un solo objeto, o una estructura. Hay varias maneras de abstraer un mismo elemento, por lo que pueden coexistir varios Modelos (ver Figura 10).

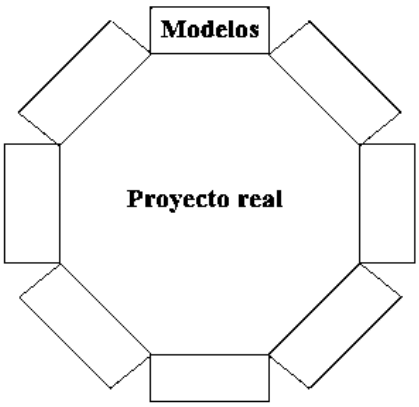


Figura 10: Varios modelos para un mismo objeto real
Autor: Trygve Reenskaug
Fuente: http://folk.uio.no/trygver/2007/MVC_Originals.pdf

Un modelo computacional consiste en un conjunto de datos y los métodos necesarios para procesarlos. Idealmente todos los modelos deberían ser completamente consistentes, pero esto no es siempre posible en la práctica, por lo que se deben aceptar ciertas inconsistencias. El objetivo es que el grupo de modelos sea razonablemente preciso en su representación del proyecto, sin perder tiempo en trivialidades. Debe existir una correspondencia entre el modelo y sus partes por un lado, y entre el modelo y lo representado por otro.

Vista

Una vista es una representación gráfica de un modelo. Actúa como un filtro, mostrando ciertos atributos y ocultando otros, según la necesidad. Para cada Modelo del sistema se tiene una o más Vistas, cada una siendo capaz de mostrar una o más representaciones del modelo en la pantalla. Una Vista también puede hacer operaciones sobre el Modelo, como actualizaciones. Puede por ejemplo haber una vista que muestra una lista de ítems, una para visualizar o editar sus atributos, o una que muestre un diagrama.

Controlador

Un Controlador es la interface entre el usuario y el sistema. Le ofrece al usuario un sistema de control. Presentas las Vistas apropiadas en el momento y lugar adecuados, muestra los controles, por ejemplo, un menú que cambia dinámicamente según el contexto. El Controlador recibe los mensajes del usuario y los traduce a mensajes apropiados que son después pasados a una o más vistas.

MVC para aplicaciones web

Los conceptos de MVC fueron desarrollados para aplicaciones de escritorio, donde existe una conexión directa entre la interfaz gráfica y el controlador responsable (ver Figura 11).

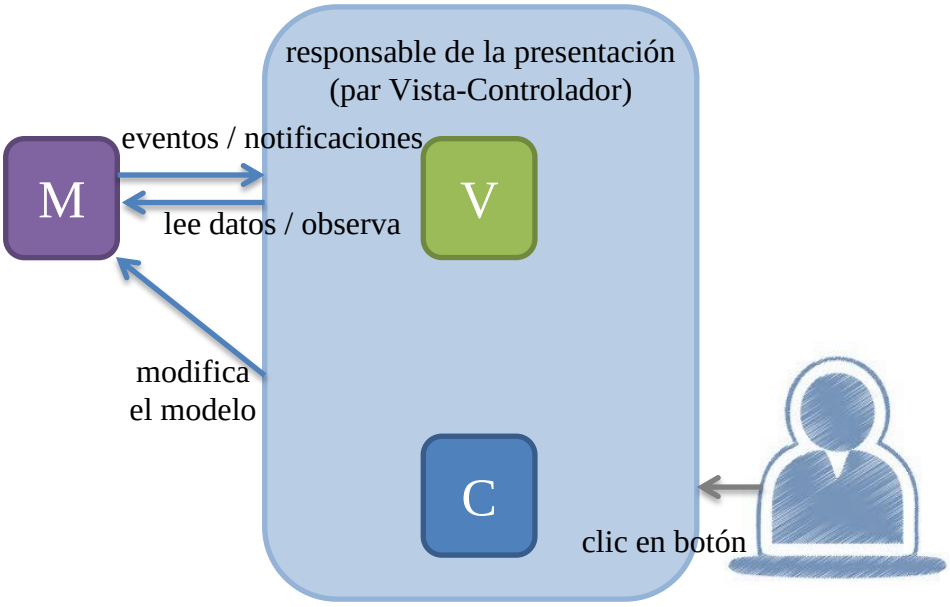


Figura 11: MVC clásico
Autor: Typo3-media
Fuente: http://www.typo3-media.com/blog/mvc-mvp-typo3-introduction.html

En aplicaciones web no se tiene esta situación, ya que las páginas se actualizan y tienen que ser enviadas desde el servidor hasta el cliente. Si el usuario hace una acción que manda una petición al servidor, esta tendrá que ser interpretada y devolver una respuesta HTML al cliente para actualizar la pantalla (ver Figura 12).

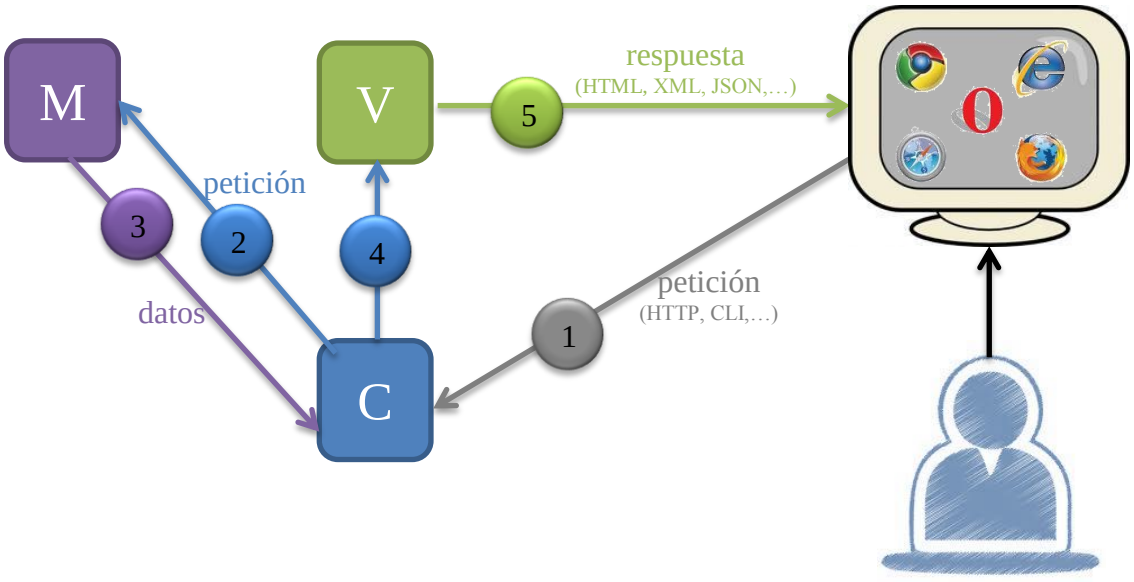


Figura 12: MVC para web
Autor: The Tech labs
Fuente: http://www.thetechlabs.com/tech-tutorials/codeigniter-tutorial-codeigniter-explained/

El HTML para una pantalla puede entenderse como sub vistas, por ejemplo, un menú o una lista. Un MVC para aplicaciones web funciona de la siguiente manera:

1. El usuario envía la petición a través del navegador al controlador
2. El controlador interactúa con el modelo, generando una petición
3. El modelo procesa la petición y devuelve información al controlador
4. El controlador analiza los datos, y, de ser necesario, envía otra petición al modelo
5. Finalmente, los datos se envían del controlador a la vista, que se muestra en el navegador

4.5.1.2. Grails

Grails es un framework de código abierto construido para desarrollar aplicaciones web. Desarrollado sobre el lenguaje de programación Groovy (el cual a su vez se basa en Java), Grails pretende ser un marco de trabajo altamente productivo siguiendo paradigmas como COC (convention over configuration, en español: convención sobre configuración) y DRY (Don't repeat yourself, en español: no te repitas), proporcionando un entorno de desarrollo estandarizado y ocultando gran parte de los detalles de configuración al programador.

Grails ha sido impulsado principalmente por la empresa G2One, la cual fue adquirida por la desarrolladora de software libre SpringSource en noviembre de 2008. En agosto de 2009 SpringSource fue a su vez adquirida por VMWare, empresa especializada en virtualización de sistemas.

El paradigma de Convención sobre Configuración significa que las aplicaciones desarrolladas en Grails, se estructuran automáticamente y por sí mismas, basándose en esquemas de denominación en lugar de tediosos archivos de configuración escritos en XML. Gracias a las funcionalidades implementadas en Grails, las clases definidas por el usuario heredan "mágicamente" los métodos y atributos necesarios para su funcionamiento, sin ninguna intervención adicional del programador, y sin abarrotar el código fuente. Todo esto simplemente usando convenciones que el desarrollador debe seguir al momento de nombrar y de ubicar a los archivos de acuerdo a su funcionalidad. Es decir, todas las clases que vayan a tener una funcionalidad de controlador dentro de la arquitectura MVC, deben ser ubicadas en la carpeta de nombre "controllers" y nombradas de tal forma que el nombre contenga la palabra "Controller" al final, por ejemplo: FacturacionController.groovy.

Para explicar cómo este paradigma es la base para lograr un framework de alta productividad, es necesario utilizar un ejemplo. Supongamos una aplicación con dos clases principales: Jugador y Equipo. Como estas dos clases son parte del dominio del problema y

por tanto representan la capa del Modelo en el patrón de diseño MVC, deben ser ubicadas bajo de la carpeta “domains” dentro del directorio de la aplicación. El código de la clase Jugador sería el siguiente:

```
class Jugador {
    String nombre
    String apellido
    Equipo equipo
    static constraints = {
        nombre(blank: false,nullable: false,size: 3..30)
        apellido(blank: false,nullable: false,size: 5..30)
        equipo(blank:true,nullable: true)
    }
}
```

En esta clase podemos notar dos secciones: la primera donde definimos los atributos de la clase con su tipo de dato y la sección de constraints o restricciones, donde especificamos las limitaciones o las características de los atributos. Estas limitaciones (constraints) servirán posteriormente para la generación del formulario de ingreso de datos y para que Hibernate (cuyo funcionamiento explicaremos posteriormente) se encargue de reflejarlos en la base de datos.

