12

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

SISTEMA DE AUTOMATIZACIÓN DE INFORMACIÓN DE PROYECTOS DE INTERNET2

DISEÑO DE UN SISTEMA PARA UNA ORGANIZACIÓN

QUE PARA OBTENER

EL TÍTULO DE:

LICENCIADO EN INFORMÁTICA

PRESENTA:

WENDY PATRICIA NAVARRETE ESCOBEDO

ASESOR:

ACTUARIO DAVID MEJÍA RODRÍGUEZ

México, D.F. **2001**

Al amor, a la confianza y al respeto
que siempre han existido en mi familia
Gracias a mis Papás, Cristy y André
que son mi familia

A la amistad y al cariño
que nos ha unido
Gracias Antonio

Al tiempo e interés que se
le dedicó a este trabajo
Gracias German

Al estímulo profesional
que se da en las aulas
y fuera de ellas
Gracias David

ÍNDICE

INTRODUCCIÓN

Al igual que muchas otras cosas, los medios de comunicación masiva han estado en un proceso de constante evolución. Sin alejarnos demasiado de los tiempos actuales, remontaré a principios de siglo cuando la radio se convirtió en el medio de mayor alcance. En un inicio, sus grandes emisoras transmitían programas producidos y emitidos desde un punto central, tiempo después, aparecieron las emisoras de frecuencia modulada. Entonces comenzaron también las emisoras en las que era frecuente no sólo escuchar a los locutores sino también a los radioescuchas participar en los programas. De esta manera se les fue introduciendo a participar en los contenidos que se transmitían por radio: el hecho de poder votar por la canción favorita para que sea transmitida en el siguiente bloque; poder formular preguntas y recibir respuestas en los programas de asesoría; el anunciar productos o servicios; y poder lograr una comunicación entre los radioescuchas y las radiodifusoras, son algunos de los hechos que confirman la participación de las personas, que escuchan la radio, en la estructura de la programación radial, aunque dicha participación fuera muy escasa.

En la mitad del siglo veinte la televisión llega a ser un importante medio que difunde información no sólo en forma de ondas sonoras, sino que incluye imágenes en movimiento. De forma muy similar como sucede en la radio, los televidentes tienen la capacidad de participar en la emisión de cierta información, sin embargo es la emisora la que decide los contenidos de los programas, la forma de tratarlos y el horario al que se emiten, situación que supedita al televidente a recibir información no elegida por él.

En la década de los noventa, Internet se populariza y pone en contacto a una gran cantidad de personas alrededor del mundo. En Internet la mayor parte de la información se encuentra disponible a cualquier hora para cualquier parte del mundo que esté conectada a la red mundial; permite al usuario conseguir datos y discriminar entre los que son falsos y los que son ciertos; se puede ir a sitios desde donde se distribuye la información y ponerse en contacto con las personas que la generan para cuestionarla de manera personal y directa, y por supuesto involucrarse en la creación de contenidos.

Tanto la radio como la televisión trabajan con un modelo en el cual ellos son los que eligen los contenidos de información que se deben escuchar o ver, un modelo en el que cada parte juega únicamente un rol, emisor o receptor, pero no ambos. Internet está rompiendo con la estructura de este modelo y permite que las partes sean tanto receptores como emisores en un mismo tiempo.

Aún presentando grandes ventajas, la Internet sobre otros medios de comunicación, existen personas preocupadas por sacar el mayor provecho de esta red y sus servicios. Para ello hay una serie de investigaciones que se han unificado en un proyecto llamado Internet2.

Internet2 propiamente dicho no es una red que sustituirá la Internet de nuestros días, es un esfuerzo que se plasma en un conjunto de desarrollos de nuevas tecnologías e innovadoras aplicaciones que puedan ser empleadas en la red global Internet. Así como el correo electrónico y la world wide web, tan útiles en nuestros días, son la herencia de investigaciones "recientes", la herencia de Internet2 será el mejoramiento de las capacidades de transmisión de datos por Internet.

Se están desarrollando tecnologías tales como IPv6, multicasting o calidad de servicio (QoS) que permitirán revolucionarias aplicaciones dentro de Internet, tales como bibliotecas digitales,

laboratorios virtuales y centros de enseñanza a distancia. Una de las metas prioritarias para Internet2 es asegurar la transferencia de aplicaciones para la transmisión de contenidos educativos y la evolución de comunidades virtuales dentro de la red enfocadas a la educación.

México también participa en Internet2, a través de diversos proyectos que se están llevando a cabo en algunas instituciones académicas en nuestro país, por lo que surge la necesidad de difundir, compartir y discutir información derivada de estos proyectos y es la razón por la cual se presenta este trabajo escrito, que describe el desarrollo de un sistema que satisfaga esta necesidad.

Dado el modelo de comunicación que posee Internet, se utilizará como plataforma para este sistema, ya que no sólo se requiere de informar a un público sino que también se requiere de su participación activa.

En este documento el lector encontrará el resultado de llevar a cabo las tareas de analizar y diseñar los esquemas que estructuren un sistema de información que permita tanto a investigadores, como desarrolladores y personas involucradas e interesadas en proyectos de Internet2, puedan ser tanto emisores como receptores en este proceso de comunicación.

En primer lugar se presenta un capítulo en donde se presenta un marco teórico que define la metodología con la que fue desarrollado el sistema. Esto con la finalidad de esclarecer algunos de los términos que se utilizan a lo largo de la documentación.

El segundo capítulo, consiste en presentar una visión general del panorama dentro del cual se inicia el desarrollo del sistema así como el alcance del sistema, la definición de personas que harán uso de éste, y sus principales características.

La fase de análisis, se muestra en el tercer capítulo, la cual a través de la exploración de los principales procesos a cubrir por el sistema y tomados en cuenta los recursos disponibles se pudieron elaborar un modelo conceptual y la descripción detallada de dichos procesos.

Estudiando el posible comportamiento del sistema y esquematizando a detalle sus requerimientos, se obtiene el conjunto de componentes de software que contribuirán al logro del objetivo del sistema. Y es, en la parte correspondiente a la fase de diseño, donde el lector podrá encontrar, el modelo del sistema, es decir la solución lógica que cumple con los requerimientos, y que finalmente pueden ser implementados en un lenguaje de programación orientado a objetos, para darle vida al sistema de automatización de proyectos de Internet2.

MARCO TEÓRICO

Paradigma[1] orientado a objetos

La solución de un problema es un proceso que involucra en primer lugar el entendimiento de éste, posteriormente la creación de una resolución y finalmente llevar a cabo dicha resolución. A su vez, el hecho de entender un problema involucra representarlo en distintas formas (ideas); crear una resolución para el problema, involucra la manipulación de esas ideas para derivar hacia la idea que dé la solución deseada; y por último el llevar a cabo está resolución, es construir la propia solución. Bajo este esquema está el uso de un paradigma que determina posibles tipos de representaciones usadas para resolución de un problema, el paradigma de orientado a objetos.

El paradigma orientado a objetos, es una forma de conceptuar al mundo a través de cosas u objetos, cada uno de estos constituye una construcción que combina las estructuras de datos y el comportamiento en una sola unidad. Las raíces del paradigma se encuentran dentro de la teoría de las formas de Platón, y se centra básicamente en cuatro conceptos: abstracción, encapsulación, herencia y polimorfismo.

Existen dos conceptos básicos manejados por este paradigma, objeto y clase. Un objeto es una abstracción del mundo real que posee un estado (determinado por los valores de sus atributos) y un determinado comportamiento (que se proyecta a través de sus operaciones) Y una clase es una especificación de todas las operaciones y atributos para un tipo de objetos.

La Teoría de las Formas

Fue concebida por Platón en participación con Sócrates y Aristóteles para entender las preguntas relacionadas con la naturaleza del conocimiento. Esta teoría evolucionó a lo largo de tres periodos.

Primer periodo. Basados en la premisa de que *algo es comprendido si puede ser definido*, se propuso el método socrático para el encuentro de definiciones. Dicho método consistía en que un maestro guía a su alumno, en el reconocimiento de una conclusión, cuestionándolo de manera progresiva hasta que la conclusión es alcanzada por el propio alumno. Concluyeron que el aprendizaje involucra una recolección de ideas del alumno.

Segundo periodo. Platón asienta que las ideas, elementos básicos del conocimiento, son formas universales, en cambio las cosas del mundo real son formas particulares. En otras palabras las ideas son abstracciones del mundo real.

Tercer periodo. Platón redefine el método socrático en un procedimiento de colección y división, cuyo principio más importante, de muchos otros que componen este procedimiento, especifica que donde hay un número de elementos del mismo tipo que compartan un conjunto de características, hay una sola cosa que posee ese conjunto de propiedades.

[1] Reglas que se usan como fundamento para usar una tecnología.

Los pilares del paradigma

Abstracción es una simplificación de hechos. Es enfocarse en similitudes y diferencias de entre un conjunto de cosas para extraer características esenciales y evitar características irrelevantes, con el fin de formular una única representación que tenga estas características y que permita la definición de cada elemento en el conjunto.

Encapsulación es empaquetar el comportamiento de un objeto, ocultando los detalles de cómo trabajan internamente los objetos, facilitando de esta manera la abstracción

Herencia es una relación entre clases que define nuevas representaciones y permite que una instancia o representación posea las mismas características y funcionalidad que la clase a la que heredó.

Polimorfismo es la creación de nuevas instancias como variantes de las ya existentes.

Relación entre la Teoría de formas y el paradigma

La teoría de Formas define los conceptos fundamentales del paradigma orientado a objetos. En la siguiente tabla se muestra la relación entre la Teoría de las Formas y el paradigma.

Teoría de las Formas		**Paradigma orientado a objetos**
El método socrático	→	Abstracción
Ideas, elementos universales	→	Clases (y encapsulación)
Cosas, elementos particulares	→	Objetos (y encapsulación)
El procedimiento de colección y división	→	Herencia (y polimorfismo)

Debido a que el paradigma orientado a objetos está vivamente aplicado en la ingeniería de software, en este proyecto se utiliza una metodología bajo el paradigma orientado a objetos como una tecnología que proporciona una técnica de modelados de sistemas, donde un sistema se modela como un número de objetos que interactúan entre sí por medio de mensajes.

Desarrollo orientado a objetos

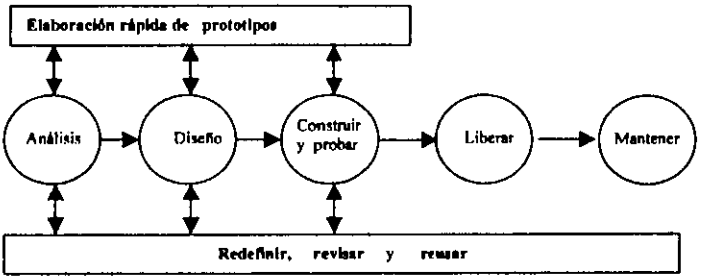
Es un proceso conceptual basado en abstracciones que existen en el mundo real, es fundamentalmente una manera de pensar cuyos beneficios vienen de ayudar a los desarrolladores y a clientes a expresar conceptos de manera clara y facilitar la comunicación entre ellos. En este contexto, desarrollo se refiere a un ciclo de desarrollo de software, divido en distintas fases: análisis, diseño e implementación.

Por otra parte, existe también el desarrollo estructurado introducido en la década de los 70's y adoptado por autores como Yourdon, DeMarco y Senn, entre otros. Principalmente se concentra en la especificación y descomposición de la funcionalidad del sistema. Esto parecería la manera más

directa para implementar el objetivo deseado, pero el sistema resultante puede ser frágil, ya que se asume que los requerimientos del cliente han sido correctamente capturados por los desarrolladores, lo cual no siempre resulta ser así y si los requerimientos tienen alguna modificación el cliente sufrirá los efectos de una reestructuración masiva en su sistema. Por otra parte, el tradicional ciclo de vida de sistemas creados bajo esta perspectiva, presenta la desventaja de ser secuencial, y no permite continuar con el proceso hasta no terminar por completo una fase del desarrollo, de igual modo presenta inflexibilidad para poder hacer ajustes dentro de etapas que ya se dieron por terminadas.

En contraste con el desarrollo estructurado, el desarrollo orientado a objetos se concentra en identificar los objetos del dominio de la aplicación y después en ajustarles, a cada uno de ellos, las funciones que desempeñaran dentro del sistema. Su ventaja se visualiza en cuanto a que se mantiene mejor en relación a los ajustes que se dan dentro de los requerimientos, ya que las modificaciones no se hacen en el sistema por completo sino en el o los objetos encargados de satisfacer el requerimiento que se está ajustando.

Etapas del desarrollo OO



Es posible que las etapas del desarrollo OO se realicen en forma paralela, sin embargo el punto de inicio es la etapa del análisis, dentro de la cual el desarrollador se enfocará principalmente en identificar el alcance del sistema, decidiendo que queda dentro y fuera de su comportamiento asimismo identificará los conceptos claves del negocio o conceptos del dominio, adoptando el vocabulario del negocio para un mejor entendimiento con el cliente, también se ocupará de investigar y documentar como el usuario interactuará con el sistema y de que manera es éste afectado. Y el último paso dentro de la etapa del análisis es la investigación del comportamiento del sistema, descubriendo objetos propios del negocio.

Durante la etapa del diseño el desarrollador refinará los diagramas que han sido resultado de la fase del análisis y continuará trabajando sobre ellos, posiblemente descubra nuevas características que deban ser modeladas y se encuentre realizando actividades propias de la etapa del análisis y también de la del diseño. La etapa del diseño se identifica porque en ella se definen el diseño de la arquitectura, la manera en como se relacionan los objetos a través de mensajes, la descripción de las clases y sus asociaciones.

Los esquemas obtenidos en la fase de diseño sirven de entrada al proceso de generación de código, el cual se lleva a cabo dentro de la fase de construcción. Durante esta etapa se definen la arquitectura del software, y por fin se hace la implantación para su posterior liberación. Hasta antes de llegar a la etapa de liberación, el desarrollo OO permite la creación de prototipos desde la fase de

análisis, lo cual proporciona la ventaja de confirmar si se va en el camino correcto de la elaboración de un sistema que realmente satisfaga los requerimientos.