Grails se encargará de implementar todos los métodos necesarios para que este dominio funcione al momento de la compilación del programa. De esta forma el código fuente permanece simple y el programador no tiene que preocuparse por la implementación de los métodos repetitivos de interacción con la base de datos. Los métodos implementados automáticamente incluyen:

```
boolean save()
void delete()
Object get(String id)
Int count()
Array List()
```

Además, Grails implementa automáticamente métodos de búsqueda de acuerdo a los campos declarados en el dominio. Para nuestro ejemplo del jugador, los métodos implementados serían:

```
Array findAllByNombre(String nombre)
Array findAllByApellido(String apellido)
Array findAllByEquipo(Equipo equipo)
Array findAllByNombreAndApellido(String nombre, String apellido)
Array findAllByApellidoAndEquipo(String apellido, Equipo equipo)
Array findAllByNombreAndEquipo(String nombre, Equipo equipo)
Object findByNombre(String nombre)
Object findByApellido(String apellido)
Object findByEquipo(Equipo equipo)
Object findByNombreAndApellido(String nombre, String apellido)
Object findByApellidoAndEquipo(String apellido, Equipo equipo)
Object findByNombreAndEquipo(String nombre, Equipo equipo)
```

Además se implementan métodos de búsqueda utilizando operaciones básicas de acuerdo al tipo de dato del atributo por ejemplo:

```
Object findByNombreLike(String nombre)
Object findByNombreNotLike(String nombre)
Object findByNombreIlike(String nombre)
Object findByNombreIsNotNull(String nombre)
Object findByNombreIsNull(String nombre)
Object findByNombreInList(Array nombres)
```

Todos estos métodos son implementados al momento de la compilación, es decir, el programador no los va a ver reflejados en el código del dominio Jugador, pero podrá utilizarlos en cualquier parte del programa.

Como podemos ver, al seguir la convención de ubicar a nuestro dominio dentro de la carpeta "domains", hemos ahorrado mucho tiempo al no tener que implementar todos los métodos repetitivos de interacción con la base de datos. Este es uno de los principales elementos por los que se considera a Grails como un framework de alta productividad.

Para entender cómo funciona Grails es necesario analizar cada uno de los sus tres componentes principales: Groovy, Spring y Hibernate.

4.5.1.3. Groovy

Groovy es un lenguaje orientado a objetos que proporciona capacidades dinámicas, de fácil uso e integración con la JVM (Java virtual machine, en español: máquina virtual de Java). Aprovecha todas las bondades de Java y añade poderosas funcionalidades inspiradas en lenguajes dinámicos como Python, Ruby, Perl y Smalltalk. Su principal atractivo es que, a pesar de que propone una sintaxis mucho más intuitiva y simple, soporta cualquier programa o porción de código escrito en Java, reduciendo así el tiempo de aprendizaje y adaptación para programadores versados en el lenguaje Java.

Entre las principales características de Groovy se encuentran:

- Soporte para Closures
- Sintaxis simplificada para el uso de mapas y listas
- Soporte nativo para manejo de XML y lenguajes de etiquetas
- Compatibilidad con el 100% de librerías desarrolladas para Java
- Groovlets para implementar Servlets utilizando Groovy scripts
- Groovy SQL para facilitar el uso de las librerías de conexión a la base de datos
- Ant Scripting
- Soporte nativo para expresiones regulares

- Compila bytecode 100% compatible con la JVM (java virtual machine)

Como ya hemos mencionado, el principal objetivo de Groovy es ser un lenguaje más dinámico, intuitivo y fácil de usar que Java. Para demostrarlo empecemos con una muestra del lenguaje, usando el típico ejemplo de imprimir una cadena "Hola Mundo".

```
public class HolaMundo {
   public static void main( String[] args )
      System.out.println("Hola Mundo!");
   }
}
```

El código en realidad está escrito Java, este ejemplo muestra algo muy importante sobre el lenguaje Groovy, y es que cualquier clase/objeto Java también es una clase/objeto Groovy.

Para el siguiente ejemplo haremos uso de las siguientes características de Groovy:

- Groovy soporta tipos de datos dinámicos, como JavaScript o Perl, por lo que en la declaración del método main() podemos omitir el uso de String[] e inclusive el tipo de retorno void.
- La visibilidad por omisión es public así que podemos omitirla también.
- Existe un método para salida de caracteres llamado println, como el de Perl, que se resulta más "amigable" que llamar a System.out.println.
- Los paréntesis son opcionales cuando se hace una llamada a un método.
- El ';' al final de cada sentencia también es opcional.

Incorporamos los cambios y obtenemos

```
class HolaMundo {
   static main( args ){
      println "Hola Mundo!"
   }
}
```

A pesar de que Groovy es un lenguaje dinámico, esto no significa que sea un lenguaje interpretado, como JavaScript o Perl, sino que cada clase se compila a código byte (cosa que lo diferencia mucho de los otros lenguajes que se ejecutan en una JVM), y sin embargo es posible escribir y ejecutar código Groovy como scripts, de tal forma que el código más pequeño que podemos escribir en Groovy para solucionar el ejemplo "Hola Mundo" es el siguiente.

```
println "Hola Mundo!"
```

También es posible definir clases en un script y ejecutar sentencias sobre instancias de la misma, así:

```
class HolaMundo {
   def saluda( nombre ){
      "Hola ${nombre}!"
   }
}

def hm = new HolaMundo()
println hm.saluda("Groovy")
```

En este ejemplo hemos hecho uso de otras 3 características de Groovy:

- El valor de retorno del método saluda no es específico, por lo que usamos la palabra reservada def.
- No es necesario definir la palabra reservada return, la última sentencia ejecutada determinará el valor de retorno, en este caso la creación de una cadena de caracteres.
- La cadena de caracteres no es un java.lang.String cualquiera, en realidad es GString. Los GStrings permiten la evaluación de expresiones regulares, de la misma manera que se hace en Perl.

Por último, en el siguiente ejemplo veremos como el lenguaje Groovy simplifica el trabajo del programador, implementando métodos prácticos y útiles que, a fin de cuentas, logran aumentar la productividad del desarrollador. A continuación se presenta un programa completo escrito en Java, que abre e imprime el contenido de un archivo:

Clase `SampleFileReader` y clase principal (`main`)

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;

public class SampleFileReader {
    static public String readFile(File file) {
        StringBuffer contents = new StringBuffer();
        try {
            BufferedReader reader = new BufferedReader(new FileReader(file));
            try {
                String line = null;
                while((line = reader.readLine()) != null) {

contents.append(line).append(System.getProperty("line.separator"));
                }
            } finally {
                reader.close();
            }
        } catch(IOException ex) {
             contents.append(ex.getMessage());
            ex.printStackTrace();
        }
        return contents.toString();
    }

    public static void main(String[] args) {
        File file = new File("C:\\temp\\test.txt");
```

```
        System.out.println(SampleFileReader.readFile(file));
    }
}
```

En Groovy el mismo programa sería reemplazado por las siguientes dos líneas:

```
def f = new File("C:\\temp\\test.txt")
f.eachLine { println it }
```

Dentro de una aplicación desarrollada en Grails, encontraremos que la capa del modelo y del controlador están escritas en lenguaje Groovy, mientras que en la capa de la vista encontraremos solo HTML y JavaScript. Esta es una característica importante del framework, que a diferencia de lenguajes como PHP, Grails no permite escribir código de servidor dentro de un archivo .gsp (Groovy server pages, que reemplazan a los tradicionales Java server pages, jsp), por lo que la aplicación se mantiene mucho más ordenada y su mantenimiento y actualización son más sencillos de realizar. En su lugar, Grails implementa el uso de etiquetas, las cuales hacen referencia a código escrito en Groovy en archivos específicos que se encuentran en el directorio "tagLib" dentro de la aplicación.

Por defecto, Grails contiene etiquetas predefinidas que por lo general son suficientes para cubrir las necesidades del desarrollador, sin embargo, permite añadir etiquetas personalizadas escritas por el mismo programador. Esta característica es muy importante puesto que dichas etiquetas pueden ser reutilizadas en cualquier proyecto desarrollado en Grails, lo cual implica que el programador puede personalizar el funcionamiento del framework a su gusto.

Una vez que la aplicación esté lista para correrse, Grails compilará todo el código escrito en Groovy en archivos .class, los cuales pueden ser interpretados por la JVM.

Otra característica importante del framework, es que incluye un servidor web incorporado (Apache Tomcat en la versión actual de Grails: 2.1.2). Al momento de la ejecución de la aplicación, Grails compila todo el código Groovy, genera el archivo .war (Web aplication Archive), sube el servidor web y despliega la aplicación. Como ventaja adicional, Grails recompila automáticamente los archivos que detecta que han sido modificados por el programador, y refleja los cambios realizados en la aplicación sin que el desarrollador tenga que preocuparse de tener que recompilar, generar y desplegar la aplicación con cada cambio que realice.

4.5.1.4. Spring

Spring es un framework de código abierto utilizado para desarrollar aplicaciones en la plataforma Java. Su principal característica, es que proporciona una base sólida y probada sobre la cual se puede implementar cualquier tipo de aplicación desarrollada en Java. Spring se enfoca en controlar los aspectos de nivel más bajo de la aplicación, liberando de esta forma a los desarrolladores, para que centren su trabajo en implementar la lógica de negocio.

El MVC web de Spring está diseñado sobre un DispatcherServlet, o controlador frontal, que reparte las peticiones a diferentes manejadores, que pueden configurarse para tener diferentes resoluciones de visualización, diferentes temas, diferentes idiomas. Sus características principales son:

- Clara separación de los roles: controlador, validador, objetos de comando, objetos de formulario, objetos de modelo, el controlador frontal, el manejador de mapeo, las vistas, etc.
- Fácil configuración del framework y de las clases, incluyendo referencias entre contextos.
- Adaptabilidad, no intrusismo: se puede definir cualquier método en los controladores.
- Código reutilizable: se pueden reutilizar los objetos en lugar de copiarlos
- Validación configurable: se utilizan fechas localizadas al idioma configurado, números con presentación con formato, etc. en lugar de cadenas de texto. Utiliza además validación a nivel de la aplicación para evitar, por ejemplo, errores de tipos de datos.
- Manejador de mapeo y de resolución de las vistas configurables con un simple URL, o más globalmente.
- Taglib (librería de tags) que permite una mayor flexibilidad al momento con los marcadores de código.

La Figura 13 muestra el proceso de las peticiones con el MVC de Spring.

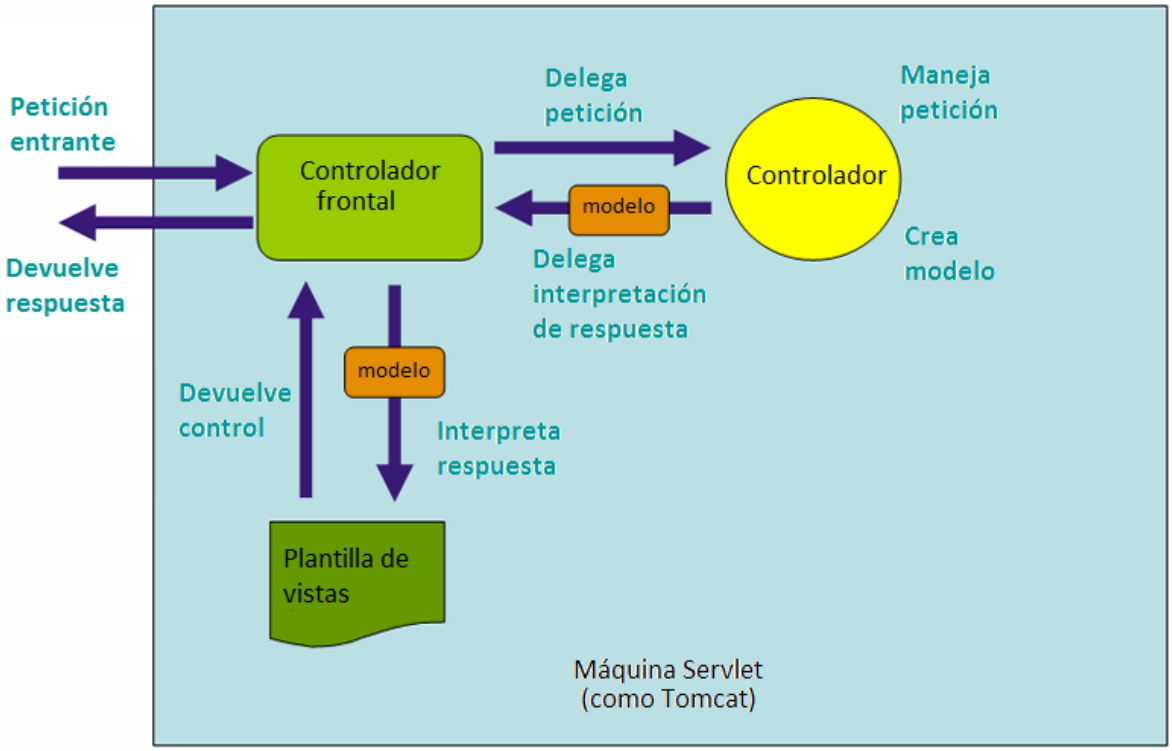


Figura 13: MVC Web de Spring
Autor: springsource.org
Fuente: http://static.springsource.org/spring/docs/3.0.x/reference/mvc.html

Grails utiliza el MVC de Spring, ya que se pueda extender fácilmente. Utiliza el controlador frontal de Spring y un controlador del MVC de Spring que maneja todas las peticiones de los controladores de Grails. Las peticiones se manejan a partir del URI de la siguiente manera (ver Figura 14):

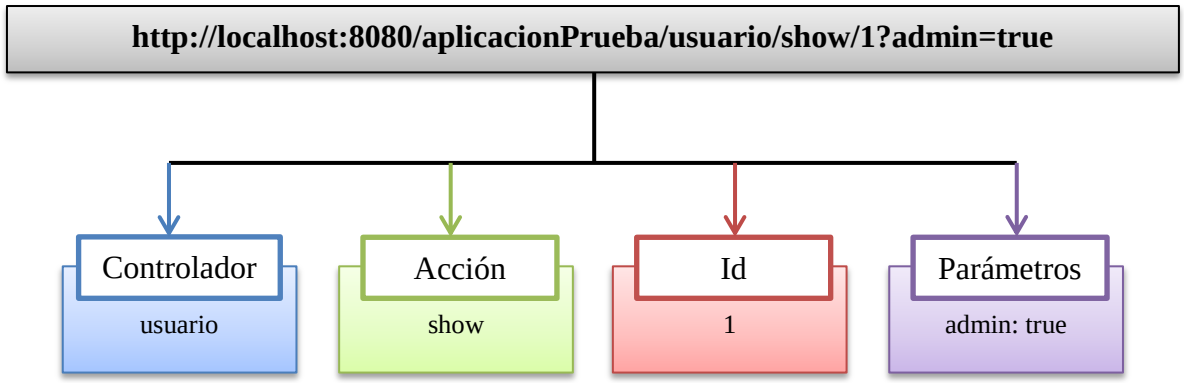


Figura 14: MVC Web de Spring aplicado a Grails
Autores:Luz Marina Unda – Valentin Zapata

1. Se analiza el URI para determinar sus componentes (nombre del controlador, nombre de la acción, id, parámetros)

2. Se busca la instancia del controlador correspondiente al URI
3. Se crea una instancia del controlador
4. Se configura los métodos dinámicos y propiedades de la instancia del controlador
5. Se obtiene el scaffolding si está activado para el controlador
6. Se obtiene la referencia a la acción correspondiente al URI
7. Se incrementa el scope flash para que pase al siguiente estado
8. Se obtiene el nombre de la vista correspondiente al URI
9. Se ejecutan los interceptores previos si existen
10. Si los interceptores previos no retornaron falso, se ejecuta la acción del controlador
11. Se crea una instancia del ModelAndView (ModeloYVista) del MVC de Spring a partir del nombre de la acción y del modelo retornado por la misma
12. Se ejecutar los interceptores posteriores si existen, pasando el modelo retornado
13. Se retorna la instancia del ModelAndView del MVC de Spring

4.5.1.5. Hibernate

Hibernate es una librería ORM (object-relational mapping, en español: mapeo objeto-relacional) que permite al programador mapear el modelo conceptual de la resolución del problema (la capa del Modelo en el patrón de diseño MVC) a una base de datos relacional, eliminando así la necesidad de crear modelos de tipo entidad-relación, ya que Hibernate se encargará de mapear el diagrama de clases a la base de datos.

Hibernate se encarga además de implementar los métodos de inserción, modificación, eliminación y búsqueda para cada una de las clases mapeadas a la base de datos. Con esto se libera al programador de tener que repetir tediosas implementaciones de métodos con alto contenido de SQL para controlar las iteraciones con la base de datos. Esto se traduce en una ventaja fundamental, puesto que al utilizar una librería ORM, conseguimos independizar completamente al programa de la base de datos. Al cederle el control de la capa de persistencia a Hibernate, este se encarga de resolver la compatibilidad entre los tipos de datos Java o Groovy con los de la base de datos.

Otra característica importante de Hibernate es que nos ofrece la posibilidad de realizar consultas complejas a la base de datos, mediante el uso del HQL o Hibernate Query

Language (lenguaje de consultas Hibernate). HQL contiene poderosas funcionalidades que nos permiten usar funciones de agregación, agrupación, junturas (joins) y subconsultas, pero con una sintaxis simple y coherente con nuestro modelo conceptual de clases. Para explicarlo es necesario un ejemplo. Supongamos una clase *empleado* con los atributos *nombre*, *apellido*, *cédula* y *sueldo*. Nuestro objetivo será realizar una consulta para recuperar todas las tuplas de la base de datos de empleados cuyo nombre empiece con "J" y su sueldo sea mayor 200, usando HQL.

```
Empleado.findAll("from empleado where nombre like 'J%' and sueldo > 200")
```

Como podemos ver, utilizamos el método "findAll()" de la clase Empleado y como parámetro enviamos un String con la consulta HQL. Para escribir la consulta como tal, utilizamos los nombres de las clases y de sus atributos tal como están declarados en nuestro modelo. Además podemos usar operadores de comparación y de agrupación igual que como lo haríamos en una consulta SQL.

Volviendo al ejemplo de la clase Jugador, vamos a explicar cómo Grails se integra con Hibernate.