Características del ciclo de vida OO

Iterativo e incremental, son las dos propiedades que caracterizan al desarrollo OO. El término incremental, se refiere a agregar funcionalidad a cada una de las piezas que componen el sistema, mientras que el término iterativo se refiere al proceso de evolucionar dicha funcionalidad a partir de su revisión, tantas veces como sea necesario.

Ciclo de vida iterativo e incremental

Asimismo este ciclo reconoce que las etapas de desarrollo se pueden dar en paralelo, por lo que el cliente no está obligado a determinar todo lo que quiere desde un principio sino que en cada etapa del proyecto se convierte en un mini desarrollo. Otra de sus ventajas es la posibilidad de introducir los resultados de un ciclo anterior al iniciar el siguiente, así pues los resultados del análisis y diseño subsecuentes se perfeccionan sin cesar y aprovechan el trabajo de la implantación precedente.

Metodologías OO

Cuando se está llevando a cabo la realización de un proyecto es conveniente seguir una metodología ya que ésta indicará la forma en que se describen y elaboran los modelos. La creación de modelos tiene como finalidades el facilitar el entendimiento de una cosa antes de construirla; la comunicación entre clientes(demostraciones que imitan parte o todo el comportamiento deseado); la visualización (permitiendo enfocar ideas) y la reducción de complejidad (separando un número de cosas importantes con las cuales trabajar al mismo tiempo)

Las metodologías OO de sistemas consisten en la construcción de modelos de un dominio de aplicación en los que se determina que clases y objetos existentes, la estructura y comportamiento que poseen, y las relaciones entre ellos. Posterior a esto se agregan detalles para su implementación.

Existen diversas metodologías, una de ellas es OMT[2], creada por James Rumbagh, la cual propone la creación de tres modelos. El modelo de objetos, representa la estructura estática, es decir, los datos de un sistema. El modelo dinámico, representa los aspectos del comportamiento del sistema. Por último el modelo funcional representa aspectos de transformación, es decir, las funciones del sistema. Cada uno de los tres modelos evoluciona durante el ciclo de desarrollo. En el análisis se construye un modelo del dominio de la aplicación(modelo de objetos) sin importar su implementación futura. En la etapa del diseño, se agregan ya componentes de software a partir del

[2] Object Modeling Tenique

dominio de la solución al modelo. Y en la etapa de la implementación se codifican dichos componentes, tomando en cuenta tanto el dominio de la aplicación como el dominio de la solución.

Otra metodología es OOSE[3], creada por Ivar Jacobson, propone la utilización de modelos de casos de uso, los cuales describen la completa funcionalidad del sistema identificando la manera en que los elementos que se encuentran fuera del sistema, interactúan con él. Este modelo es la base en las fases del análisis, construcción y pruebas. El objetivo en el análisis de los modelos de casos de uso, es el entendimiento del sistema de acuerdo con sus requerimientos funcionales, se encuentran los objetos, se organizan y sus relaciones son descritas. Las operaciones de los objetos son descritas durante el análisis, también. En la construcción se propone el diseño e implementación del código. Y durante las pruebas, se verifica que el sistema funcione de acorde con sus especificaciones.

Los clientes se benefician de los casos de uso ya que describen, en un lenguaje natural, todos los escenarios dentro de un sistema, por su parte los desarrolladores se benefician de los casos de uso porque les ayudan a identificar los objetos del sistema. OOSE combina tres diferentes técnicas las cuales han sido usadas por largo tiempo. La primera técnica es la programación orientado a objetos, la cual fue desarrollada en los 60's y de la cual tomó principalmente conceptos como encapsulasión, herencia y relaciones entre clases e instancias. La segunda técnica es el modelado conceptual, que es usado para la creación de distintos modelos del sistema u organización, para entender su comportamiento y arquitectura. Y por último la técnica de "block design" cuyo origen es el diseño del hardware en el área de telecomunicaciones. Esta técnica modela un numero determinado de módulos teniendo cada uno su propia funcionalidad, y estos son conectados a través de interfaces bien definidas.

La metodología de Grady Booch abarca las fases de análisis y diseño dentro del ciclo de vida OO. Define una gran cantidad de símbolos para mostrar en un modelo casi cualquier decisión del diseño, propone también el uso de cuatro diagramas básicos(diagramas de clases, diagramas de objetos, diagramas de módulos y diagramas de procesos) y dos diagramas suplementarios (diagramas de transición de estados y diagramas de interacción)

En el análisis se incluirán diagramas de objetos(para expresar el comportamiento del sistema), diagramas de clases (para representar papeles y responsabilidades de los agentes que proporcionan el comportamiento al sistema) y diagramas de transición de estados (para representar el comportamiento de los agentes en relación al orden que siguen los eventos) Durante el diseño, incluyendo la arquitectura e implantación del sistema, contendrá diagramas de clases, diagramas de objetos, diagramas de módulos y diagramas de procesos, con sus correspondientes vistas.

La importancia de mencionar las tres anteriores metodologías radica en que son la base de UML[4].

UML Lenguaje unificado de modelado

Es un sistema que utiliza una notación y es destinado al modelado de sistemas que utilizan conceptos orientados a objetos.

Breve historia

Los lenguajes de modelado OO aparecieron entre las décadas de los años 70 y los 80 y se encontraban enfrentándose con una nueva generación de lenguajes de programación OO y con aplicaciones cuya complejidad iba en aumento. Se empezó a experimentar con alternativas que se

[3] Objetc Oriented Software Engineering
[4] Unified Modeling Language

acercaban a lo que ahora son el análisis y diseño OO. El número de metodologías OO se incrementó de 10 a mas de 50 en este periodo. Muchos usuarios de estas metodologías tenían problemas al encontrar un lenguaje de modelado que satisficiera por completo sus necesidades y esto provocó la explosión de la llamada "guerra de los métodos".

Nuevas metodologías empezaron a aparecer, entre las más notables mencionadas anteriormente, OOSE, OMT y la metodología de Booch. Otros métodos también importantes, Fusion, Shlaer-Mellor y Coad-Yourdon. Cada uno era un método completo pero con sus respectivas limitaciones y ventajas.

A mediados de la década de los años 90, una lluvia de ideas inicio cuando Booch, Rumbagh y Jacobson comenzaron a adoptar ideas de cada uno de sus métodos, este trabajo colectivo estaba comenzando a ser reconocido como el que encabezaba a los demás métodos OO, y animó a los autores a generar un lenguaje de modelado unificado, con las mejores características de cada uno de sus métodos.

UML nace en 1994, primero con la combinación del método de Booch y el OMT. Posteriormente se les unió Jacobson, con su método OOSE. En respuesta a una petición de OMG[5] (asociación para fijar estándares de la industria)para definir una lenguaje y una notación estándar del lenguaje de construcción de modelos, en 1997 propusieron a UML como candidato.
Este lenguaje ha recibido la aceptación de la industria, pues sus creadores representan métodos muy difundidos de la primera generación del análisis y diseño orientado a objetos. Y además porque UML es expresivo y flexible.

UML

Se define como un lenguaje unificado de modelado que permite especificar, visualizar, construir y documentar los artefactos de los sistemas de software.

Como lenguaje, es usado para la comunicación. Es decir, es un medio para capturar conocimiento (semántica) acerca de un asunto y expresar el conocimiento (sintaxis) en relación a ese asunto. El asunto es el sistema que se esté desarrollando.

Como unificado, se entiende la unión que se da entre los sistemas de información y las mejores practicas de ingeniería en la industria de la tecnología.

Como lenguaje de modelado, se centra en el entendimiento de un asunto por medio de la formulación de un modelo de este tema. El modelo expresa el conocimiento que se tenga en relación a ese tema.

En su tarea de especificar, comunica que es requerido de un sistema y como lo podrá realizar un sistema

En su tarea de visualizar, representa a un sistema antes de ser construido.

En su tarea de construcción, guía la realización de un sistema de un sistema similar a un "plano".

En su tarea de documentación, captura el conocimiento de un sistema durante todo su ciclo de vida.

[5] Object Managment Group

Los beneficios del UML:

- Captura procesos
- Mejora la comunicación entre desarrolladores y clientes
- Maneja la complejidad
- Permite definir una arquitectura
- Permite la reusabilidad de componentes

UML no es un lenguaje visual de programación, pero sí un lenguaje de modelado visual. Tampoco es un proceso, pero si permite realizarlos.

Es importante señalar que UML es un lenguaje para construir modelos, y no guía al desarrollador en la forma de realizar el análisis y diseño orientado a objetos ni le indica cuál proceso de desarrollo adoptar.

Conceptos UML

Una de las metas de los lenguajes de modelado es entender la arquitectura de una aplicación. UML hace uso de los siguientes esquemas para la creación de modelos fácilmente entendibles.

- *Diagramas de actividad.* Representa el dinamismo de un sistema. Muestra el flujo del control entre operación y operación. En la siguiente figura se muestra un ejemplo de estos diagramas, en donde cada círculo representa una actividad, las flechas representan la transición del control entre las actividades. Cuando dos o más operaciones entregan el control, es necesaria su sincronización, y para representar dicha sincronización UML utiliza una línea horizontal que tiene como entrada una flecha o más flechas. Si la situación requiere de una decisión, ésta será incluida en el diagrama con la forma de un diamante.

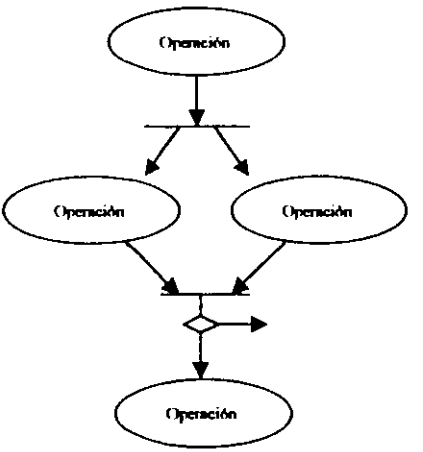


Diagrama de actividad

- *Diagramas de casos de uso.* Un caso de uso es una parte del comportamiento del sistema y muestra la relación entre un sistema y un actor, algo o alguien que interactúa o proporciona información al sistema. Los diagramas de casos de uso agrupan a los casos de uso y exhiben todas las maneras de usar al sistema y las cosas que los actores podrán hacer con él, además describen la secuencia de transacciones que se dan en un encuentro entre un actor y el sistema.

Diagrama de casos de uso

- *Diagramas de interacción* (realización de casos de uso) Se dividen en dos, los diagramas de secuencia y los de colaboración. Un escenario es la realización de un caso de uso y presenta con mayor detalle la descripción de eventos que se da entre el actor y el sistema. Para graficar los escenarios se crearon los diagramas de interacción. Los diagramas de secuencia describen las interacciones entre los objetos en un formato de muro o cerca, por su parte los diagramas de colaboración las describen en un formato de red o grafo.

Diagrama de secuencia

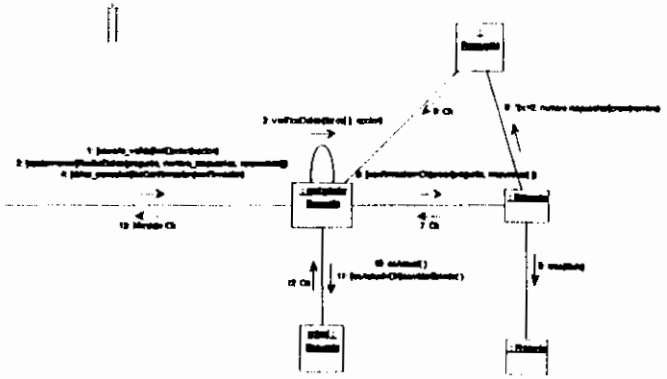


Diagrama de colaboración

- *Diagrama de clases*. Muestra tanto la estructura, dibujando las clases que conforman el sistema, como el comportamiento del sistema, incluyendo los métodos de cada clase. En este modelo se incluyen, las clases, las asociaciones entre ellas, el nombre y navegabilidad de dichas asociaciones, y en el caso de que existieran, asociaciones como la herencia y agregación. Estos dos conceptos se discutirán más adelante.

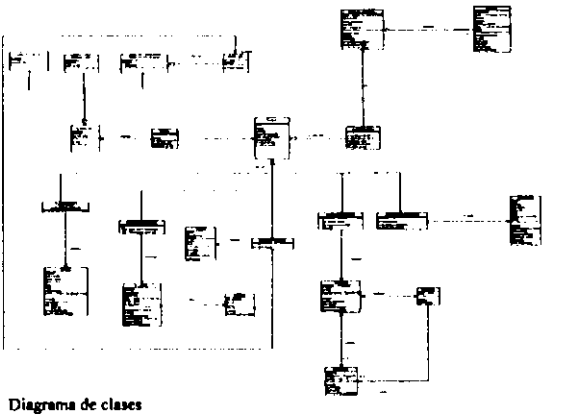
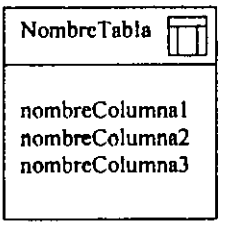


Diagrama de clases

- *Modelado de datos*. Es similar al diagrama de clases, pero en este diagrama en lugar de clases, los cuadros tienen un aspecto distinto y representan tablas que contendrán los datos.



Representación de una tabla

- *Diagrama de componentes*. Muestran la arquitectura lógica de un sistema, es decir, los elementos de software que conforman el sistema, tales como los programas fuente, programa objeto y ejecutables. Este esquema muestra también las dependencias entre ellos.

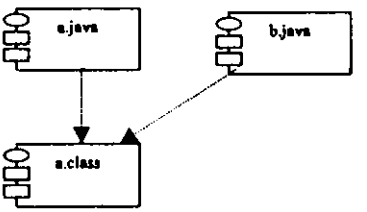


Diagrama de componentes

- *Diagrama de despliegue.* Modela la arquitectura del hardware. Cada elemento de hardware lo representa como un nodo, dibujando una caja, y conectándolos a través de líneas.