```
class Jugador {
    String nombre
    String apellido
    Equipo equipo
    static constraints = {
        nombre(blank: false,nullable: false,size: 3..30)
        apellido(blank: false,nullable: false,size: 5..30)
        equipo(blank:true,nullable: true)
    }
    static mapping = {
        cache usage: 'read-write', include: 'non-lazy'
        id generator: 'identity'
        table 'jgdr'
        id column: 'jgdr__id'
        nombre column: 'jgdrnmbr'
        apellido column: 'jgdrapll'
        equipo column: 'eqpo__id'
    }
}
```

Hemos añadido una última sección a nuestro dominio: dentro "mapping" pondremos todo el mapeo para que Hibernate convierta nuestra clase en una entidad dentro de la base de datos. Como podemos ver, en esta sección especificamos el nombre que queremos que los atributos de la clase tengan en la tabla de la base de datos, así por ejemplo el campo "nombre" se mapeará como "jgdrnmbr". Cabe destacar que en todos y cada uno de los dominios de una aplicación en Grails, existe un atributo "id" que será la clave primaria de la tabla una vez que sea mapeada a la base de datos. Hibernate no soporta claves primarias complejas, es decir, de más de un campo. Pero esto no limita de ninguna forma la

complejidad de las aplicaciones que podemos desarrollar, puesto que siempre podremos resolver el problema usando claves primarias simples y, de ser necesario, se puede mapear con el atributo "unique" a los campos que formarían parte de la clave primaria compleja.

4.5.1.6. PostgreSQL

PostgreSQL es un sistema de administración de base de datos orientada a objetos (ORDBMS por sus siglas en inglés) basado en POSTGRES v. 4.2 desarrollado en la universidad de California en el Departamento de ciencias de la computación en Berkeley. POSTGRES fue pionero en varios conceptos que no estuvieron disponibles en varias bases de datos comerciales sino hasta mucho después.

PostgreSQL es un descendiente de código abierto del original de Berkeley y ha estado en desarrollo por más de 20 años. Es reconocido por su capacidad de procesamiento y desempeño en aplicaciones con múltiples consultas concurrentes.

Está disponible para diferentes plataformas y se ha convertido en uno de los administradores de bases de datos más populares. Soporta gran parte de los estándares de SQL y cumple con el estándar ANSI SQL. Ofrece muchas características modernas como consultas complejas, claves foráneas, triggers, vistas, integridad transaccional, control de concurrencia de multiversión. Además PostgreSQL puede ser extendido por el usuario de varias maneras, por ejemplo, agregando tipos de datos, funciones, opredores, funciones agregadas, lenguajes procedurales. Puede mantenerse corriendo por largos periodos de tiempo y requiere poco o ningún mantenimiento en la mayor parte de casos.

Es uno de los proyectos de código abierto y libre más antiguos, y puede ser utilizado, modificado y distribuido por cualquiera y de manera gratuita, ya sea para uso privado, comercial o académico.

Soporta diferentes lenguajes de programación, como PL/pgSQL, PL/Java, PL/Ruby, PL/Python, PL/PHP, PL/Perl, C/C++, PL/Tcl, PL/Scheme; y diferentes librerías de interface, como libpq, JDBC, ODBC, .NET, Perl, python, Ruby, C/C++, PHP, Lisp, Scheme, Qt.

4.5.1.7. JavaScript, JQuery y JQuery UI

4.5.1.7.1 JavaScript

JavaScript es un lenguaje de scripting, desarrollado por Brendan Eich en 10 días para Netscape en Abril de 1995 como un lenguaje interpretado y ligero que complemente a

Java. Eich es un programador con una larga experiencia creando lenguajes de programación, y su reto fue crear un lenguaje rico y poderoso sin utilizar la sintaxis de orientación a objetos de Java. Quiso embeber características avanzadas para que el lenguaje parezca simple y ligero en un principio pero de tal manera que programadores más experimentados pudieran explotar su poder subyacente. Este lenguaje debería servir para programadores no profesionales, como el Visual Basic de Microsoft, y debía ser interpretable para poder ser embebido en páginas Web sin requerir software ni lenguajes de servidor. Su sintaxis es intencionalmente similar a Java y a C++ para facilitar su aprendizaje. Eich creó un lenguaje con un modelo de objetos simplificado que combina la estructura de C, la sintaxis de Java, la herencia basada en prototipos de Self, las funciones de primera clase de Scheme, los patrones de SmallTalk y la simetría entre datos y código de LISP. Puede funcionar como un lenguaje orientado a objetos o como un lenguaje procedural. El núcleo, o core, incluye funcionalidad para trabajar con texto, arreglos, fecha y expresiones regulares, pero no incluye nada de entrada y salida de datos. Estas características, junto con funcionalidad más sofisticada como manejo de redes, almacenamiento o gráficos son responsabilidad del entorne en el que se incluye JavaScript, generalmente el navegador web.

JavaScript se ejecuta completamente del lado del cliente, en el navegador, por lo que es responsable de la mayor parte de la interactividad de una página Web, como cambios de imágenes, cambio de color de texto al pasar el mouse por encima, rastro o huellas tras el mouse, etc. También ha sido ampliamente utilizado para validar formularios antes de que estos sean enviados al servidor. Actualmente le pertenece a Oracle y es utilizado en millones de páginas web y aplicaciones de servidor alrededor del mundo.

Contrariamente a lo que el nombre pueda sugerir, JavaScript no es “Java interpretado”, ni un subconjunto de Java. El nombre fue solamente parte de una táctica de marketing de Netscape Communications. Inicialmente se llamó Livescript, pero cuando Sun Microsystems liberó Java y se volvió popular, Netscape cambió el nombre a JavaScript.

Todos los navegadores modernos, ya sea para computadores, consolas de video juegos, tabletas, teléfonos, incluyen un intérprete de JavaScript, por lo que existen diferentes implementaciones. La primera, creada por Brendan Eich, fue llamada SpiderMonkey, ha sido actualizada con el tiempo y se le han agregado optimizaciones como TraceMonkey (para Firefox 3.5), IonMonkey o JägerMonkey (a partir de Firefox 4), mientras que Google

utiliza V8, Safari, JavaScriptCore y Opera utiliza Carakan. Internet Explorer utiliza Chakra, que técnicamente es un motor de JScript, y no de JavaScript.

JavaScript estuvo en los navegadores por casi 10 años cuando comenzó la revolución de Ajax. Microsoft inició la dominación de Ajax en las interfaces Web cuando agregó la característica de XMLHHTP-Request a su Internet Explorer. Otros navegadores rápidamente agregaron características similares para permitirle a JavaScript recuperar información de servidores y actualizar el documento HTML sin actualizar la página entera. Esta innovación permitió la creación de aplicaciones con interfaces de usuario muy ricas y altamente interactivas, cada vez más cercanas a la experiencia de escritorio, como Google Mail o Google Maps.

JavaScript tiene diferentes usos al momento de crear una página web interactiva. Permite por ejemplo agregar animaciones, interactividad y efectos visuales dinámicos a los documentos HTML. Principalmente permite dar retroalimentación inmediata al usuario, lo que hace páginas Web más útiles. Por ejemplo, un carrito de compras puede mostrar instantáneamente el total de las compras, los impuestos, los costos de envío, todo dependiendo de ciertas selecciones hechas por el cliente. Se puede también hacer un carrusel animado que presenta fotografías, o insertar información oculta que se muestra únicamente bajo petición del usuario, ya sea con un clic de botón, o bajo forma de ayudas al pasar el mouse sobre otro elemento.

4.5.1.7.2 JQuery

JavaScript puede presentar algunos problemas al momento de programar. Aunque es un lenguaje simple, es un lenguaje de programación y a muchos usuarios, como diseñadores web, puede parecerles una tarea difícil. Aparte de esto, los diferentes navegadores, al utilizar cada uno una implementación diferente de JavaScript, entienden el código de diferente manera. Así, un programa que funciona perfectamente en Google Chrome puede no funcionar en Internet Explorer 9. Esto provoca horas de pruebas y modificaciones tediosas al código para lograr un funcionamiento uniforme en todos los navegadores.

Por estas razones se crearon las librerías de JavaScript. Una librería es un conjunto de funciones JavaScript que proveen soluciones simples a muchos problemas cotidianos de programación. Estas librerías de JavaScript facilitan la tarea de programación y resuelven los problemas de compatibilidad entre navegadores. Con estas librerías se pueden escribir

pocas líneas de código y conseguir la misma funcionalidad que hubiera necesitado muchas líneas y horas de pruebas utilizando solamente JavaScript.

Las librerías de JavaScript permiten facilitar las tareas repetitivas, como seleccionar elementos, agregar, mostrar o eliminar contenido, modificar atributos de etiquetas HTML o sus estilos, determinar valores de campos de formularios, reaccionar ante acciones del usuario, encontrar elementos en la página para manipularlos, entre otras. Estas operaciones parecen bastante básicas, pero su implementación puede complicarse bastante, especialmente si es necesario que funcione en los principales navegadores.