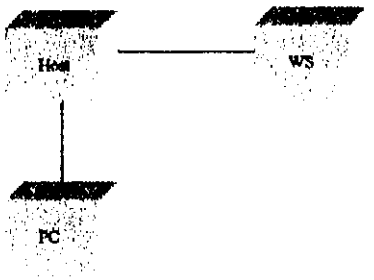


Diagrama de despliegue

Utilizando UML

Para el desarrollo de este proyecto se tomó la decisión de situar el dominio del problema con su solución lógica dentro de la perspectiva de los objetos y además utilizar una notación gráfica como medio de presentación de los resultados obtenidos, por lo que se optó por seguir una metodología orientada a objetos que consiste en tres etapas, planeación, análisis y diseño. En cada una de estas etapas, dentro de un ciclo incremental e iterativo, se propusieron ideas que durante el desarrollo fueron evolucionando y finalmente algunas se convirtieron en decisiones de diseño, las cuales se presentan en forma textual y en forma gráfica a través de diagramas construidos con UML. El uso de este lenguaje de modelado, se debe a que posee la flexibilidad necesaria para expresar con facilidad los resultados de un diseño y permite la conciliación entre las ideas del cliente y las del desarrollador.

FASE DE PLANEACIÓN

Presentación General

Este proyecto tiene por objeto diseñar un sistema para la publicación automatizada de información relacionada con proyectos de Internet2 que se desarrollan en la UNAM y otras instituciones académicas. Básicamente se está hablando de un sistema cuya aportación sea delegar la tarea de administrar la información del sitio a personas especialistas del tema y no a aquellas que poseen un conocimiento técnico en relación con la programación del web.

Cliente

La Dirección General de Servicios de Cómputo Académicos UNAM

Metas

En términos generales la meta es la automatización de los procesos de alimentación y actualización de contenidos de información a través de un sitio web cuyo funcionamiento sea por medio de páginas dinámicas.

Antecedentes del sistema

En abril de 1999 la Universidad Nacional Autónoma de México a través del entonces rector Dr. Francisco Barnés de Castro realizó un acuerdo con la presidencia de la Republica Mexicana en el que se conviene la participación de la UNAM en el desarrollo de proyectos que apliquen Internet2.

Se ideó entonces la elaboración de páginas web para la difusión de nuevos proyectos. Dicho sitio[6] es creado y actualmente existe con información referente a los proyectos de Internet2 que se están desarrollando en la UNAM y en otras instituciones académicas. Los objetivos que se persiguen al publicar este sitio en el web son el mantener informado oportunamente a todas aquellas personas interesadas en estos proyectos y ser un punto de encuentro entre desarrolladores investigadores y personas participantes o interesadas en involucrarse en los proyectos.

La problemática actual

Para el mantenimiento de este sitio colaboran únicamente dos personas un diseñador gráfico y un creador de páginas web. Este último tiene como principales tareas la de fungir como receptor de las peticiones de cambios que los líderes de proyecto hagan con relación a la información de sus propios proyectos y otra de sus tareas es la de ser el responsable de realizar dichos cambios ya sea en la redacción o presentación de la información publicada en el sitio. El diseñador por su parte tiene la responsabilidad de cuidar aquellos aspectos que tengan que ver con la imagen del sitio.

El proceso de actualización es demasiado lento y esto trae como consecuencia que la información que presenta el sitio no sea oportuna, situación que genera la necesidad de un sistema cuyo mantenimiento sea sencillo y tan oportuno como lo requiere el ambiente en el que se desenvuelve, Internet.

[6] http://www.internet2.unam.mx

Otro de los problemas que se presentan es que la comunicación entre usuarios que visitan el sitio y responsables de proyectos, queda algunas veces bloqueada ya que es necesario un intermediario que redireccione los correos electrónicos dirigidos a los especialistas del tema.

El sitio por sus propias deficiencias no satisface la demanda de los usuarios de Internet y el sitio no resulta atractivo al presentar información estática que no permite la participación interactiva de los usuarios ni la comunicación entre los mismos visitantes.

Panorama general del nuevo sistema

El nuevo sistema se compondrá de un conjunto de páginas dinámicas publicadas en el web. Y presentará dos facetas, como un medio informativo y como el editor de ese medio informativo.

Visto el sistema como un medio informativo, en su página inicial presentará información de los proyectos de Internet2 en desarrollo, una sección de noticias, un calendario de eventos y una encuesta abierta a todos los visitantes. También incluirá información general de todo el sitio indicándole al visitante las opciones que tiene al navegar por las demás páginas.

Se planea que el sitio posea foros de discusión virtuales, como una opción para que los visitantes puedan interactuar entre ellos, a través de la emisión de sus opiniones que tomarán forma de mensajes escritos y serán publicados en las paginas correspondientes a los foros de discusión. Otra opción para los visitantes del sitio, es el repositorio de documentos que el sistema maneje con el fin de poner al alcance del visitante artículos de interés.

En su faceta como editor, el sistema permitirá a través de páginas web, la publicación, modificación y eliminación de material informativo que pudiera ser publicado. Dichas acciones serán llevadas a cabo por personas que estén autorizadas. Como consecuencia a esto, también se requiere que el sistema sea capaz de identificar a cada usuario que lo utilice y saber que rol juega dentro del mismo; determinará a partir de esta identificación que faceta mostrarle a cada usuario y logrará delimitar las opciones de operación que tengan los usuarios sobre el sistema.

Funciones del sistema

Explorando a detalle lo que se requiere que el sistema haga, se ha elaborado la siguiente tabla en la que se muestran todas las actividades que se realizarán a través del uso del sistema.

Número de Referencia	Función	Categoría
1	Actualizar la información de los proyectos del sitio a través de interfaces (páginas web)	Evidente[7]
1.1	Identificar a una persona como usuario del sistema, a través de una clave y una contraseña	Evidente
1.2	Comparar la identificación y contraseña del usuario con los datos que se tienen registrados como validos	Oculta[8]
1.3	Capturar la información sobre un proyecto de Internet	Evidente

[7] Debe realizarse y el usuario debe saber que se ha realizado

[8] Debe realizarse, aunque no es visible para los usuarios. Esto se aplica a muchos servicios técnicos como guardar información en un mecanismo persistente de almacenamiento.

Numero de Referencia	Función	Categoría
1.4	Registrar y almacenar los datos correspondientes a los proyectos	Oculta
1.5	Manipular la información de los proyectos(incluye agregar eliminar o cambiar la información actual)	Evidente
1.6	Crear reportes con la descripción y avance de los proyectos	Evidente
2	Manejar foros de discusión	Evidente
2.1	Crear foros de discusión	Evidente
2.2	Dar seguimiento a los mensajes que inicien o den contestación a algún tema	Evidente
2.3	Registrar y almacenar los mensajes	Oculto
2.4	Publicar en el sitio las discusiones del foro	Evidente
2.5	Moderar los mensajes del foro de discusión	Opcional
3	Calendarizar actividades	Evidente
3.1	Capturar fecha hora titulo descripción y lugar donde se llevará a cabo alguna actividad	Evidente
3.2	Registrar y almacenar los datos correspondientes al evento	Oculta
3.3	Publicar los eventos registrados en el sitio	Evidente
3.4	Enviar correo electrónico de invitación a los eventos a los miembros del sitio	Opcional
3.5	Cancelar eventos (eliminar el registro del evento)	Evidente
4	Crear y conservar un acervo de documentos	Evidente
4.1	Transferir archivos desde la máquina cliente	Evidente
4.2	Registrar y almacenar la información referente a los documentos	Oculta
4.3	Publicar los nombres de los documentos y ponerlos a disposición de los visitantes del sitio.	Evidente
4.4	Registrar calificaciones de los documentos, otorgadas por visitantes del sitio.	Evidente
4.5	Publicar calificaciones de los documentos	Evidente
4.6	Eliminar documentos y su información.	Evidente
5	Publicación de noticias	Evidente

Número de Referencia	Función	Categoría
5.1	Capturar titulo y cuerpo de una noticia	Evidente
5.2	Registrar y almacenar los datos de la noticia	Oculta
5.3	Publicar la noticia en el sitio	Evidente
5.4	Moderación de noticias	Opcional
6.	Publicar una encuesta	Evidente
6.1	Registrar la participación en la encuesta	Evidente
6.2	Almacenar resultados de la encuesta	Oculta
6.3	Publicar resultados de la encuesta	Evidente
7	Registrar usuario	Evidente

Actores del sistema

Hasta el momento se han definido las funciones del sistema, sin mencionar quien o que se encargará de indicarle cuando realizarlas. Bajo el contexto de orientado a objetos, se dice que son los actores los que hacen esta tarea. El concepto *actor* define a una persona, organización, sistema o dispositivo que usa o que tiene interacción lógica con el sistema.

Son cinco los actores con los que el sistema tendrá que relacionarse, todos ellos representan personas. Los actores del sistema son:

Visitante	Es la persona que llega al sitio y tiene la posibilidad de navegar por las paginas y encontrar información.
Visitante activo	Es la persona que llega al sitio y ya está registrada como miembro del sitio. Esto le concede además de navegar por las paginas del sitio las opciones de participación activamente en los foros de discusión; calificar los documentos que se publiquen dentro del sitio.
Líder de proyecto	Es la persona encargada de algún proyecto de Internet2 y proporcionará al sistema la información necesaria para que sea publicada en el sitio, de igual forma será el responsable de la actualización de ésta.
Editor de información	Es quien tiene como responsabilidad alimentar el contenido del sitio con información en forma de noticias eventos documentos y foros de discusión.
Editor jefe	Es una especialización del editor de información con la característica de poder tomar las decisiones referentes a la elección de noticias y encuesta que serán publicadas.

FASE DE ANÁLISIS

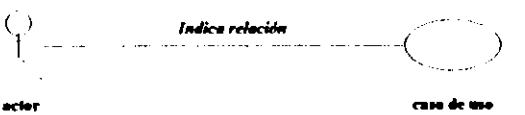
Durante la etapa del análisis se investigaron y documentaron las decisiones acerca del alcance del sistema, definiendo hasta donde abarca su comportamiento y poniendo especial énfasis en los procesos que abarcará el sistema. Además, se examinó la manera en cómo el usuario interactuaría con el sistema y se identificaron los principales conceptos del dominio, así como su significado y la relación que se da entre ellos dando lugar a uno de los diagramas más importantes dentro del análisis, el modelo conceptual.

Descripción de procesos

Para mejorar la comprensión de los requerimientos, se ha utilizado una técnica que consiste en la creación de casos de uso, que son definidos como descripciones narrativas de los procesos del dominio, es decir, describen la secuencia de eventos de un actor que utiliza el sistema para completar un proceso. Propiamente dicho, un caso de uso, no son los requerimientos ni las especificaciones en sí, sino que ejemplifican e incluyen implícitamente los requerimientos en las sucesiones de hechos que narran.

La presentación de los casos de uso, se ha dividido en tres diagramas bajo el criterio de, qué actor realiza qué funciones. De esta manera al primer diagrama se le ha nombrado información de proyectos y todos los casos de uso que se identificaron en este diagrama, son hechos por el actor "líder de proyecto"; el siguiente diagrama tiene que ver con el manejo del resto de la información que se publicará en el sitio y que de ninguna manera está involucrada directamente con la de los proyectos, es decir, es información suplementaria a los proyectos y que se presenta en forma de noticias, eventos, documentos y foros de discusión. La realización de estos casos de uso, la llevan a cabo, los actores "editor de información" y "editor jefe". En el último diagrama aparecen los usos que le dan al sistema, los actores "visitante" y "visitante activo".

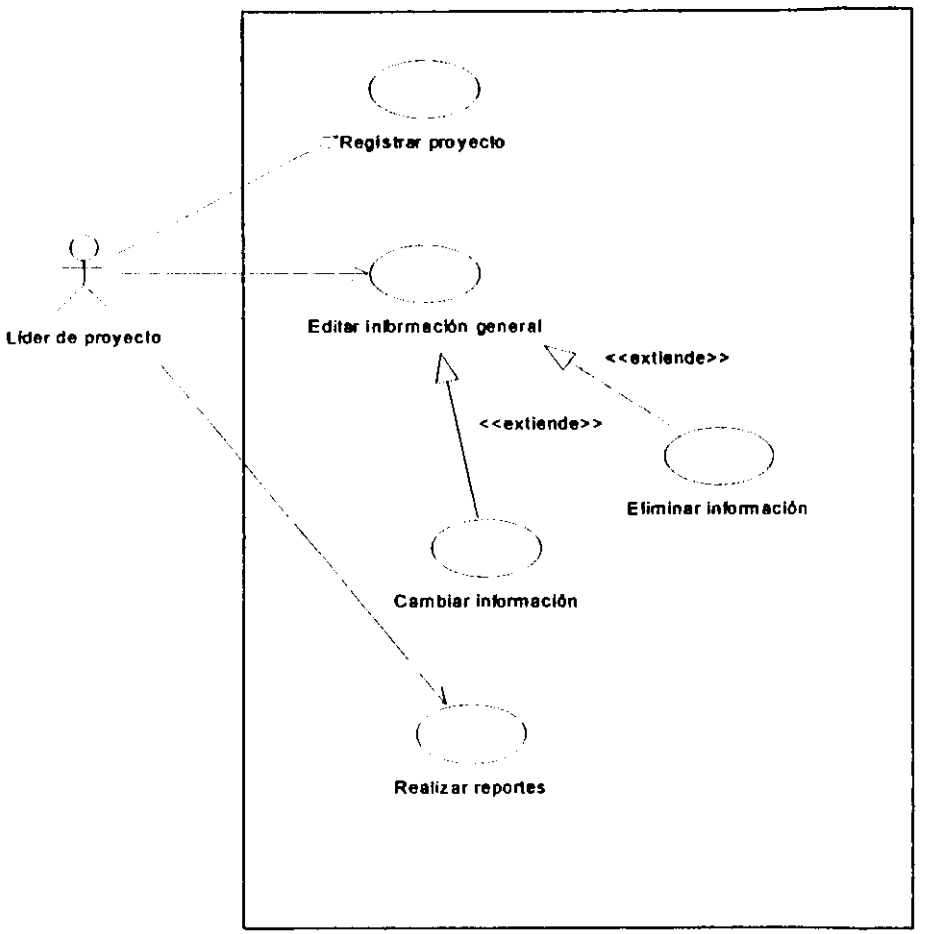
Cada diagrama encierra en un recuadro los casos de uso, lo cual indica lo que queda dentro y fuera del sistema. Explican gráficamente el conjunto de casos de uso, actores y la relación entre estos dos. Los casos de uso están representados por un ovalo, indicando en la parte inferior de estos, el nombre que les es asignado. Los actores son representados por una figura humana estilizada e igualmente, se indica su nombre en la parte inferior de dicha figura. La relación entre casos de uso y actores, se representa a través de líneas rectas; las flechas indican la dirección del flujo de información o el estímulo. Algunas de estas líneas, tienen a un costado las palabras *extiende*, que se interpreta como un tipo especial de un caso de uso, y *usa*, que se interpreta como, necesita de un caso de uso para que exista. Ambas palabras indican el nombre que reciben algunas de las relaciones que existen entre los casos de uso.



Las descripciones genéricas de los casos de uso que conforman cada diagrama, se presentan inmediatamente después de cada uno de estos. El formato que se utiliza, es muy sencillo y expone el nombre y tipo[8] del caso de uso, el o los actores que participan en él, una breve descripción, una referencia a las funciones enlistadas en el capítulo anterior, y curso de los eventos que conforman un proceso completo y que se describe a través del caso de uso. En las descripciones, se permiten interacciones en un determinado orden, pero también se permiten rumbos alternos, los cuales se mencionan al final de cada caso de uso, si es que existen.

[8] El tipo de caso de uso puede ser de tipo primario o secundario, y está dado por el impacto que tenga la realización o no del caso de uso sobre el sistema.

Diagrama de casos de uso: **Información de proyectos**



Interpretaciones del diagrama:

- El actor "líder de proyecto" sólo puede llevar a cabo los procesos que aparecen como casos de uso dentro del recuadro (*Registrar proyectos, Realizar reportes y Editar información general*)

- Los casos de uso *Eliminar información* y *Cambiar información* son las maneras específicas de *Editar información general*. Dicho de otra manera el caso de uso *Editar información general*, se presenta con más detalle al *Eliminar información* o *Cambiar información* de un proyecto.

Caso de uso: Registrar proyecto

Actor(es):	Líder de proyecto
Tipo:	Primario
Descripción:	Este caso de uso comienza cuando el líder de proyecto ya ha sido identificado por el sistema, como usuario valido y entonces el líder de proyecto proporciona los datos de su proyecto, a través de un formato, el cual, cuando es debidamente llenado y revisado, el sistema almacena en la base de datos esta información.
Referencias:	1.3, 1.4

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Comienza cuando el líder de proyectos ya ha sido identificado como usuario válido y desea registrar por primera vez un proyecto	2. Muestra un formato con espacios reservados para ser llenados por el líder de proyecto con información referente a los proyectos
3. El líder de proyectos introduce los datos que le requiere el sistema.	4.Verifica que los datos introducidos por el líder de proyecto estén completos,
	5. Solicita confirmación al usuario, para realizar la operación.
6. Confirma operación.	7. Hace el registro en la base de datos y muestra un mensaje que indica que el registro se ha llevado a cabo normalmente.

Cursos alternos

- 3. El líder no introduce datos, cambia de página y se cancela la operación.
- 4. Datos incompletos o incorrectos, mensaje de error.
- 6. El líder no confirma operación, cancela la operación, puede cambiar de página o modificar datos.

Caso de uso: Editar información

Actor(es):	Líder de proyecto
Tipo:	Primario
Descripción:	Al líder de proyecto se le presenta la información actual de su proyecto, y opciones que le permitirán manipular la información del proyecto, de tal manera que él decide si agrega, modifica o elimina algunos datos. Los cambios que se hacen, manipulan directamente la información del repositorio donde se almacenan.
Referencias:	1.5

Curso normal de los eventos(Sección principal)

Acción de los actores	Respuesta del sistema
1. Inicia cuando el líder de proyecto decide manipular la actual información de su proyecto.	2. Muestra la información actual del(os) proyecto(s) que estén a su cargo, y le presenta las opciones de: ▪ eliminar información[9] ▪ cambiar información[10]
3. El líder elige la opción y realiza la operación	4. El sistema registra los cambios en la base de datos. 5. Indica al líder de proyecto, a través de un mensaje, que dichos cambios han sido realizados sin problemas.

Curso normal de los eventos(Sección: Eliminar información)

Acción de los actores	Respuesta del sistema
1. El líder de proyecto elige la opción de eliminar información.	2. Presenta una forma en la que se indica que datos son los que se pueden eliminar
3. El líder de proyecto indica que datos desea eliminar de la información actual.	4. Mensaje de confirmación
5. Confirma eliminación	6. Borra de la base de datos, la información que el líder ha elegido para ser eliminada y devuelve un mensaje para hacerle saber al líder de proyecto que la operación ha sido completada.

[9] Véase la sección Eliminar información
[10] Véase la sección Cambiar información

Cursos alternos(Sección: Eliminar información)

- 3. El líder decide no realizar operación alguna y cambia de página.
- 5. El líder no confirma la eliminación y corrige los datos a eliminar.
- 5. El líder no confirma la eliminación, cambia de página. La operación se cancela.

Curso normal de los eventos(Sección: Cambiar información)

Acción de los actores	Respuesta del sistema
1. El líder de proyecto elige la opción cambiar información.	2. Presenta una forma en la que se indica que datos son los que se pueden cambiar
3. El líder de proyecto introduce los cambios	4. El sistema confirma los cambios
	5. El sistema valida los datos que han sido introducidos.
	6. Registra los cambios en la base de datos e indica al líder de proyecto que los cambios han sido realizados.

Cursos alternos(Sección: Cambiar información)

- 3. El líder no desea hacer cambios, cambia de página. Se cancela la operación.
- 4. El líder de proyecto no confirma los cambios y corrige los datos a modificar
- 4. El líder de proyecto no confirma los cambios, navega a otra página. Se cancela la operación
- 5. Datos incorrectos o incompletos. Mensaje de error.

Caso de uso: Realizar Reportes

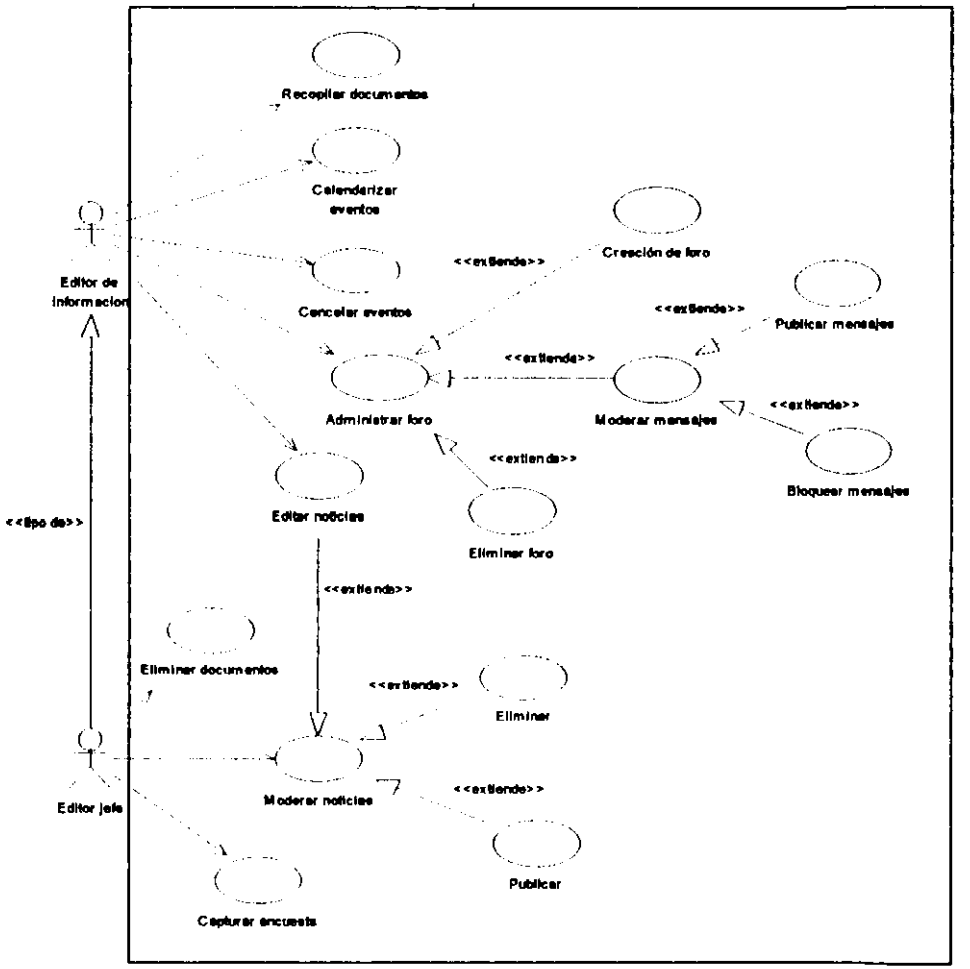
Actor(es): Líder de proyecto
Tipo: Primario
Descripción: El líder de proyecto introduce los datos propios de un reporte correspondiente a un proyecto y esta información se guarda en un repositorio de datos.
Referencias: 1.6

Acción de los actores	Respuesta del sistema
1. El líder de proyecto elige la opción de registrar reportes.	2. Presenta una forma en la que se indica que datos deberán llevar el reporte
3. El líder de proyecto introduce los datos	4. El sistema valida los datos que han sido introducidos.
	5. Mensaje de confirmación.
6. Confirma datos	7. Registra los nuevos datos en la base de datos y se lo indica al líder de proyecto.

Cursos alternos

- 3. El líder no introduce datos, cambia de página y se cancela la operación.
- 4. Datos incorrectos o incompletos, mensaje de error.
- 6. No confirma datos, corrige datos.
- 6. No confirma, cambia de página y se cancela la operación

Diagrama de casos de uso: **Alimentación de contenidos**



Interpretaciones del diagrama:

- El actor "Editor de información" podrá utilizar el sistema únicamente para *Recopilar documentos, Calendarizar eventos, Cancelar eventos, Administrar foro* y *Editar noticias.*
- El caso de uso *Administrar Foro* tiene tres maneras específicas de llevarse a cabo, creando foros, moderando mensajes o eliminando mensajes. Esto es, los casos de uso *Creación de foro, Moderar mensajes y Eliminar foro* <<extienden>> del caso de uso *Administrar Foro.*
- Los casos de uso *Publicar mensaje* y *Bloquear mensaje* son dos casos de uso que resultan más complejos que *Moderar mensajes*, ya que <<extienden>> de él.

- El actor "Editor jefe" es un <<un tipo de>> "Editor de información", lo cual significa que posee todas las características del actor "Editor jefe" más algunas particularidades.
- El actor "Editor jefe" tiene la capacidad de llevar a cabo todos los casos de uso que lleva a cabo el "Editor de información" y también los de *Eliminar documentos*, *Moderar noticias* y *Capturar encuesta*.
- Los casos de uso *Eliminar*, *Publicar y Editar noticias* son casos extendidos o particulares del caso de uso *Moderar noticias*.
- El actor "Editor de información" no podrá realizar los casos de uso *Eliminar documentos*, *Capturar encuesta*, ni los casos extendidos de *Moderar noticias* a excepción de *Editar noticias*.

Caso de uso: Recopilar documentos[11]

Actor(es):	Editor de información
Tipo:	Primario
Descripción:	El editor de información obtendrá documentos con información relacionada a Internet 2, y los pondrá a disposición de los usuarios.
Referencias:	4,4.1,4.2,4.3,4.4,4.5

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Indica que registrará un nuevo documento.	2. El sistema presenta un formulario, a través del cual, se piden datos tales como el nombre que se le dará al archivo, una breve descripción y la ruta en la que se encuentra el archivo (obligatoriamente tendrá que ubicarse físicamente en la máquina cliente)
3. El editor proporciona los datos que el sistema le pide.	4. Valida los datos
	5. Pide confirmación.
6. Confirma	7. Transfiere el archivo de la máquina cliente hacia el servidor donde residirá el archivo.
	8. Registra los datos del documento en la base de datos y devuelve al editor un mensaje de confirmación de la operación.

Cursos alternos

- 2. El editor no envía datos y cambia de página. La operación es cancelada
- 4. Datos incorrectos o incompletos. Envía mensaje de error y presenta formulario al editor, para reiniciar el proceso.
- 7. Error en la red. Cancela la operación de transferencia.

[11] El editor de información transfiere un documento que esté disponible de alguna otra fuente de información, cuidando que los derechos de autor estén debidamente autorizados.

Caso de uso: Calendarizar eventos

Actor(es): Editor de información
Tipo: Primario
Descripción: Este caso de uso inicia cuando el editor publica oportunamente aquellas actividades que estén relacionados con Internet 2, esta publicación la logra a través de indicar en un calendario la fecha de inicio, la fecha de término, y si se requiere el horario en los cuales se llevara a cabo el evento. Así como también la descripción y un titulo para el evento.
Referencias: 3,3.1,3.2,3.4

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. El editor de información indica al sistema que quiere registrar un evento en el calendario	2. Devuelve un formulario, solicitando los siguientes datos: fecha de inicio, fecha final, titulo, descripción, lugar, horario, una dirección de correo para mayores informes del evento en algunos casos será la dirección del organizador.
3. El editor ingresa todos los datos que le solicita el sistema	4. Valida los datos. 5.Envía al editor un mensaje de confirmación.
6. El editor confirma que los datos que ha ingresado son los correctos	7. Hace el registro del evento en la base de datos. Devuelve al editor un mensaje de indicación que el evento ha sido registrado.

Cursos alternos

- 3. El editor no ingresa datos y cambia de página. Se cancela la operación.
- 4. Fecha final no incluida por el editor, el sistema da por hecho que el evento tiene de duración un día.
- 4. Dirección de correo no incluida, el sistema no envía mensaje de error.
- 4. Fecha de inicio, titulo, descripción, horario ó lugar del evento vacíos, envía mensaje de error.
- 6. El editor cancela la operación.
- 7. Envía un correo electrónico, de aviso del evento, a los miembros del sitio.