Existen diferentes librerías de JavaScript, entre las más importantes se encuentran DOJO, Yahoo, Prototype, JQuery, MochiKit.

Para este trabajo hemos seleccionado JQuery por diferentes razones: es gratuita, es muy ligera (la librería comprimida pesa alrededor de 90k); es muy fácil de aprender y de personalizar; hay una gran cantidad de documentación, ejemplos tutoriales, foros, etc. disponibles pues dispone de una comunidad de desarrolladores muy grande que continua creciendo; la cantidad de plug-ins que tiene y que permiten agregar características muy avanzadas de manera muy fácil y rápida, y la facilidad de crear plug-ins nuevos, creados a medida para las características deseadas.

JQuery utiliza selectores para identificar grupos de elementos y retorna un objeto que los representa para poder manipularlos. Las diferentes funciones de JQuery normalmente retornan el elemento utilizado, lo que permite encadenar acciones. Las acciones que permite ejecutar JQuery son muy variadas. Entre ellas está obtener y fijar o modificar atributos HTML, clases CSS, valores de elementos de formularios, contenido de elementos como *div*s, tamaños y posiciones, data. También permite alterar la estructura del documento agregando, remplazando, eliminando, copiando, envolviendo, mostrando u ocultando elementos. Otra característica importante de JQuery es que permite manejar los diferentes eventos de los diferentes elementos. Por ejemplo, se puede controlar lo que hace un clic del mouse sobre un cierto elemento, o un cambio de valor en un elemento de un formulario. Los eventos son muy variados e incluyen el clic, doble clic, presión de una tecla, mouse sobre o fuera de un elemento, etc.

4.5.1.7.3 JQuery UI

JQuery UI no es más que un grupo de elementos de interacción con el usuario, efectos, temas visuales desarrollados sobre JQuery. Permite construir páginas y aplicaciones web altamente interactivas agregando elementos como selectores de fecha en forma de calendario, botones con imágenes que se ven igual en todos los navegadores, acordeones, ventanas de diálogo, pestañas; interacciones como arrastrar y soltar, seleccionar, ordenar, cambiar tamaño; efectos como agregar o quitar clases CSS con una transición animada, mostrar y ocultar con transiciones. Sobre esta base de elementos, efectos e interacciones existen además un gran número de plugins que permiten ampliar la funcionalidad, facilitando la creación de nuevos elementos, por ejemplo, una galería que muestra miniaturas de las fotografías mientras pasan una por una en un mayor tamaño. La documentación existente es muy amplia, por lo que la creación de plugins nuevos es una tarea bastante simple, permitiendo así además crear plugins si no existe uno que tenga la funcionalidad deseada.

4.5.1.7.4 Cuadro resumen

A continuación presentamos un cuadro para resumir las características principales de las herramientas seleccionadas.

Herramienta	**Tipo**	**Características**
Grails	Lenguaje de servidor	• Basado en java • Framework de alta productividad • Usa el patrón de diseño MVC de Spring • Usa el lenguaje de programación Groovy • Usa la librería Hibernate para las interacciones con la base de datos • Genera bytecode

		compatible con la máquina virtual de java • Software libre
PostgreSQL	Motor de base de datos	• Robusto • Es uno de los proyectos de código abierto y libre más antiguos. • Ampliamente documentado. • Soporta diferentes lenguajes de programación, como PL/pgSQL, PL/Java, PL/Ruby, PL/Python, PL/PHP, PL/Perl, C/C++, PL/Tcl, PL/Scheme; y diferentes librerías de interface, como libpq, JDBC, ODBC, .NET, Perl, python, Ruby, C/C++, PHP, Lisp, Scheme, Qt.
Jquery	Librería para el manejo de javascript	• Simplifica el uso de javascript. • Implementa selectors para facilitar el navegación por el árbol DOOM. • Fácil de aprender • Tiene gran cantidad de

		plugins para agregar funcionalidad. • Personalizable • Permite manejar los eventos de los diferentes elementos en una página web.

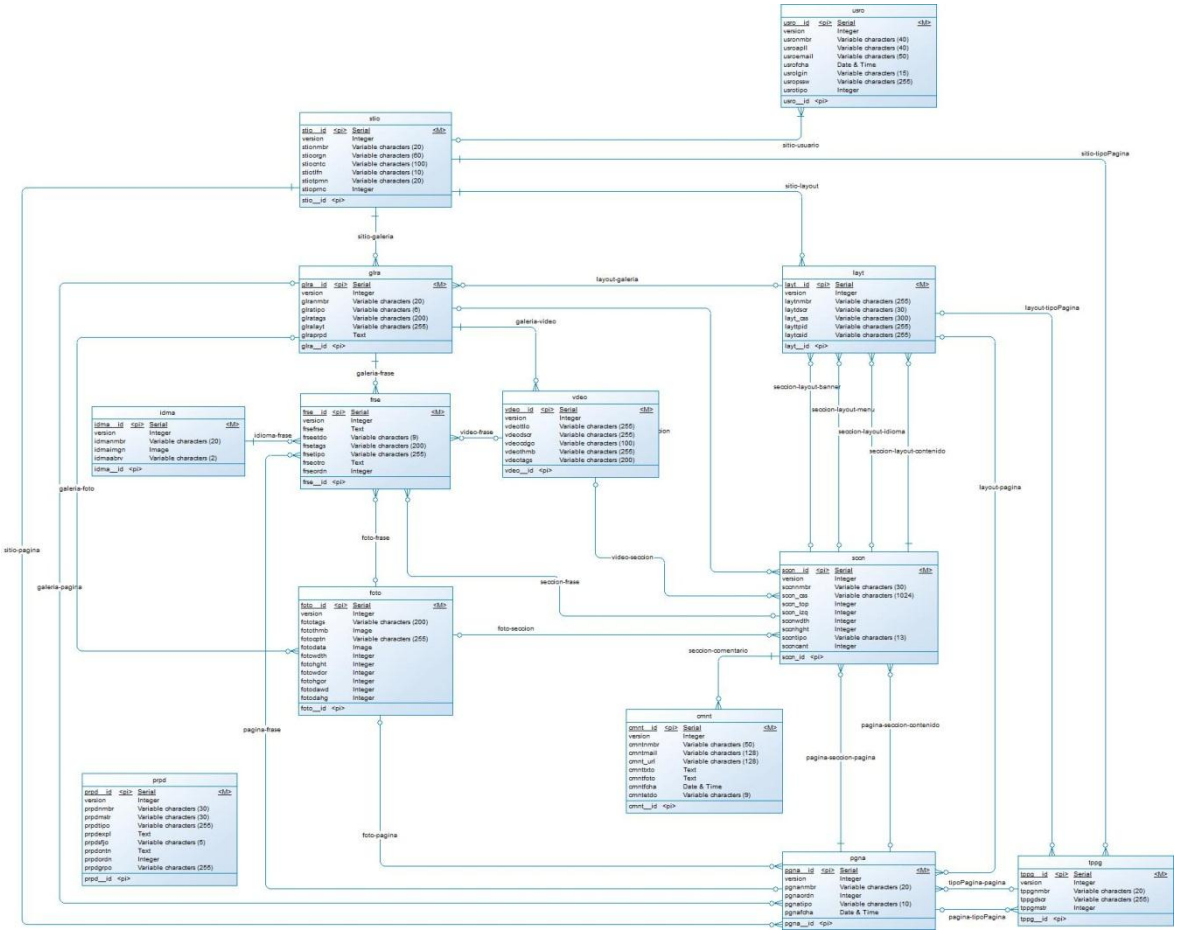
4.5.2. Transformación del modelo de clases Entidad a Entidad-Relación

4.5.2.1. Reglas

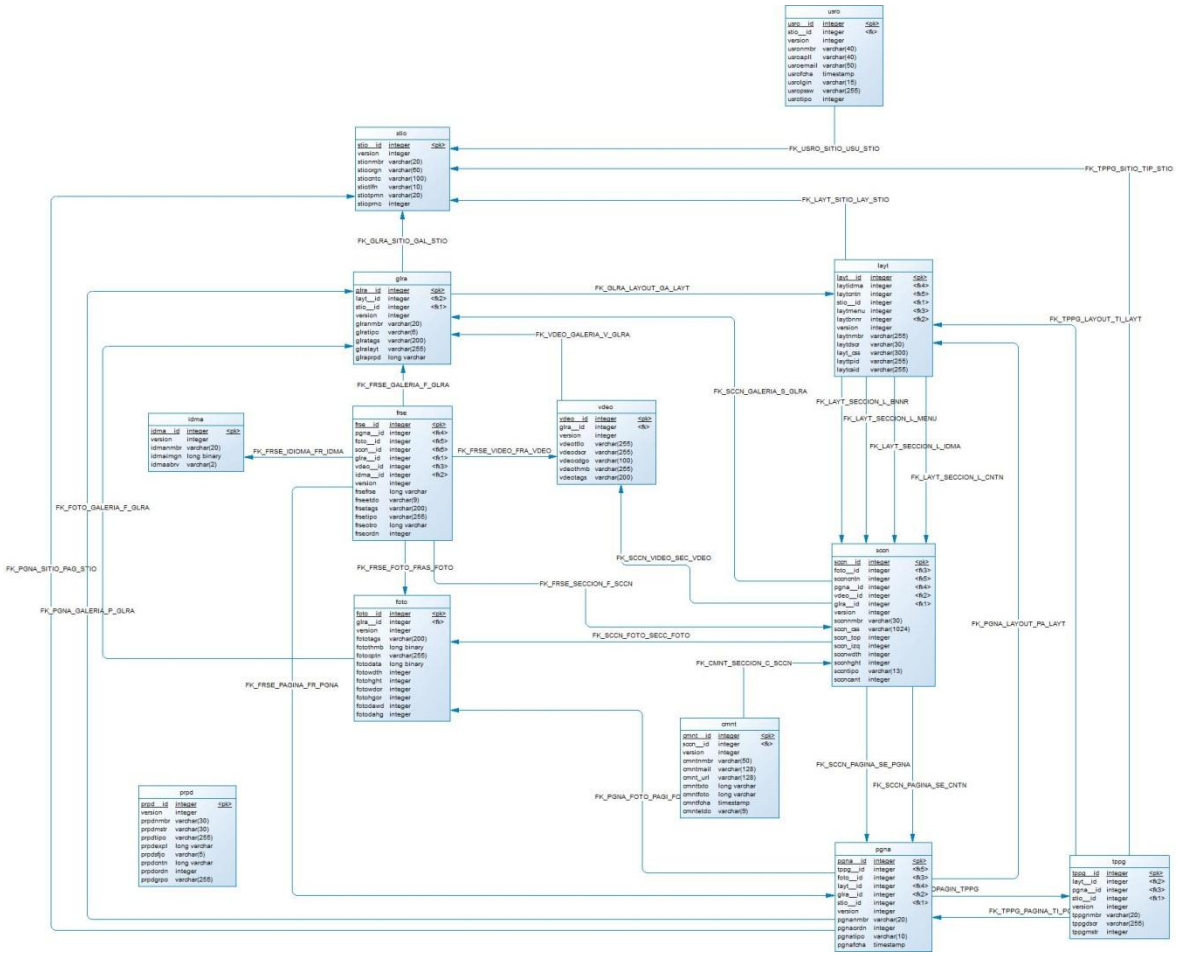
6. Toda clase entidad es representada por una entidad en el diagrama Entidad-Relación
7. El nombre de las entidades se construye de la siguiente manera:
    - Los nombres tienen siempre 4 letras
    - Se utilizan las primeras letras del nombre de la clase, utilizando siempre la primera letra y las 3 siguientes consonantes.
    - En caso de no haber suficientes consonantes se utilizan las vocales
    - En caso de no haber suficientes letras, se duplica la última letra.
    - Ejemplo: Propiedad → prpd
8. El nombre de los campos se construye de la siguiente manera:
    - Los nombres tienen siempre 8 letras
    - Las primeras 4 son del nombre de la tabla.
    - Las siguientes 4 utilizan las mismas reglas que el nombre de las entidades, con excepción del caso en que no hay suficientes letras, en el que se utilizan guiones bajos para completar los 4 caracteres.
    - Ejemplo: Propiedad/nombre → prpdnmbr
9. Todas las claves primarias son simples y secuenciales.

4.5.2.2. Modelo conceptual

El modelo conceptual se podrá encontrar en tamaño real en el Anexo4: Modelo conceptual.


usro
usro_id <pi> Serial <M>
version Integer
usronmbr Variable characters (40)
usroapll Variable characters (40)
usroemail Variable characters (50)
usrofcha Date & Time
usrologin Variable characters (15)
usropswe Variable characters (256)
usrotipo Integer
usro_id <pi>
sitio-usuario
sitio
stio__id <pi> Serial <M>
version Integer
stionmbr Variable characters (20)
stioorgn Variable characters (60)
stiocntc Variable characters (100)
stictlfn Variable characters (10)
stiotpmn Variable characters (20)
stiopmc Integer
stio__id <pi>
sitio-tipoPagina
sitio-layout
sitio-galeria
glra
glra__id <pi> Serial <M>
version Integer
glranmbr Variable characters (20)
glratipo Variable characters (5)
glratags Variable characters (200)
glratayt Variable characters (255)
glrapopd Text
glra__id <pi>
layout-galeria
galeria-video
galeria-frase
layt
layt__id <pi> Serial <M>
version Integer
laytnmbr Variable characters (255)
laytclsr Variable characters (30)
layt_css Variable characters (300)
layttpid Variable characters (255)
laytpid Variable characters (255)
layt__id <pi>
layout-tipoPagina
seccion-layout-banner
seccion-layout-menu
seccion-layout-idioma
seccion-layout-contenido
layout-pagina
idma
idma__id <pi> Serial <M>
version Integer
idmanmbr Variable characters (20)
idmaimgn Image
idmaabrv Variable characters (2)
idma__id <pi>
idioma-frase
galeria-foto
frse
frse__id <pi> Serial <M>
version Integer
frsetitle Text
frsestdo Variable characters (9)
frsetags Variable characters (200)
frsetipo Variable characters (255)
frsecdto Text
frseordn Integer
frse__id <pi>
video-frase
video
video__id <pi> Serial <M>
version Integer
videotitlo Variable characters (256)
videoclsr Variable characters (256)
videocdgo Variable characters (100)
videothmb Variable characters (256)
videotags Variable characters (200)
video__id <pi>
sitio-pagina
galeria-pagina
foto-frase
video-seccion
seccion-frase
foto
foto__id <pi> Serial <M>
version Integer
fototags Variable characters (200)
fotothmb Image
fotoaptn Variable characters (255)
fotodata Image
fotowdth Integer
fotohght Integer
fotowdor Integer
fotohgor Integer
fotodawd Integer
fotodahg Integer
foto__id <pi>
foto-seccion
scon
scon__id <pi> Serial <M>
version Integer
sconnmbr Variable characters (30)
scon_css Variable characters (1024)
scon_top Integer
scon_lrg Integer
sconwdth Integer
sconhght Integer
scontipo Variable characters (13)
sconcant Integer
scon_id <pi>
seccion-comentario
omnt
omnt__id <pi> Serial <M>
version Integer
omntnmbr Variable characters (50)
omntmail Variable characters (128)
omnt_url Variable characters (128)
omntbdo Text
omntfoto Text
omntfcha Date & Time
omntestdo Variable characters (9)
omnt__id <pi>
pagina-seccion-contenido
pagina-seccion-pagina
pagina-frase
prpd
prpd__id <pi> Serial <M>
version Integer
prpdnmbr Variable characters (30)
prpdmail Variable characters (30)
prpdtloc Variable characters (255)
prpdexpl Text
prpdstjo Variable characters (5)
prpdordn Text
prpdordn Integer
prpdgroo Variable characters (255)
prpd__id <pi>
foto-pagina
pgna
pgna__id <pi> Serial <M>
version Integer
pgnanmbr Variable characters (20)
pgnaordn Integer
pgnatipo Variable characters (10)
pgnafcha Date & Time
pgna__id <pi>
tipoPagina-pagina
pagina-tipoPagina
tppg
tpgg__id <pi> Serial <M>
version Integer
tppgnmbr Variable characters (20)
tppgdscr Variable characters (255)
tppgmstr Integer
tpgg__id <pi>

4.5.2.3. Modelo Físico



4.5.2.4. Diccionario de datos

4.5.2.4.1 Comentario (cmnt)

Comentarios de los visitantes de la página en el guestbook

Columna	Campo	Descripción
cmnt__id	id	Id de la tabla (secuencial)
cmntnmbr	nombre	Nombre de la persona que deja el comentario
cmntmail	mail	E-mail de la persona que deja el comentario