Caso de uso: Cancelar eventos

Actor(es):	Editor de información
Tipo:	Secundario
Descripción:	Este caso de uso inicia cuando un editor de información que previamente a registrado un evento, decide cancelarlo
Referencias: .	3.5

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Desea cancelar un evento	2. Envía una lista con los nombres de aquellos eventos que aún están vigentes y han sido registrados por el editor de información
3. Elige el evento que desea cancelar	4. Solicita confirmación de la cancelación
4. Confirma	5. Elimina el registro del evento y envía un mensaje al editor para indicarle que la operación se ha llevado a cabo.

Cursos alternos

- 3. No elige evento alguno, cambia de página y se cancela la operación
- 4. No confirma y se cancela la operación

Caso de uso: Editar noticias

Actor(es):	Editor de información
Tipo:	Primario
Descripción:	Inicia cuando el editor recibe una noticia y decide editarla, para su publicación en la página principal
Referencias: .	5, 5.1,5.2,5.4

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. El editor de información desea editar una noticia	2. Devuelve un formulario, que permitirá a través de sus campos de textos, la captura de la información concerniente a la noticia. Solicita al editor de información anote, el encabezado de la noticia, la descripción de la noticia, la referencia y una imagen que ilustre la noticia.
3. Ingresa los datos que le solicita el sistema	4. Envía mensaje de confirmación al editor.
5. Confirma los datos	6. Valida los datos
	7. Registra la información de la noticia y los registra en la base de datos.
	8. Envía correo de aviso de la nueva noticia al editor jefe.
	9. Envía un mensaje de confirmación al editor.

Cursos alternos

- 3. No ingresa datos y cambia de página. La operación se cancela
- 5. No confirma los datos y corrige.
- 5. No confirma y cancela la operación.
- 6. Los datos no contienen archivo de imagen, no envía mensaje de error.
- 6. Los datos de encabezado, descripción o referencia están incompletos. Envía mensaje de error.

Caso de uso: Moderar noticias

Actor(es):	Editor jefe
Tipo:	Primaria
Descripción:	Este caso de uso inicia cuando el jefe editor, elige las noticias que aparecerán en la página principal, así como también desechará aquellas que resulten extemporáneas.
Referencias:	5.4

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. El editor jefe decide moderarlas noticias	2. Muestra un listado con los datos de las noticias que están sin publicar y otro listado con los datos de las noticias que aparecen actualmente en la página principal. Los datos que aparecen son: titulo de la noticia, descripción, referencia e imagen (sí la contiene) El listado de las noticias que no han sido publicadas, tiene las siguientes opciones: o Eliminar noticia o Publicar noticia
3. El editor jefe elige opción	4. Confirma la operación
5. Confirma operación	6. Realiza la operación solicitada y envía mensaje que indica que la operación ha sido efectuada.

Cursos alternos

- 3. No elige ninguna opción y cambia de página, La operación se cancela
- 5. No confirma la opción elegida. El editor jefe tiene la opción de elegir nuevamente.

Caso de uso: Administrar foro

Actor(es):	Editor de información
Tipo:	Primario
Descripción:	Inicia cuando el editor propone un tema para ser discutido y se crea un foro. Dicho foro será revisado periódicamente por el editor, para depurar aquellos mensajes. Termina cuando el foro de discusiones cerrado y ya no habrá mas mensajes que depurar.
Referencias:	2.1, 2.5

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. El editor desea administra foros	2. Muestra las siguientes opciones: • Creación de un nuevo foro [12] • Moderar mensajes [13] • Moderación de foro [14]
2. Indica operación a realizar	3. Realiza la operación
	4. Devuelve mensaje de confirmación de operación realizada.

Cursos alternos

- 2. No indica operación a realizar y cambia de página. Se cancela la operación

Curso normal de los eventos(Creación de foros)

Acción de los actores	Respuesta del sistema
1. Elija loa opción de creación de foros	2. Solicita al editor escriba el nombre, descripción del foro, e indique si el foro será moderado o abierto.
3. Escribe los datos solicitados	4. Valida datos.
	5. Solicita confirmación,
6. Confirma operación.	7. Registra el nuevo foro de discusión en la base de datos. Devuelve mensaje al editor para hacerle saber que se ha creado el foro.

[12] Véase sección creación de foros
[13] Véase sección moderar mensajes
[14] Véase sección moderación de foro

Cursos alternos

- 4. Datos incompletos o inválidos, envía mensaje de error.
- 6. No confirma, corrige datos.
- 6. No confirma. Cambia de página, se cancela la operación.

Curso normal de los eventos(Moderar mensajes)

Acción de los actores	Respuesta del sistema
1. Elige la opción de moderación de foros	2. Devuelve la lista de los foros que el editor ha creado.
3. Indica que foro quiere moderar	4. Muestra una lista de mensajes, con lo que se pueden llevar a cabo las siguientes operaciones: • Publicar (sólo en caso de que el mensaje aún no esté publicada) • Bloquear (se elimina completamente su registro)
5. Indica que mensajes aceptará y cuáles eliminará.	6. Realiza los registros correspondientes en la base de datos y devuelve al editor una mensaje de que ha efectuado las operaciones.

Cursos alternos

- 3. No indica foro y cambia de página. Se cancela la operación
- 6. El editor no indica ningún mensaje. Envía mensaje de error.

Curso normal de los eventos(Eliminar foro)

Acción de los actores	Respuesta del sistema
1. El editor de información elige la opción eliminar mensaje	2. Envía un mensaje de confirmación
3. Confirma eliminación	4. Realiza la eliminación de los registros en la base de datos. Y envía un mensaje al editor, indicándole que las eliminaciones han sido hechas.

Cursos alternos

- 3. No confirma eliminación y puede elegir de nuevo, no hay cambio de página.

Caso de uso: Capturar encuesta

Actor(es): Editor jefe
Tipo: Primaria
Descripción: El editor jefe captura tanto la pregunta como sus opciones de respuesta que formarán parte de la encuesta.
Referencias: 6

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. El editor jefe captura en una forma, los datos de la encuesta que será publicada a partir de su registro. Dichos datos son, la pregunta y las opciones de respuesta	2. Valida los datos y espera la confirmación de la operación
3. Confirma datos	4. Registra la encuesta en la base de datos
	5. Envía mensaje de operación realizada.

Cursos alternos

- 2. Datos incorrectos. Mensaje de error
- 3. No confirma datos, y corrige.
- 3. No confirma datos y cancela la operación.

Caso de uso: Eliminar documentos

Actor(es):	Editor jefe
Tipo:	Secundario
Descripción:	Este caso de uso inicia cuando el editor jefe selecciona todos aquellos documentos que desee eliminar
Referencias:	4.6

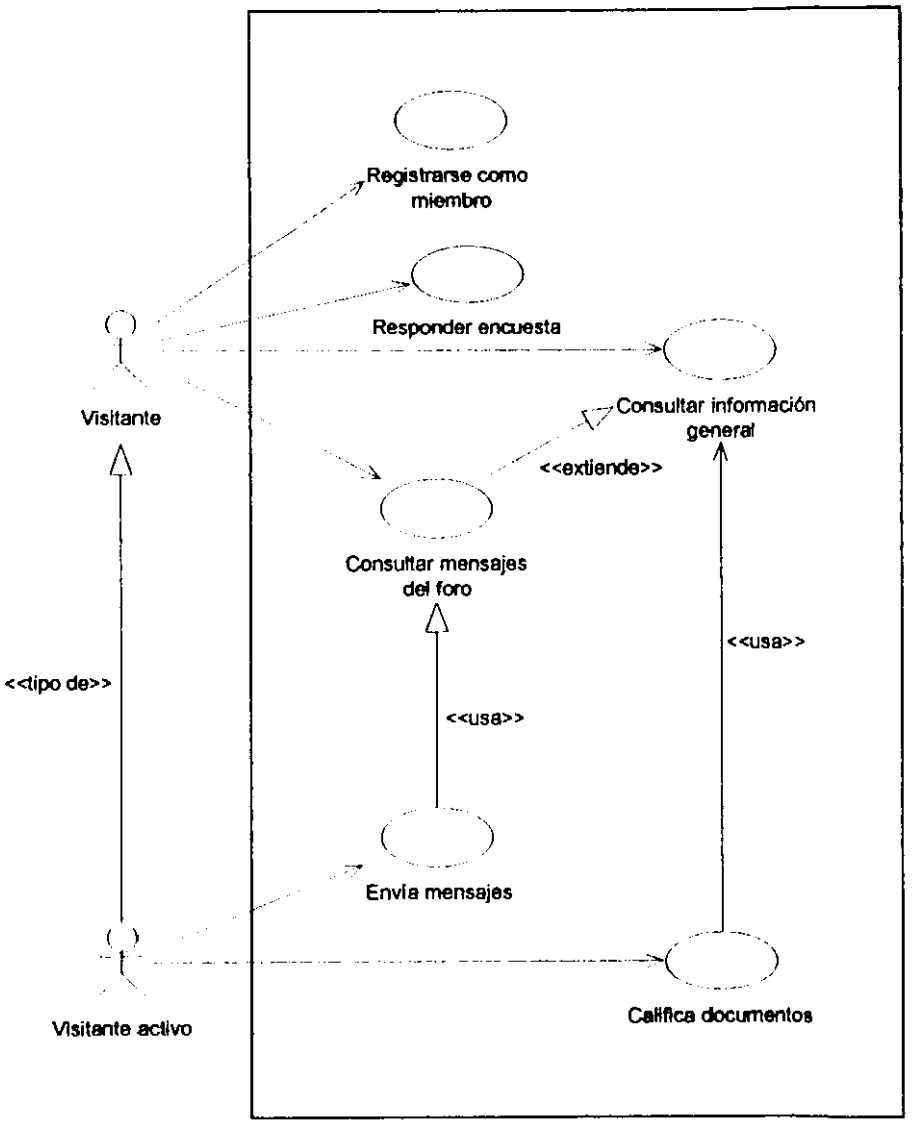
Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. El editor jefe indica al sistema que eliminara algún(os) documentos	2. Devuelve una lista de los documentos actualmente publicados, con información muy general acerca de ellos
3. Selecciona el o los documentos que eliminará	4. Envía mensaje de confirmación
5. Confirma operación	6. Realiza la operación eliminar y envía un mensaje al usuario.

Cursos alternos

- 3. No selecciona ningún documento. Mensaje de error
- 5. No confirma operación. Tiene la opción de elegir de nuevo o cambiar de página

Diagrama de casos de uso: **Visita al sitio**



Interpretaciones del diagrama:

- El actor "Visitante" es el que lleva a cabo los procesos de *Registrase como miembro, Responder encuesta, Consultar información general* y el caso de uso que <<extiende>> de éste, *Consultar mensajes del foro*

- El actor "Visitante activo" es un <<tipo de>> "Visitante", por lo que posee todas sus características y puede llevar a cabo todos los casos de uso que realice el actor "Visitante"
- Sólo el actor "Visitante activo" podrá *Enviar mensajes* y *Calificar documentos*
- El caso de uso *Envía mensajes* <<usa>> al caso de uso *Consultar mensajes*, lo cual significa que para llevarse a cabo el proceso de envío de mensajes es necesario consultar los mensajes del foro
- Únicamente si se consulta información en general se podrá calificar a los documentos. En otras palabras el caso de uso *Calificar documentos* <<usa>> al caso de uso *Consultar información general.*

Caso de uso: Registrarse como miembro

Actor(es):	Visitante
Tipo:	Secundario
Descripción:	Este caso de uso inicia cuando un visitante proporciona sus datos, y de esta manera puede participar en los foros de discusión, puede recibir noticias y avisos de eventos que se publiquen en el sitio
Referencias:	7

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. El visitante solicita registrarse como miembro del sitio	2. Solicita los datos del visitante, a través de un formato.
3. Ingresa los datos	4. Valida los datos y espera confirmación de la operación
5. Confirma datos y operación	6. Registra sus datos en la base de datos.
	7. Devuelve un mensaje de confirmación del registro.

Cursos alternos

- 3. No ingresa datos y cambia de página. Se cancela la operación.
- 4. Datos incorrectos. Envía mensaje de error
- 5. No confirma datos. Permanece la página de captura de datos.

Caso de uso: Responder encuesta

Actor(es):	Visitante
Tipo:	Secundaria
Descripción:	Inicia cuando el visitante elige una de las opciones que responda a la pregunta de la encuesta que se esté publicando en ese momento en el sitio, y termina cuando el sistema le muestra los resultados, contando ya su participación.
Referencias:	6.1,6.2,6.3

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. El visitante decide participar en la encuesta y elige una de las opciones que responden a la pregunta.	2. Valida que se haya elegido una respuesta
	3. Registra la nueva participación en la base de datos y modifica los resultados.
	4. Muestra los resultados actuales de la encuesta al visitante.

Cursos alternos

- 2. El visitante no eligió ninguna respuesta. Mensaje de error.

Caso de uso: Consultar información general

Actor(es):	Visitante
Tipo:	Primario
Descripción:	Inicia cuando un visitante llega al sitio y empieza a navegar por las páginas, obtiene información a través de la lectura de la información de los proyectos, los eventos del calendario, los foros de discusión, y los documentos.
Referencias:	2.4,3.3,4.4, 5

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. El visitante solicita visualizar alguna página del sitio.	2. Regresa la página solicitada por el visitante, con información que hasta ese momento registre la base de datos.

Caso de uso: Consultar mensajes del foro

Actor(es): Visitante
Tipo: Primario
Descripción: Este caso de uso inicia cuando el visitante del sitio ha elegido la pagina del foro de discusión, y al elegir un tema, obtendrá un listado con los mensajes que han sido enviados en relación de este tema y entonces lee el contenido de los mensajes que se han hecho hasta el momento
Referencias: 2, 2.4

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. El visitante llega a la página de discusión virtual o foros de discusión.	
2. El visitante elige un foro	3. Despliega la lista de mensajes que hasta el momento han sido enviados con relación a este foro, siguiendo un orden cronológico y poniendo sangría a aquellos mensajes que ya son contestación de otros.
4. El visitante lee el titulo de cada uno de estos mensajes y elige uno	5. Muestra el cuerpo del mensaje que ha elegido el visitante, y muestra también otros datos como son: nombre del usuario que lo envío, fecha y hora de envío.