cmnt_url	url	URL de la persona que deja el comentario
cmnttxto	texto	Comentario
cmntfoto	foto	Foto del comentario (sube la foto en una carpeta y guarda el path)
cmntfcha	fecha	Fecha y hora en la que se hizo el comentario
cmntetdo	estado	Estado del comentario
sccn__id	seccion	Sección (guestbook) a la que pertenece el comentario

4.5.2.4.2 Foto (foto)

Fotografías para las páginas y galerías

Columna	Campo	Descripción
foto__id	id	Id de la tabla (secuencial)
glra__id	galeria	Galería a la que pertenece la foto
fototags	tags	Tags para identificar la foto
fotothmb	thumbnail	Información de la miniatura de la foto
fotocptn	caption	Título de la foto para identificarla
fotodata	data	Información de la foto
fotowdth	width	Ancho de la foto
fotohght	height	Alto de la foto
fotowdor	widthOriginal	Ancho original de la foto
fotohgor	heightOriginal	Alto original de la foto
fotodawd	dataWidth	Ancho calculado de la foto después del redimensionamiento
fotodahg	dataHeight	Alto calculado de la foto después del redimensionamiento

4.5.2.4.3 Frase (frse)

Frases de las páginas, fotos, videos, etc.

Columna	Campo	Descripción
frse__id	id	Id de la tabla (secuencial)
frsefrse	frase	Texto de la frase
pgna__id	pagina	Página a la que pertenece la frase
idma__id	idioma	Idioma en el que está la frase
frseetdo	estado	Estado de la frase
sccn__id	seccion	Sección a la que pertenece la frase
glra__id	galeria	Galería a la que pertenece la frase
foto__id	foto	Foto a la que pertenece la frase
vdeo__id	video	Video al que pertenece la frase
frsetags	tags	Tags para identificar la frase
frsetipo	tipo	Tipo de frase

frseotro	otros	
frseordn	orden	Orden de la frase

4.5.2.4.4 Galería (glra)

Galerías de fotos o contenidos

Columna	Campo	Descripción
glra__id	id	Id de la tabla (secuencial)
glranmbr	nombre	Nombre de la galería
glratipo	tipo	Tipo de galería
glratags	tags	Tags para identificar la galería
glraprpd	propiedades	Propiedades visuales de la galeria
layt__id	layoutContenido	Layout a usar para los contenidos
stio__id	sitio	Sitio al que pertenece la galería

4.5.2.4.5 Idioma (idma)

Idiomas para las frases

Columna	Campo	Descripción
idma__id	id	Id de la tabla (secuencial)
idmanmbr	nombre	Nombre del idioma
idmaimgn	imagen	Imagen del idioma, utilizado para el selector de idioma
idmaabrv	abreviación	Abreviación del idioma, utilizado para el selector de idioma

4.5.2.4.6 Layout (layt)

Layouts o plantillas para la creación de páginas y contenidos

Columna	Campo	Descripción
layt__id	id	Id de la tabla (secuencial)
laytnmbr	nombre	Nombre del layout
laytdscr	descripción	Descripción del layout
laytcntn	contenido	Sección del contenido del layout
laytmenu	menu	Sección del menú del layout
laytbnnr	banner	Sección del banner del layout
laytidma	idioma	Sección del selector de idioma del layout
layt_css	css	Estilos CSS del layout
stio__id	sitio	Sitio al que pertenece el layout
layttpid	tipoIdioma	Tipo de selector de idioma que utiliza el layout
laytcsid	cssIdioma	CSS del selector de idioma que utiliza el layout

4.5.2.4.7 Página (pgna)

Páginas de contenido del sitio

Columna	Campo	Descripción
pgna__id	id	Id de la tabla (secuencial)
stio__id	sitio	Sitio al que pertenece la página
pgnanmbr	nombre	Nombre de la página

pgnaordn	orden	Orden de la página en el menú
layt__id	layout	Layout de la página
pgnatipo	tipo	Tipo de página
foto__id	foto	Foto que identifica la página
glra__id	galeria	Galería a la que pertenece la página
tppg__id	tipoPagina	Tipo de página
pgnafcha	fecha	Fecha de creación de la página

4.5.2.4.8 Propiedades (prpd)

Propiedades para las galerías, las secciones y las páginas

Columna	**Campo**	**Descripción**
prpd__id	id	Id de la tabla (secuencial)
prpdnmbr	nombre	Nombre de la propiedad (la que se va a utilizar en el css o js)
prpdmstr	mostrar	Nombre de la propiedad (la que se va a mostrar al usuario)
prpdtipo	tipo	Tipo de propiedad
prpdexpl	explicación	Explicación corta de lo que hace la propiedad
prpdsfjo	sufijo	Sufijo de la propiedad (por ejemplo, px)
prpdcntn	contenido	Contenido de la propiedad (es decir, número, color, texto, etc.)
prpdordn	orden	Orden de la propiedad
prpdgrpo	grupo	Grupo de la propiedad (por ejemplo, Ubicación)

4.5.2.4.9 Sección (sccn)

Secciones que componen una página

Columna	**Campo**	**Descripción**
sccn__id	id	Id de la tabla (secuencial)
pgna__id	pagina	Página a la que pertenece la sección
sccnnmbr	nombre	Nombre de la sección
sccn_css	css	Estilo CSS de la sección
sccn_top	top	Posición de la sección desde el borde superior de la página
sccn_izq	izq	Posición de la sección desde el borde izquierdo de la página
sccnwdth	width	Ancho de la sección
sccnhght	heigh	Alto de la sección
sccntipo	tipo	Tipo de sección (puede ser texto, foto, video, etc.)
foto__id	foto	Foto que contiene la sección
vdeo__id	video	Video que contiene la sección
glra__id	galeria	Galería que contiene la sección
sccncntn	contenido	Página que contiene la sección
sccncant	cantidad	Cantidad de contenido a mostrar

4.5.2.4.10 Sitio (stio)

Sitios que contienen páginas

Columna	**Campo**	**Descripción**
stio__id	id	Id de la tabla (secuencial)
stionmbr	nombre	Nombre del sitio

stioorgn	organizacion	Organización a la que pertenece el sitio
stiocntc	contacto	Nombre de la persona de contacto
stiotlfn	telefono	Teléfono de la persona de contacto
stiotpmn	tipoMenu	Tipo de menú que va a utilizar el sitio
stioprnc	principal	Indica si es o no el sitio principal

4.5.2.4.11 Tipo de Página (tppg)

Tipos de página

Columna	**Campo**	**Descripción**
tppg__id	id	Id de la tabla (secuencial)
tppgnmbr	nombre	Nombre del tipo de página
tppgdscr	descripcion	Descripción del tipo de página
stio__id	sitio	Sitio al que pertenece el tipo de página
pgna__id	pagina	Página en la que se muestra, solo valido para publicaciones
layt__id	layout	Layout con el que se muestra el tipo de página
tppgmstr	mostrar	Atributo que determina si se muestra en el menú de tipos de página

4.5.2.4.12 Usuario (usro)

Usuario para la utilización del sistema

Columna	**Campo**	**Descripción**
usro__id	id	Id de la tabla (secuencial)
stio__id	sitio	Sitio al que pertenece el usuario
usrotipo	tipo	Tipo de usuario
usronmbr	nombre	Nombre del usuario
usroapll	apellido	Apellido del usuario
usrolgin	login	Login que utilizará el usuario para ingresar al sistema
usropssw	pass	Password que utilizará el usuario para ingresar al sistema
usroemail	email	Email del usuario
usrofcha	fecha	Fecha de creación del usuario

4.5.2.4.13 Video (vdeo)

Videos para utilizar en las páginas

Columna	**Campo**	**Descripción**
vdeo__id	id	Id de la tabla (secuencial)
vdeottlo	titulo	Título del video
vdeodscr	descripción	Descripción del video
vdeocdgo	codigo	Código del video (de YouTube para hacer embed)
vdeothmb	thumbnail	Thumbnail del video (de YouTube)
glra__id	galeria	Galería a la que pertenece el video
vdeotags	tags	Tags para identificar el video

4.5.3. Estándares de codificación

Para la codificación del sistema manejador de contenidos utilizamos los siguientes estándares:

Elemento	Nombre	Descripción
Proyecto	`Nth`	Es el nombre que escogimos para nuestro pryecto
Paquete	`nth`	Al tener un nivel de complejidad no muy elevado decidimos crear un solo paquete.
Dominios	Nombre del elemento, comenzando con mayúscula	El nombre de los dominios es el nombre del elemento que representa, siempre comenzando con una mayúscula, sin tildes ni caracteres especiales. Por ejemplo, `Pagina, Seccion, Foto,` etc.
Controladores	Nombre del elemento, comenzando con mayúscula, más la palabra `Controller` al final.	El nombre de los controladores es el nombre del elemento que va a controlar, siempre comenzando con una mayúscula, sin tildes ni caracteres especiales más la palabra `Controller` al final. Por ejemplo, `PaginaController, SeccionController, FotoController`, etc.
GSP (Groovy Server Page)	Nombre representativo de los contenidos de la página	El nombre de las páginas es representativo de su contenido. Por ejemplo, `nuevaGaleria.gsp, fotosGal.gsp, textosGal.gsp, editarPagina.gsp`, etc.
Variables	Nombre representativo de la variable	El nombre de las variables es representativo de lo que almacenan, por ejemplo, la variable que almacena la altura de una foto es `height`, el ancho es `width`, el nombre de una página se almacena en `nombre`, etc.

4.5.4. Aplicación de pruebas de desarrollo

A continuación presentamos un formulario de aplicación de pruebas de desarrollo. La totalidad de estos formularios podrá encontrarse en el Anexo 5: Formularios de aplicación de pruebas de desarrollo.

Nombre: Valentin Zapata **Fecha:** 12-01-2013

Módulo: Usuarios **Caso de uso:** F1

Prueba: Ingresar un nuevo usuario al sistema

Precondición: Al menos un usuario y un sitio ingresados en el sistema.

Resultado esperado: Pantalla de lista de usuarios desplegada correctamente y con todos los botones funcionales. Muestra el formulario y permite ingresar datos en los campos. Al hacer clic en Guardar se almacena la información

Prueba 1
Resultado obtenido: El usuario se guardó exitosamente y la pantalla de lista de usuarios se desplegó completamente funcional **¿Pasa la prueba?** SI NO **Errores encontrados:** La contraseña del usuario no se guardó con la debida codificación (MD5) **Solución:** Codificar la contraseña a MD5 antes de guardarla en la base de datos. **Comentarios, sugerencias, mejoras:** Ninguno