Cursos alternos

- 2. El visitante no elige foro alguno y cambia de página
- 3. Si el foro es nuevo y no contiene mensajes, el sistema desplegará una pantalla de invitación a participar en ese tema, enviando sus opiniones.
- 4. El visitante no elige ningún mensaje y regresa a la página de todos los foros que se discuten o cambia de página.

Caso de uso: Envía mensajes

Actor(es):	Visitante activo
Tipo:	Secundario
Descripción:	Este caso de uso inicia cuando el visitante activo, escribe en una forma el titulo y cuerpo del mensaje que quiere enviar al foro de discusión. Este mensaje puede o no tener un antecesor. Cuando el sistema lo recibe lo almacena en la base de datos con la finalidad de su posterior publicación.
Referencias:	2.2,2.3,2.4

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. El visitante activo elige la opción de enviar mensaje	2. Presenta un formulario con los datos que se requieren para la contestación de un mensaje.
3. El visitante activo llena los datos que le requiere el formulario y envía dichos datos	4. Verifica que los datos no estén incompletos.
	5. Registra el mensaje en la base de datos y devuelve al usuario la confirmación del registro de su mensaje

Cursos alternos

- 3. El visitante activo no llena los datos y cambia de página. La operación se cancela.
- 4. Los datos están incompletos, devuelve un mensaje de error.

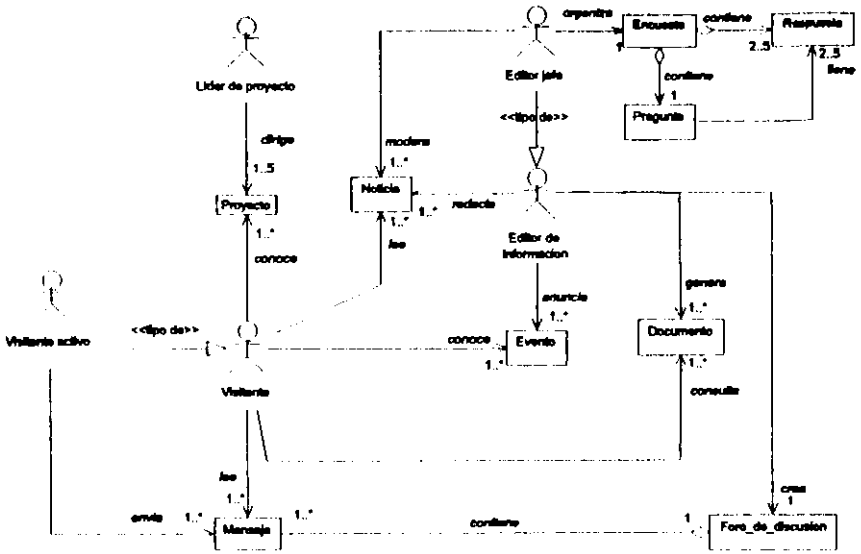
Caso de uso: Calificar documento

Actor(es):	Visitante activo
Tipo:	Secundario
Descripción:	Este caso de uso inicia cuando un visitante activo, decide dar una calificación a un documento en escala de 1 al 5. Donde el 5 representa la mejor calificación.
Referencias:	4.5, 4.6

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. El visitante activo le da una calificación a un documento determinado	2. Registra la calificación y modifica el promedio total de las calificaciones, que hasta el momento se hayan realizado.
	3. Envía el promedio actual del documento.

Modelo Conceptual



A través de los casos de uso se pueden detectar las principales entidades involucradas en los procesos que se llevarán a cabo al utilizar el sistema, y se ha descompuesto el problema en unidades llamadas conceptos, que se representan de manera estática en este diagrama. Se muestran en él, los conceptos más significativos (actores y objetos) del dominio y las asociaciones entre éstos. Cabe señalar que el modelo conceptual, se centra básicamente en el entendimiento de la terminología utilizada en los procesos y que representa cosas del mundo real, y no así componentes de software.

Los conceptos se simbolizan utilizando una figura humana estilizada en el caso de que sean actores, y se utiliza un rectángulo para representar objetos. Las relaciones se muestran por medio de líneas que comunican a los conceptos, y todas ellas poseen un nombre y una flecha que facilitan la explicación de dicha relación y la dirección en la que se da. Al terminar la flecha se muestra un número que indica el número de objetos que participan en la relación; si se habla de un rango permisible, se utilizan puntos suspensivos entre los dos números límite, y el asterisco, que en ocasiones aparece, se puede leer como, "uno o más". La relación de agregación[15] es representada por un diamante al inicio de la línea (Ver diagrama anterior)

[15] Cuando un objeto se compone de otros más.

Interpretaciones del modelo conceptual

- El "líder de proyecto" dirige al menos uno y máximo 5 proyectos.
- El "visitante" lee uno o muchos mensajes del foro
- El "visitante" consulta uno o muchos documentos
- El "visitante" conoce uno o muchos eventos del calendario
- El "visitante" conoce uno o muchos proyectos de Internet2
- El "visitante" lee una o más noticias
- Un "visitante activo" es un tipo de "visitante"
- El "visitante activo" envia uno o muchos mensajes
- El "editor de información" anuncia uno o muchos eventos en el calendario
- El "editor de información" redacta una o muchas noticias
- El "editor de información" genera uno o muchos documentos digitales
- El "editor de información" crea un foro de discusión
- Un foro de discusión contiene uno o muchos mensajes
- Un "editor jefe" es un tipo de "editor de información"
- El "editor jefe" modera una o muchas noticias
- El "editor jefe" organiza la encuesta
- La encuesta se compone de una pregunta y al menos 2 o hasta 5 respuestas
- Una pregunta tiene al menos 2 o hasta 5 respuestas

FASE DE DISEÑO

En la fase del diseño se han detallado los requerimientos descubiertos en la etapa del análisis y se ha llegado a una solución en términos de software. Para ello es necesario diseñar la arquitectura que permita definir los componentes del hardware y software necesarios para establecer la infraestructura y construcción del sistema. A lo largo de esta etapa se utilizaron diagramas de secuencia y de colaboración con el fin de especificar la manera en que los objetos logran una comunicación entre ellos a partir del envío de mensajes.

Una vez que se tuvo una descripción suficientemente detallada, se refinó el boceto del diagrama de clases y se llegó a su conclusión. Este diagrama se fue descubriendo a lo largo de este desarrollo, se le incluyeron atributos y métodos a cada clase para cumplir con su función dentro del sistema, en tanto las asociaciones de las clases, también fueron puntualizadas.

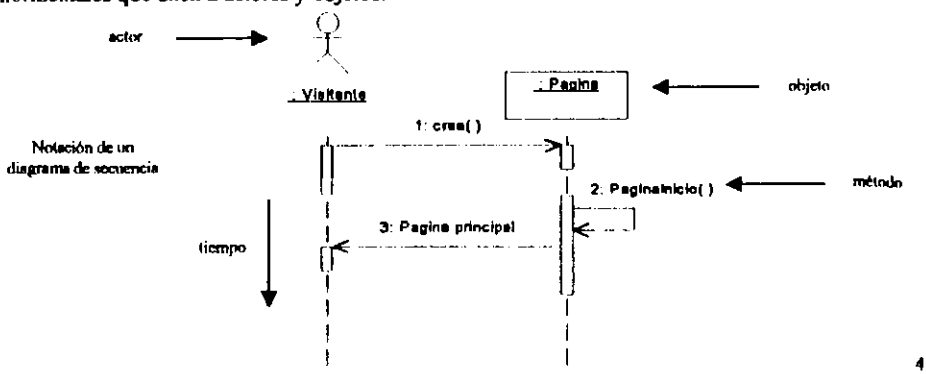
Comportamiento del Sistema

Diagramas de secuencia

Es necesario definir el comportamiento del sistema, antes de iniciar el diseño lógico de cómo funcionará la aplicación del software; es decir, se debe conocer que hace el sistema, sin importar cómo lo hace a detalle.

Durante el análisis, en los casos de uso, se ubica la interacción entre los actores y el sistema, durante esta interacción un actor genera eventos dirigidos al sistema, y solicita una operación a cambio. De manera muy similar, los diagramas de secuencia, son representaciones gráficas que muestran eventos y el orden en que son generados por actores externos, y también muestran eventos internos del sistema, ofreciendo una vista funcional y detallada del sistema, una mejor descripción de los conceptos del dominio y un modelo de cómo estos conceptos interactúan dinámicamente. En otras palabras, los diagramas de secuencia describen el comportamiento del sistema

En un diagrama de secuencia se representan los actores a través de figuras de humanos y los objetos por medio de rectángulos, cada uno porta su nombre subrayado en la parte inferior del dibujo, lo cual diferencia a los objetos de las clases que se diagraman igualmente utilizando rectángulos El tiempo avanza hacia abajo, y el ordenamiento de los eventos deberá seguir el orden indicado por los números que se anteponen a cada uno de los eventos y que se escriben sobre las líneas horizontales que unen a actores y objetos.



En el ejemplo de la página anterior se tiene a un actor "Visitante" que se comunica a través de mensajes con un objeto Página. El primer evento lo genera el "Visitante" llamando al método crea( ) del objeto Página, este objeto toma un tiempo e invoca un método propio cuyo nombre es PáginaInicio( ) para finalmente devolver como resultado al "Visitante" la página principal del sistema.

Durante este desarrollo, la presentación de los diagramas de secuencia se ha dividido en cuatro bloques de acuerdo con los casos de uso que realice cada actor. Se tomó en cuenta la misma clasificación que en los casos de uso: proyectos, contenidos de información y visita al sitio, y se añadió el bloque que describe el inicio del sistema, cada vez que un actor acceda al sitio.

Al inicio de cada bloque el lector encontrará la interpretación de algunos diagramas de cada bloque con el fin de facilitar la lectura de los siguientes diagramas.

Bloque: INICIO SISTEMA

Diagrama 1.1 Inicio del sistema

1: Un "Visitante" invoca al método crea( ) de la clase Página para que se genere una instancia de dicha clase.

2: Una vez que el objeto Página es creado, se invoca a un método propio, PaginaInicio( ), el cual tiene por objetivo desplegar en la página principal del sitio.

3: El "Visitante" recibe la página principal.

Diagrama 1.2 Reconocimiento del usuario

1: El actor "Visitante" envía un mensaje al objeto Página invocando su método ValidaUsuario( ) pasándole como parámetros el nombre y la contraseña proporcionados por el "Visitante".

2: Página invoca el método encontrar( ) del objeto Visitanteactivo.

3: El método MuestraPágina( ) se activa una vez que el método encontrar( ) ha dado un resultado y devuelve una página al "Visitante".

Diagrama 1.3 Encuesta

1: Página invoca al método crea( ) de la clase controladorEncuesta.

2: Una vez creada la instancia de la clase controladorEncuesta se invoca al método generarPágina( )

3: El objeto controladorEncuesta invoca al método esActual( ) del objeto Encuesta.

4: Si la respuesta del método esActual es verdadera, se llama al método darInformación( ).

5: El objeto Encuesta se dirige al objeto Pregunta invocando su método darTitulo( ).

6: El objeto Encuesta se dirige al objeto Respuesta invocando su método darTitulo( ).

7: Pregunta devuelve a Encuesta el nombre de la pregunta.

8: Respuestas devuelve a Encuesta el nombre de la respuesta.

9: Encuesta devuelve la encuesta a controladorEncuesta.

Diagrama 1.4 Foros de discusión

1: El objeto Página invoca la creación de una instancia de la clase controladorForo

2: Una vez hecha la instancia Página llama al método generarPágina( ) del objeto controladorPágina

3: Se llama al método enviarInformación( ) del objeto Foro_de_discusión. El asterisco indica iteración.

4: Foro_de_discusión devuelve información de él mismo.

5: controladorForo devuelve una lista de los foros de discusión.

Diagrama 1.1 Inicio del sistema

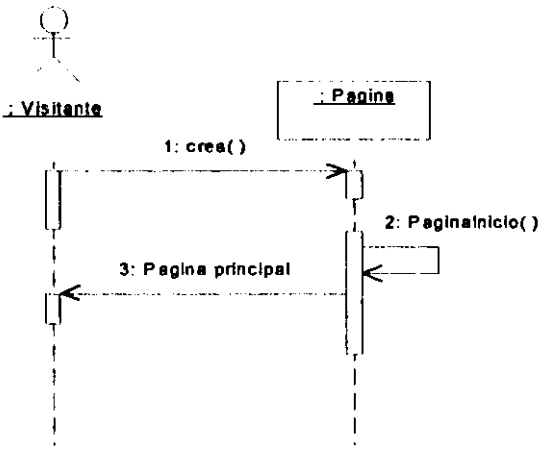


Diagrama 1.2 Reconocimiento del usuario

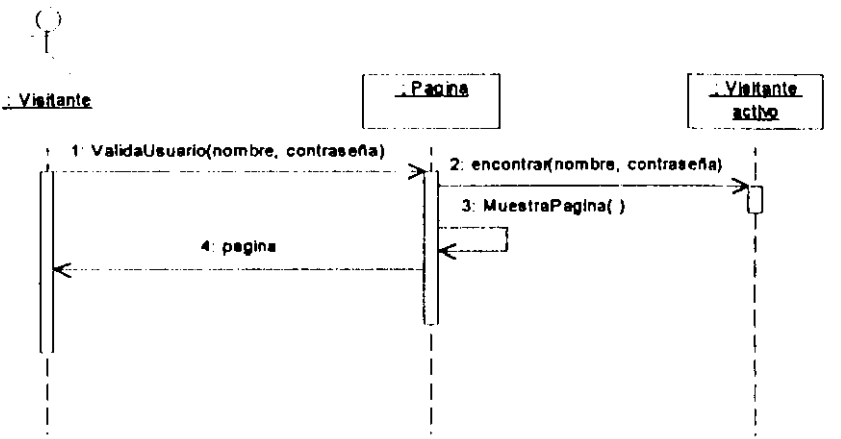


Diagrama 1.3 Encuesta

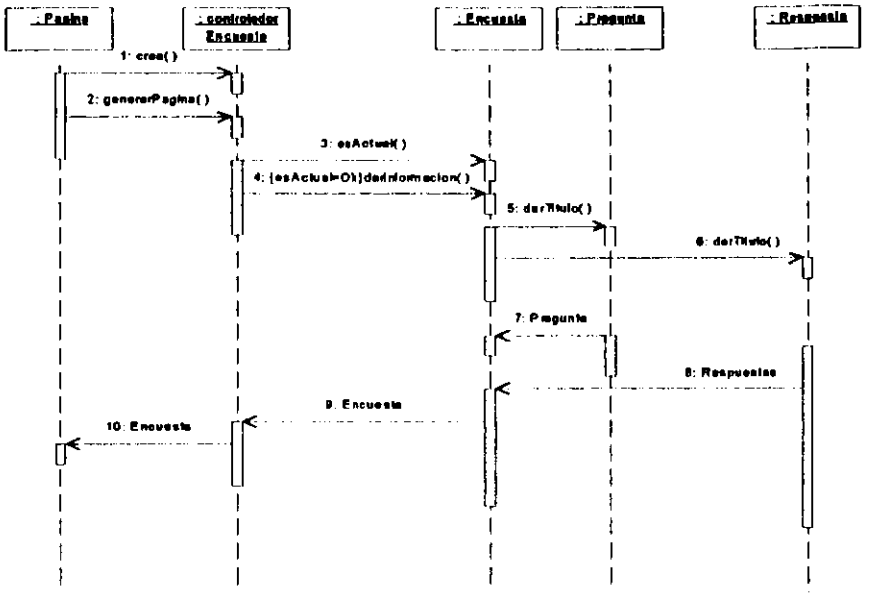


Diagrama 1.4 Foros de discusión

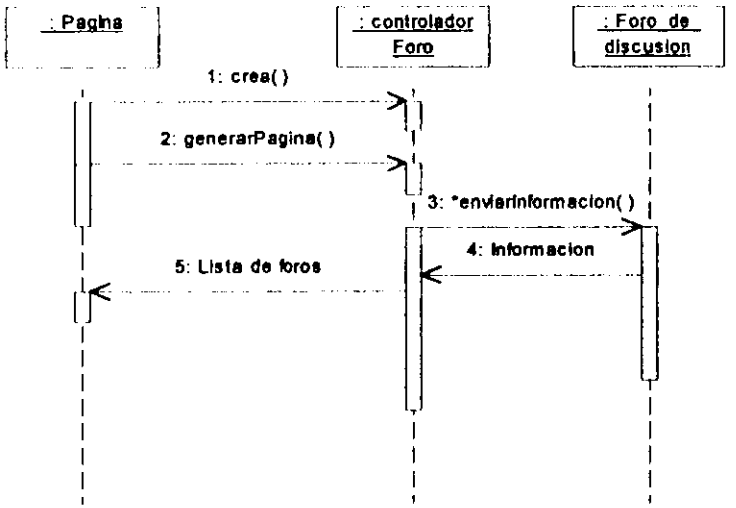


Diagrama 1.5 Proyectos

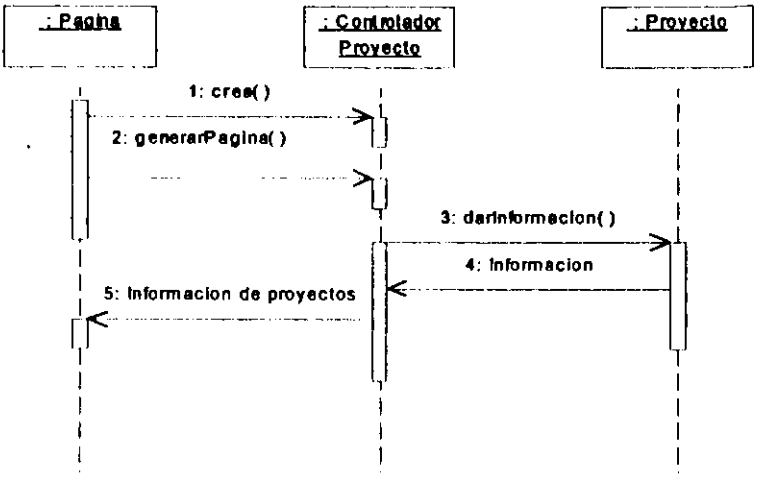


Diagrama 1.6 Noticias

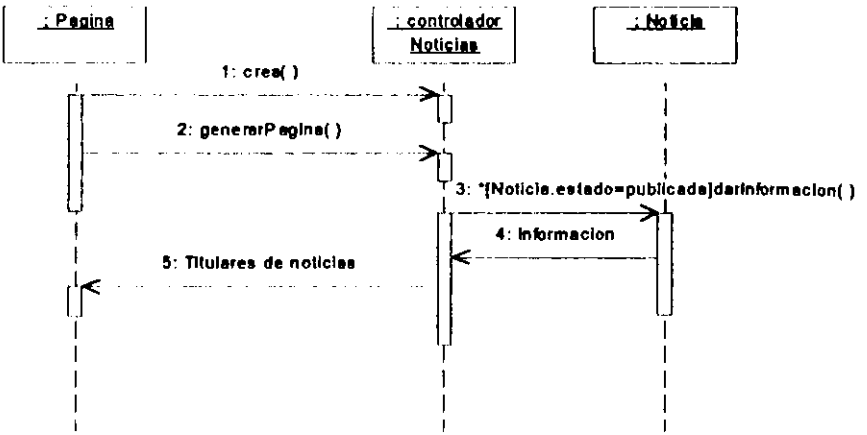


Diagrama 1.7 Documentos

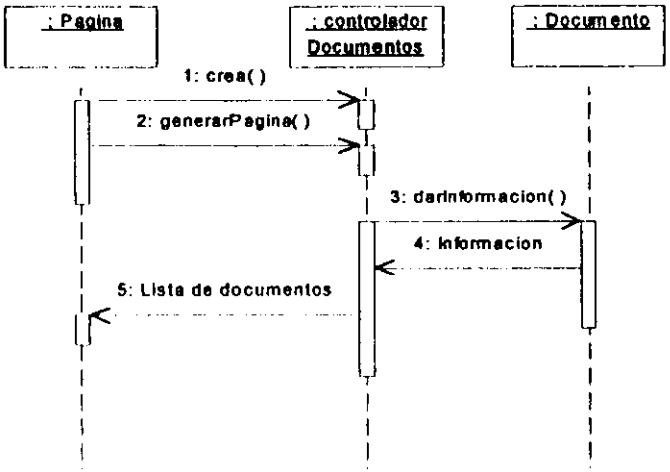
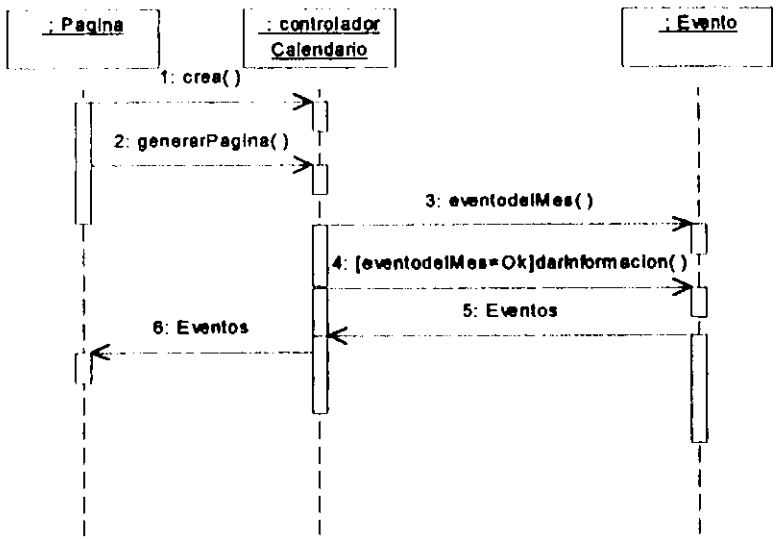


Diagrama 1.8 Calendario



Bloque: PROYECTOS

Diagrama 2.1 Inicio

1: Si el usuario que quiere acceder a esta página es un "editor de información" se invoca al método crea( ) de la clase ControladorProyecto para que genere una instancia.
2: El objeto ControladorProyecto invoca al método proyectoDe( ) del objeto Proyecto.
3: Si la respuesta del método proyectoDe( ) es verdadera se invoca al método darInformación( ).
4: El objeto Proyecto devuelve información de sí mismo.
5: El objeto ControladorProyecto devuelve al objeto Página información para que sea desplegada.

Diagrama 2.2 Registro de Proyectos

1:Si el usuario es un "Líder de proyecto" se llama al método leeOpción( ) del objeto ControladorProyecto
2: Si la opción que es enviada al método leeOpción( ) es la de registrar nuevo proyecto, se invoca al método recibeDatosproyecto( ).
3: El objeto ControladorProyeto invoca a su método verificaDatos( )
4: Si el método verificaDatos( ) devuelve un resultado verdadero, es decir que los datos enviados por "Líder de proyecto" están correctos, se invoca al método leeConfirmación( )
5: Si se confirmo la operación el objeto ControladorProyecto llama al método crear( ) de la clase Proyecto
6: La clase Proyecto confirma la creación de una instancia.
7: El objeto ControladorProyecto informa a "Líder de proyecto" que se ha registrado un nuevo proyecto.

Diagrama 2.3 Modificación de información

1: Si el usuario es un "Líder de proyecto" se llama al método leeOpción( ) del objeto ControladorProyecto
2: Si la opción elegida por el "Líder de proyecto" es modificar, ControladorProyecto invoca al método darInformación( ) del objeto Proyecto.
3: Proyecto devuelve información de sí mismo.
4: El objeto ControladorProyecto devuelve información de sus proyectos al "Líder de proyecto".
5: "Líder de proyecto" indica los datos que desea modificar a través del método recibeDatos( ) de ControladorProyecto.
6: El objeto ControladorProyecto invoca a su método verificaDatos( )
7: Si los datos son correctos solicita confirmación al "Líder de proyecto" y sabe de esta confirmación o no, a través de su método leeConfirmación( )
8: Si el "Líder de proyectos" ha confirmado su operación de modificación, se invoca al método que cambia el estado de los atributos del objeto Proyecto, modificarInformación( ).
9: Indica el resultado del método modificarInformación( )
10: Indica mensaje de aviso al "Líder de proyecto".