Prueba 2
Resultado obtenido: El usuario se guardó exitosamente y la pantalla de lista de usuarios se desplegó completamente funcional **¿Pasa la prueba?** SI NO **Errores encontrados:** Ninguno **Comentarios, sugerencias, mejoras:** Mejorar los mensajes de validación en los campos requeridos

5. PUESTA EN FUNCIONAMIENTO

En este capítulo se describe el proceso de implantación así como también el resultado del plan de pruebas de usuarios.

5.1. Implantación

5.1.1. Instalación

La implantación del proyecto se hizo en el laboratorio de la Facultad de Ingeniería, con autorización previa de la Ing. Susana Mazapanta. Se nos asignó una computadora con las siguientes características:

- Procesador: Intel Pentium 4 3GHz
- Memoria RAM: 500MB
- Disco duro: 100GB

El primer paso fue la instalación del software de base. Para este proyecto escogimos utilizar CentOS versión 6.0. Para reducir al mínimo los recursos y memoria necesarios para que el sistema operativo funcione, decidimos instalarlo sin el módulo de interfaz gráfica y realizar todas las configuraciones a través de la consola de comandos.

Una vez terminada la instalación del sistema operativo procedimos a instalar el software necesario para que funcione el gestor de contenido:

- Base de datos: PostgreSQL versión 9.1
- Lenguaje de programación: Java versión 1.6
- Servidor web: Apache Tomcat versión 7
- Framework: Grails versión 1.3.7

En cuanto a PostgreSQL, fue necesario modificar el archivo de configuración para restringir conexiones externas a la base de datos, y modificar el tipo de retorno de los datos de tipo BYTEA a una cadena de bytes. Además cambiamos la contraseña del usuario "postgres" como medida de seguridad.

El siguiente paso fue configurar el sistema operativo añadiendo las siguientes variables de entorno:

- JAVA_HOME: referencia al lugar de instalación de Java
- GRAILS_HOME: referencia al lugar de instalación de Grails
- PATH: agregando las referencias a las variables anteriores

Estas variables son necesarias para que funcione el servidor web Tomcat y para utilizar los comandos del Framework Grails en la consola del sistema operativo.

En cuanto al servidor web, modificamos el archivo "catalina.sh" para aumentar la cantidad de memoria RAM disponible para la ejecución del gestor de contenido. Además, modificamos el puerto por el cual el servidor web responde a las peticiones del usuario al puerto 80 (por defecto es 8080).

Como configuración final en el sistema operativo, desactivamos el servicio HTTPD (servicio que corresponde al servidor web Apache para aplicaciones basadas en PHP y está activado por defecto en la distribución de CentOS) y activamos los servicios de la base de datos y del servidor web Tomcat.

Una vez probado el funcionamiento tanto del servidor web como de la base de datos, modificamos el código fuente del gestor de contenidos para actualizar la contraseña del usuario "postgres" para la conexión con base de datos. A continuación, utilizando el comando `grails war` generamos el archivo .war necesario para que Tomcat despliegue la aplicación.

El paso final de la instalación fue copiar el archivo war generado a la carpeta "webapps" dentro del directorio de instalación de Tomcat, para que el servidor web despliegue la aplicación.

Después de probar la accesibilidad a la aplicación dentro de la red local, procedimos a crear un usuario de tipo administrador y a entregar oficialmente el producto.

Actividad	**Responsable**
Instalación de CentOS 6.0	Valentin Zapata
Instalación de PostgreSQL 9.1	Luz Marina Unda
Instalación de Java	Luz Marina Unda
Instalación de Apache Tomcat 7	Luz Marina Unda
Instalación de Grails 1.3.7	Luz Marina Unda
Configuración de PostgreSQL	Valentin Zapata
Configuración de variables de entorno	Valentin Zapata
Configuración de Apache Tomcat	Valentin Zapata
Configuración de CentOS	Valentin Zapata
Instalación del gestor de contenidos NTH	Valentin Zapata
Pruebas	Valentin Zapata, Luz Marina Unda

5.1.2. Capacitación

Una vez completadas la instalación y la configuración del sistema, procedimos a capacitar a una alumna designada por el Ing. Eddy Sánchez para que haga pruebas. Para esto nos reunimos en el laboratorio de la Facultad durante un periodo inicial de 2 horas.

En primer lugar hicimos una demostración del producto para que la alumna tuviera una idea de su funcionalidad. A continuación enseñamos a utilizar el sistema creando un sitio de prueba de cero, creando páginas, diferentes secciones, como texto, fotos, videos, galerías de fotos y de contenido y links.

5.2. Pruebas

5.2.1. Aplicación de pruebas de usuario

A continuación presentamos la aplicación de pruebas de usuario al Ing. Eddy Sánchez, responsable del departamento de Acción Social de la Facultad de Ingeniería de la Pontificia Universidad Católica del Ecuador. La totalidad de los formularios de pruebas a usuario podrán encontrarse en el Anexo 6: Formularios de aplicación de pruebas de usuario.

Nombre: Eddy Sanchez **Fecha:** 11-04-2013

1. ¿Pudo ingresar al sistema utilizando sus credenciales?
   SI ✓ NO
   Errores o problemas: _____
   _____
   Comentarios o sugerencias: _____
   _____

1. ¿Pudo acceder correctamente a las pantallas principales del sistema utilizando los botones del menú principal?
   SI ✓ NO
   Errores o problemas: _____
   _____
   Comentarios o sugerencias: _____
   _____

2. ¿Pudo insertar exitosamente elementos nuevos al sistema?

Elemento	¿Se insertó correctamente?	Errores o problemas encontrados	Sugerencias o comentarios
Usuario	SI ✓ NO		
Sitio	SI ✓ NO		
Layout	SI ✓ NO		
Página	SI ✓ NO		
Sección	SI ✓ NO		
Foto	SI ✓ NO		
Video	SI ✓ NO		
Texto	SI ✓ NO		
Galería de fotos	SI ✓ NO		
Galería de contenidos	SI ✓ NO		
Guestbook	SI ✓ NO		
Comentario de guestbook	SI ✓ NO		

3. ¿Pudo visualizar correctamente la información de los elementos existentes?

Elemento	¿Se visualizó correctamente?	Errores o problemas encontrados	Sugerencias o comentarios
Usuario	SI ✓ NO		
Sitio	SI ✓ NO		
Layout	SI ✓ NO		
Página	SI ✓ NO		
Sección	SI ✓ NO		
Foto	SI ✓ NO		
Video	SI ✓ NO		
Texto	SI ✓ NO		
Galería de fotos	SI ✓ NO		
Galería de contenidos	SI ✓ NO		
Guestbook	SI ✓ NO		
Comentario de guestbook	SI ✓ NO		
Menú	SI ✓ NO		
Página web	SI ✓ NO		

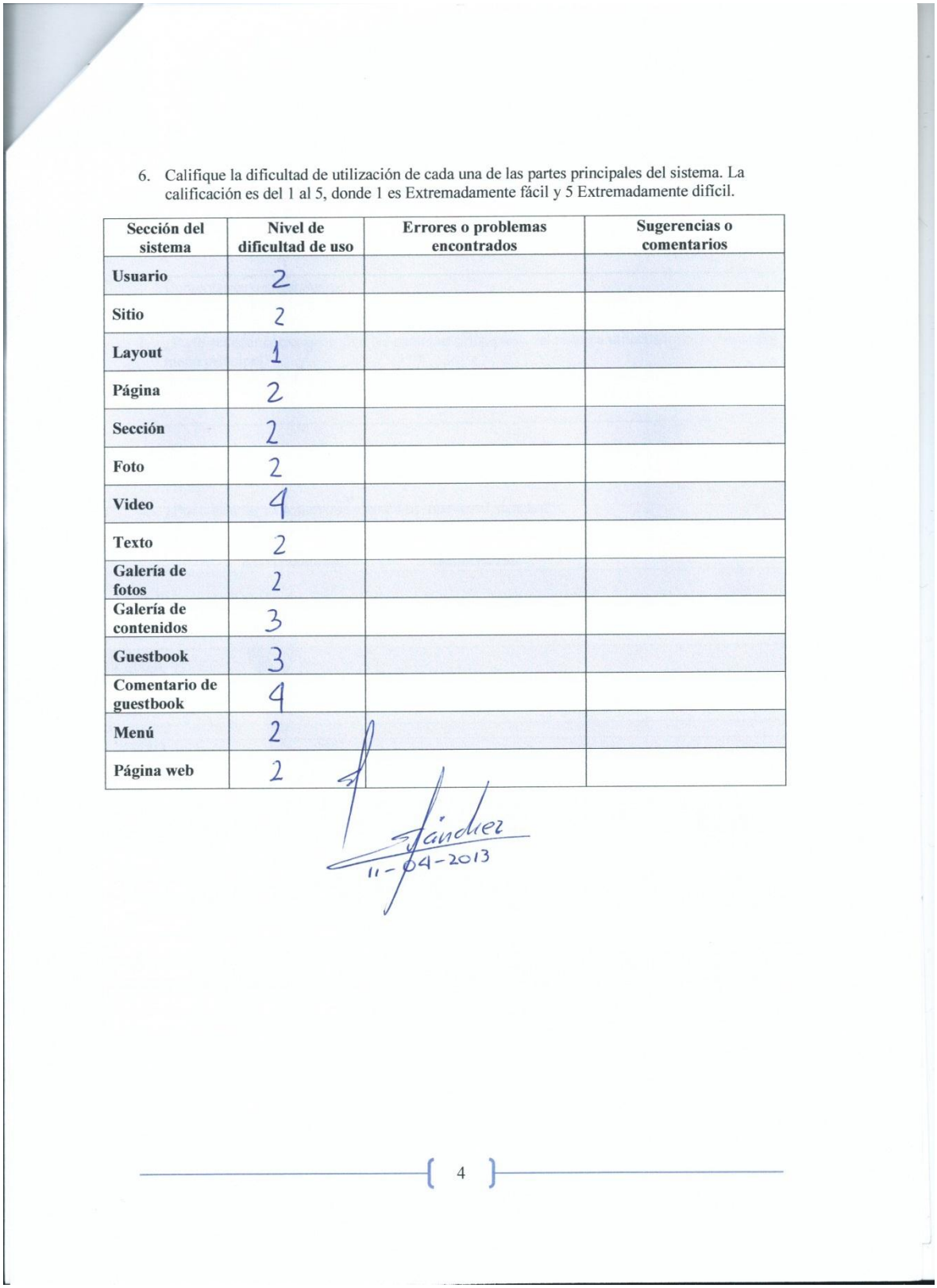
4. ¿Pudo modificar exitosamente los campos de un elemento existente?

Elemento	¿Se modificó correctamente?	Errores o problemas encontrados	Sugerencias o comentarios
Usuario	SI NO ✓	falta el boton	
Sitio	SI ✓ NO		
Layout	SI ✓ NO		
Página	SI ✓ NO		
Sección	SI ✓ NO		
Foto	SI ✓ NO		

Video	SI ✓ NO		
Texto	SI ✓ NO		
Galería de fotos	SI ✓ NO		
Galería de contenidos	SI ✓ NO		
Guestbook	SI ✓ NO		
Menú	SI ✓ NO		

5. ¿Pudo eliminar correctamente un elemento existente?

Elemento	¿Se eliminó correctamente?	Errores o problemas encontrados	Sugerencias o comentarios
Usuario	SI ✓ NO		
Sitio	SI ✓ NO		
Layout	SI ✓ NO		
Página	SI ✓ NO		
Sección	SI ✓ NO		
Foto	SI ✓ NO		
Video	SI ✓ NO		
Texto	SI ✓ NO		
Galería de fotos	SI NO ✓	Se elimino fotos pero no galeria	
Galería de contenidos	SI NO ✓	No se elimino	
Guestbook	SI ✓ NO		
Comentario de guestbook	SI ✓ NO		
Menú	SI ✓ NO		
Página web	SI ✓ NO		



6. Califique la dificultad de utilización de cada una de las partes principales del sistema. La calificación es del 1 al 5, donde 1 es Extremadamente fácil y 5 Extremadamente difícil.

Sección del sistema	Nivel de dificultad de uso	Errores o problemas encontrados	Sugerencias o comentarios
Usuario	2		
Sitio	2		
Layout	1		
Página	2		
Sección	2		
Foto	2		
Video	4		
Texto	2		
Galería de fotos	2		
Galería de contenidos	3		
Guestbook	3		
Comentario de guestbook	4		
Menú	2		
Página web	2		

Sandier
11-04-2013

4

5.2.2. Ajustes en base a sugerencias del usuario

Después de completado el formulario de pruebas de usuario, fue necesario realizar algunos cambios en la interfaz gráfica, basados en sugerencias de los usuarios finales, para

facilitar el uso del sistema especialmente en las pantallas de edición de páginas y layouts. Además se corrigieron pequeños errores de validación en los formularios de inserción de datos.

Un cambio representativo fue la inclusión de un campo “imagen ”de tipo `BYTE[]` en el dominio Idioma, lo cual derivó en cambios en todas las pantallas de la administración de idiomas, y en la construcción de una nueva interfaz para subir imágenes al servidor.

Se agregó también un campo “imagen ”de tipo `BYTE[]` en el dominio Comentario, el cual permitirá a los visitantes de la página agregar fotos junto con sus comentarios en la sección de Guestbook. Este cambio implicó modificar no solamente el formulario de ingreso de comentario, sino también el área de edición de Guestbook.

Después de realizar todos los cambios, estos fueron revisados y aprobados por los usuarios del sistema.

5.2.3. Presentación de la versión final

Después de realizados los ajustes, se procedió a instalar la versión final del sistema NTH en la computadora asignada por el cliente y a hacerle la entrega de una copia del Manual de Usuario. La versión ejecutable de esta versión se encuentra en el Anexo 7: Versión ejecutable, bajo forma de un archivo `nth.war`, que se debe desplegar utilizando un servidor Apache Tomcat. El Manual de Usuario se encuentra en el Anexo 8: Manual de usuario.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- El desarrollar un CMS en lugar de utilizar uno de los existentes, como Joomla o Droopal, permite aprender mucho, y crear un producto que se ajuste a las necesidades del que lo va a utilizar.
- La utilización de un framework como Grails, de alta productividad, automatiza las tareas repetitivas y permite centrar el trabajo en el desarrollo de la parte funcional del programa.
- Es necesario mantener nuestros conceptos e ideas presentes durante todo el proceso de desarrollo, de esta manera aseguramos que el producto final sea único y tenga un toque de originalidad y de pertenencia.

6.2. Recomendaciones

- Recomendamos utilizar frameworks únicamente cuando se tiene claro cómo funcionan y el código que generan, pues de otra manera la generación de código se vuelve un impedimento más que una ayuda.
- Recomendamos utilizar proyectos de software libre de alto prestigio y reconocimiento, de esta forma estaremos seguros de contar con gran cantidad de documentación que nos será de mucha ayuda durante el proceso de desarrollo.
- Utilizar herramientas para el versionamiento, tanto del código como de todos los documentos generados durante el proceso de desarrollo, nos ahorrara mucho tiempo y evitara perdidas de información.

BIBLIOGRAFÍA

- **Extreme Programming**
  - Don Wells, Extreme Programming: A gentle introduction, última modificación hecha el 28 de septiembre de 2009, visitada por última vez el 8 de abril de 2013, XP Extreme Programming, http://www.extremeprogramming.org/
  - Ronald E. Jeffries, XProgramming.com An Agile Software Development Resource, visitada por última vez el 8 de abril de 2013, XProgramming.com, http://xprogramming.com/
  - JavaTM Extreme Programming Cookbook, By Eric M. Burke, Brian M. Coyner, Publisher: O'Reilly, Pub Date: March 2003, ISBN: 0-596-00387-0
- **Metodologías ágiles**
  - Don Wells, Agile Software Development: A gentle introduction, visitada por última vez el 8 de abril de 2013, Agile Process, http://www.agile-process.org/
  - Ward Cunningham, Manifesto for Agile Software Development, visitada por última vez el 8 de abril de 2013, Agile Manifesto, http://agilemanifesto.org/
  - Darren Levy, The Lean Approach to Agile Software Development, 15 de mayo de 2009, visitada por última vez el 8 de abril de 2013, GatherSpace, http://www.gatherspace.com/static/agile_software_development.html
  - Agile Sherpa, A brief history of Agile, visitada por última vez el 8 de abril de 2013, agileSHERPA by and for the agile community, http://www.agilesherpa.org
  - Agile Alliance, Guide to Agile Practices, visitada por última vez el 8 de abril de 2013, Agile Alliance, http://guide.agilealliance.org/
  - Agile software development methods , Review and analysis , By Pekka Abrahamsson, Outi Salo, Jussi Ronkainen & Juhani Warsta, Publisher Julkaisija – Utgivare, 2002, ISBN: 1235-0621
- **FDD**
  - Nebulon Pty. Ltd., Agile Software Development using Feature Driven Development (FDD), visitada por última vez el 8 de abril de 2013, Nebulon Pty. Ltd., http://www.nebulon.com/fdd/
  - Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Highsmith, James A. III, 2000, Dorset House Publishing.
  - Stephen Palmer, An Introduction to Feature-Driven Development, 20 de noviembre de 2009, visitada por última vez el 8 de abril de 2013, DZone Agile Zone, http://agile.dzone.com/articles/introduction-feature-driven
  - Scott W. Ambler, Feature Driven Development (FDD) and Agile Modeling, visitada por última vez el 8 de abril de 2013, Agile Modeling, http://www.agilemodeling.com/essays/fdd.htm
  - Peter Coad, Eric Lefebvre, Jeff De Luca, Feature-Driven Development, visitada por última vez el 8 de abril de 2013, http://www.petercoad.com/download/bookpdfs/jmcuch06.pdf
- **Scrum**
  - Ken Schwaber, What is Scrum? , visitada por última vez el 8 de abril de 2013, Scrum.org Improving the Profession of Software Development, http://www.scrum.org/Resources/What-is-Scrum

- **CMS**
  - Margaret Rouse, content management system (CMS) , última modificación en enero 2011, visitada por última vez el 8 de abril de 2013, SearchSOA, http://searchsoa.techtarget.com/definition/content-management-system
  - Bernard Kohan, What is a Content Management System (CMS)?, última modificación el 15 de noviembre de 2010, visitada por última vez el 8 de abril de 2013, comentum catch the world, http://www.comentum.com/what-is-cms-content-management-system.html
  - Daniel Primo Gorgoso, Sistemas de gestión de contenidos (CMS), Nov-Dic 2007, visitada por última vez el 8 de abril de 2013, http://www.slideshare.net/delineas/introduccin-a-los-sistemas-de-gestion-de-contenidos-cms
  - Xavier García Cuerda, Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto , última modificación el 29 de noviembre de 2004, visitada por última vez el 8 de abril de 2013, Mosaic tecnologías y comunicación multimedia, http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/
  - http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos
  - http://en.wikipedia.org/wiki/Common_Gateway_Interface
- **MVC**
  - The Model-View-Controller (MVC), última modificación en diciembre 2011, visitada por última vez el 8 de abril de 2013, Trygve M. H. Reenskaug, http://heim.ifi.uio.no/~trygver/
  - The original MVC reports, última modificación en febrero 2007, visitada por última vez el 8 de abril de 2013, Trygve Reenskaug Dept. of Informatics University of Oslo, http://folk.uio.no/trygver/2007/MVC_Originals.pdf
  - Very Brief History Of MVC , última modificación en septiembre 2008, visitada por última vez el 8 de abril de 2013, KANIAN77′S weblog, http://kanian77.wordpress.com/2008/09/03/very-brief-history-of-mvc/
  - MVC - Part1: "Introduction and History", última modificación en marzo 2009, visitada por última vez el 8 de abril de 2013, Typo3 media, http://www.typo3-media.com/blog/mvc-mvp-typo3-introduction.html
  - What is MVC? , última modificación en agosto 2011, visitada por última vez el 8 de abril de 2013, Adrian Pop, http://www.thetechlabs.com/tech-tutorials/codeigniter-tutorial-codeigniter-explained/
  - MVC Architecture, última modificación en junio 2012, visitada por última vez el 8 de abril de 2013, Google, http://developer.chrome.com/apps/app_frameworks.html
  - Model View Controller: History, theory and usage, última modificación en marzo 2011, visitada por última vez el 8 de abril de 2013, Amir Salihefendic, http://amix.dk/blog/post/19615
  - A Description of the Model-View-Controller, última modificación en Diciembre 2008, visitada por última vez el 8 de abril de 2013, Glenn E. Krasner and Stephen T. Pope, http://www.itu.dk/courses/VOP/E2005/VOP2005E/8_mvc_krasner_and_pope.pdf

    - Spring MVC Integration, última modificación en Abril 2009, visitada por última vez el 8 de abril de 2013, Grails Org, http://grails.org/Developer+-+Spring+MVC+Integration

- **Javascript, JQuery, JQueryUI**
    - What is JavaScript?, última modificación en octubre 2012, visitada por última vez el 8 de abril de 2013, Mozilla, https://developer.mozilla.org/en-US/docs/JavaScript/About_JavaScript?redirectlocale=en-US&redirectslug=About_JavaScript
    - JavaScript & jQuery: The Missing Manual, Second Edition, by David Sawyer McFarland, October 2011, ISBN: 978-1-449-39902-3l
    - Programming languages on the internet, última modificación en enero 2010, visitada por última vez el 8 de abril de 2013, Webdevelopers, http://www.webdevelopersnotes.com/basics/languages_on_the_internet.php3
    - JavaScript: The Definitive Guide, Sixth Edition, by David Flanagan, March 2011, ISBN: 978-0-596-80552-4
    - Jquery, última modificación en abril 2013, visitada por última vez el 8 de abril de 2013, Jquery org, http://jquery.com/
    - JqueryUI, última modificación en abril 2013, visitada por última vez el 8 de abril de 2013, Jquery org, ,http://jqueryui.com/
- **Grails**
    - Groovy, última modificación en abril 2013, visitada por última vez el 8 de abril de 2013, Groovy Org, http://groovy.codehaus.org/
    - http://c2.com/cgi/wiki?GroovyLanguage
    - http://www.javabeat.net/2007/06/introduction-to-groovy-scripting-language/
    - http://www.genbetadev.com/java-j2ee/spring-framework-introduccion
    - http://picandocodigo.net/2010/introduccion-a-spring-framework-java/
    - http://www.springsource.org/spring-framework
    - http://docs.jboss.org/hibernate/orm/3.3/reference/en/html/queryhql.html
    - http://www.hibernate.org/
    - http://www.hibernate.org/about/orm.html
    - http://www.jamesward.com/2012/10/14/nosql-inside-sql-with-java-spring-hibernate-and-postgresql
- **PostgreSQL**
    - http://www.postgresql.org/docs/current/static/intro-whatis.html
    - http://www.enterprisedb.com/products-services-training/products/postgresql-overview
- **Pruebas**
    - http://www.scribd.com/doc/50206963/Pruebas-en-la-ingenieria-del-software