Diagrama 2.1 Inicio

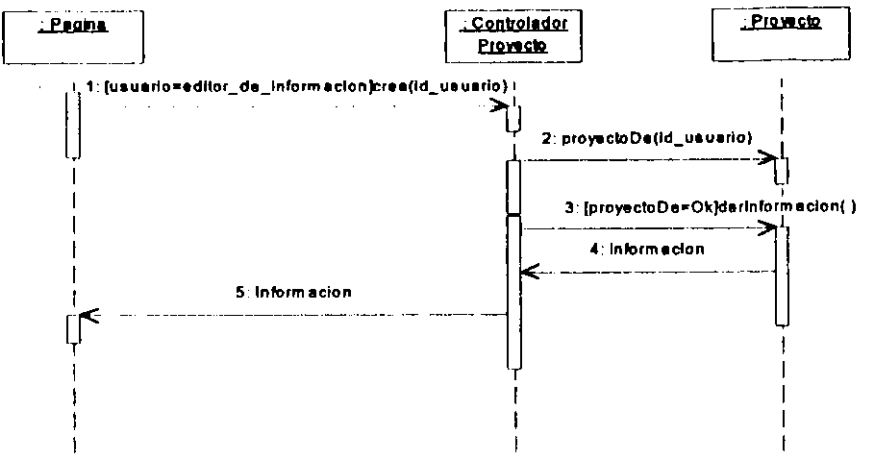


Diagrama 2.2 Registro de Proyectos

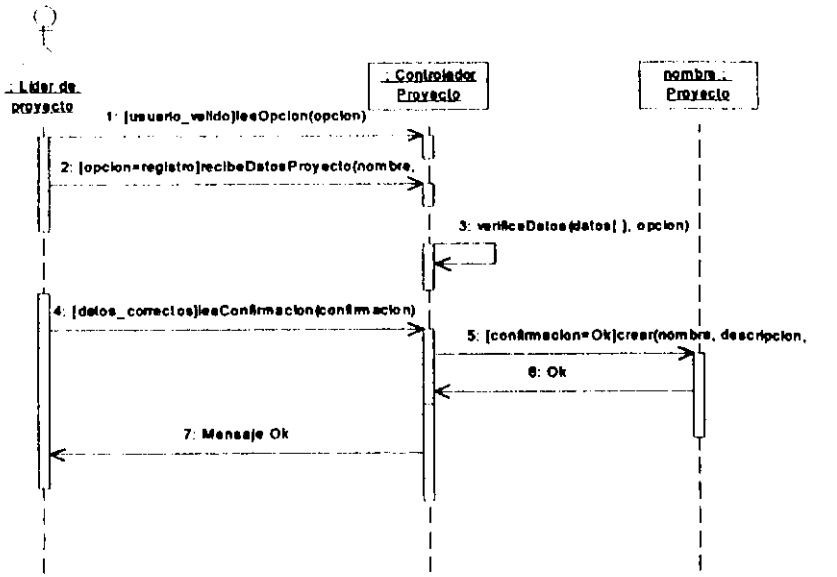


Diagrama 2.3 Modificación de información

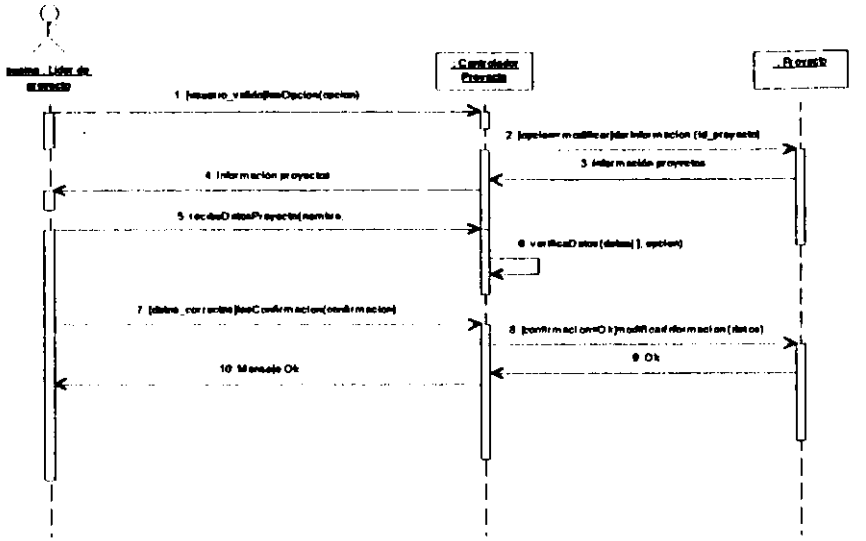


Diagrama 2.4 Eliminar información

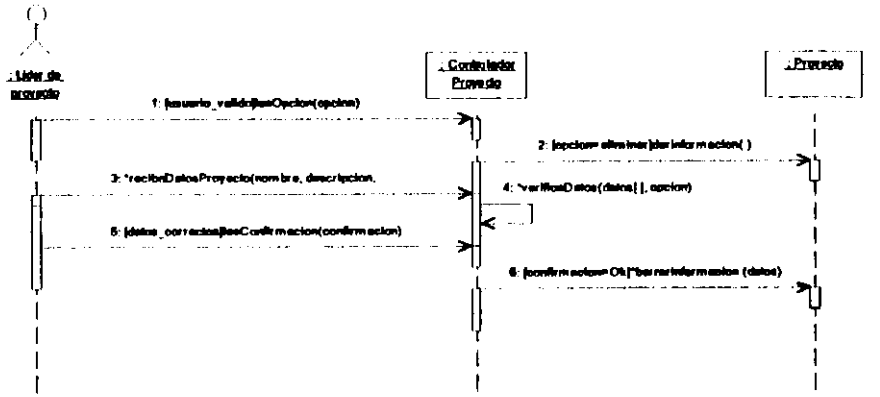
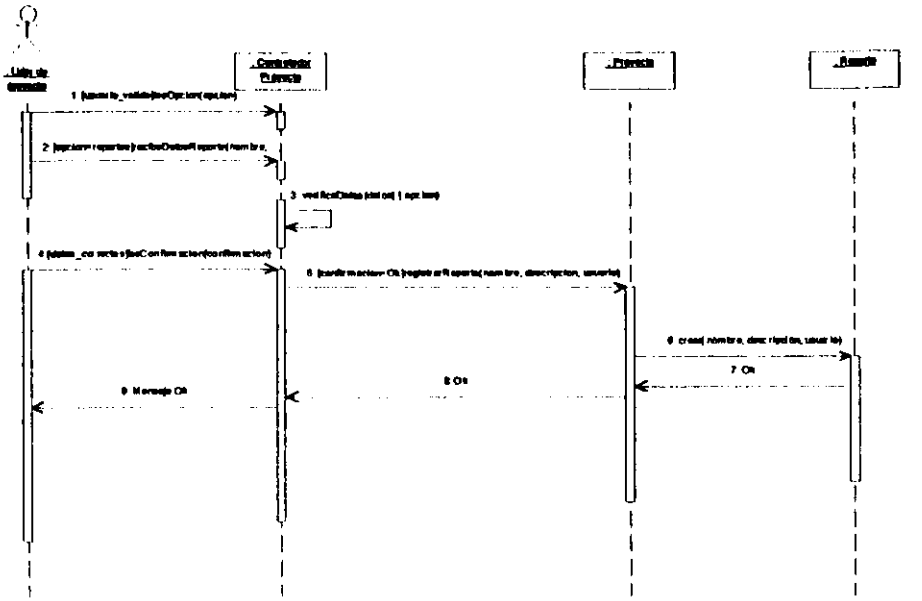


Diagrama 2.5 Registrar Reportes



Bloque: CONTENIDOS DE INFORMACIÓN

Diagrama 3.1 Inicio calendario

1: Si el usuario es "Editor de información" se invoca al método editar del objeto controladorCalendario
2: El objeto controladorCalendario llama al método eventoDe( ) del objeto Evento.
3: Si eventoDe( ) devuelve un valor verdadero se invoca al método darInformacion( ) del objeto Evento.
4: El objeto Evento devuelve información acerca del valor de sus atributos.
5: El objeto controladorCalendario devuelve información al objeto Página para que pueda ser desplegada.

Diagrama 3.7 Capturar encuesta

1: Si se trata de un "Editor Jefe" se invoca al método leeOpción( ) del objeto ControladorEncuesta
2: Si la opción es crear, se llama al método RecibeDatos( ) del objeto ControladorEncuesta
3: ControladorEncuesta verifica que los datos enviados por "Editor Jefe" sean correctos.
4: Si los datos son correctos, se solicita confirmación del "Editor Jefe".
5: Si "Editor Jefe" confirma que desea registrar una encuesta, se solicita la creación de una instancia de la clase Encuesta.
6: Se solicita la creación de un objeto Pregunta
7: Encuesta regresa aviso a ControladorEncuesta
8: Se invoca al método crear de la clase Respuesta, con el fin de crear una instancia de esta clase. Esto se realiza por lo menos dos veces y hasta el número de respuestas que posea la pregunta de la encuesta.
9: Devuelve un aviso de que las instancias han sido creadas.
10: ControladorEncuesta verifica la que sea la encuesta actualmente publicada.
11: ControladorEncuesta invoca al método cambiaEstado( ), para que el objeto actual:Encuesta cambia el estado de sus atributos y no sea más la encuesta publicada actualmente.
12: Envía mensaje de aviso.
13: Envía mensaje de aviso.

Diagrama 3.10 Eliminación de foro

1: El "Editor de información" llama al método leeOpción( ) del objeto controladorForo
2: Si la opción es eliminar foro, se llama al método leeConfirmación( ) del objeto controladorForo
3: Si se confirmó que se deseaba eliminar un foro, se invoca al método eliminar( ) del objeto Foro de discusión.
4: El objeto Foro de discusión invoca el método delForo( ) de los objetos mensaje, con el fin de conocer que mensajes pertenecen a ese foro.
5: Si el valor devuelto por mensaje es verdadero entonces, se invoca al método elimina( ) del objeto mensaje.
6: Devuelve aviso.
7: Devuelve aviso
8: Se indica a "Editor de información" el resultado de la operación.

Diagrama 3.1 Inicio calendario

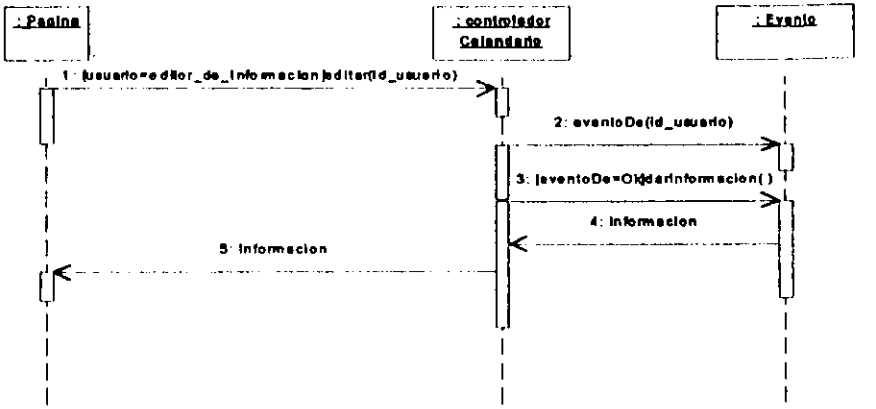


Diagrama 3.2 Calendarizar evento

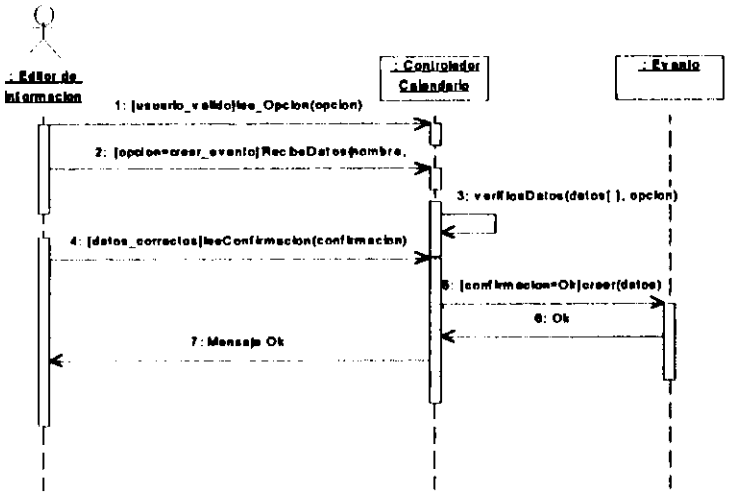


Diagrama 3.3 Cancelar eventos

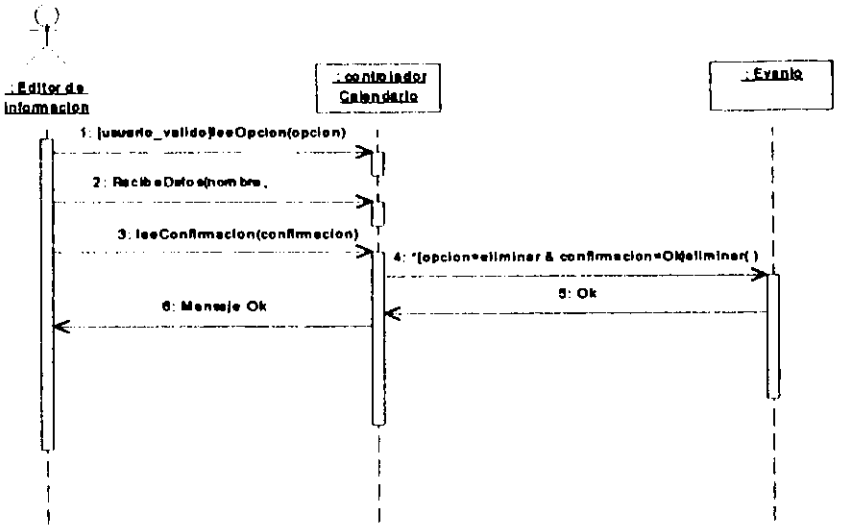


Diagrama 3.4 Inicio documentos

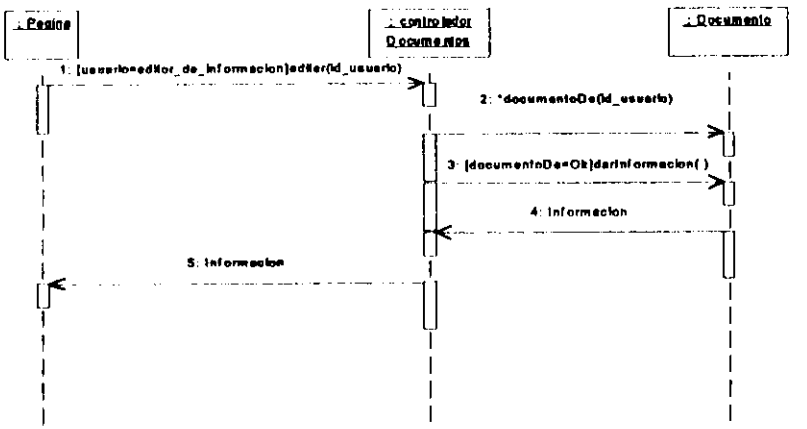


Diagrama 3.5 Recopilar documentos

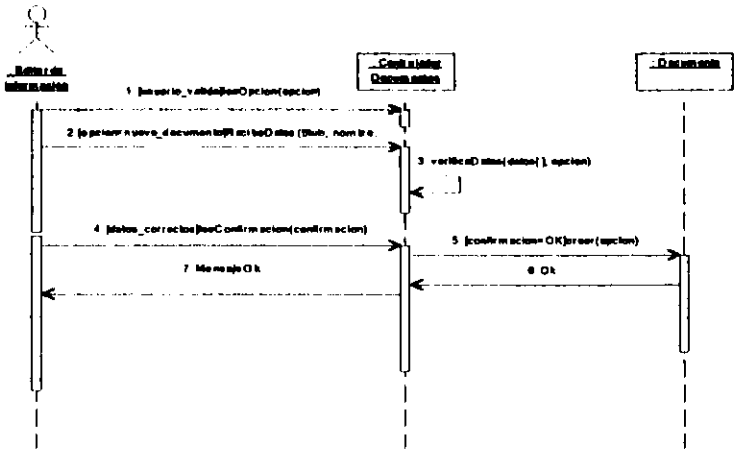


Diagrama 3.6 Eliminar documentos

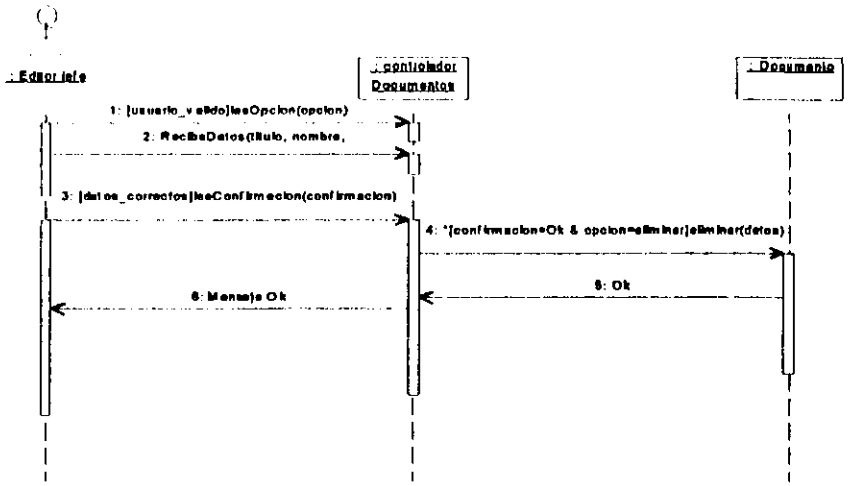


Diagrama 3.7 Captura encuesta

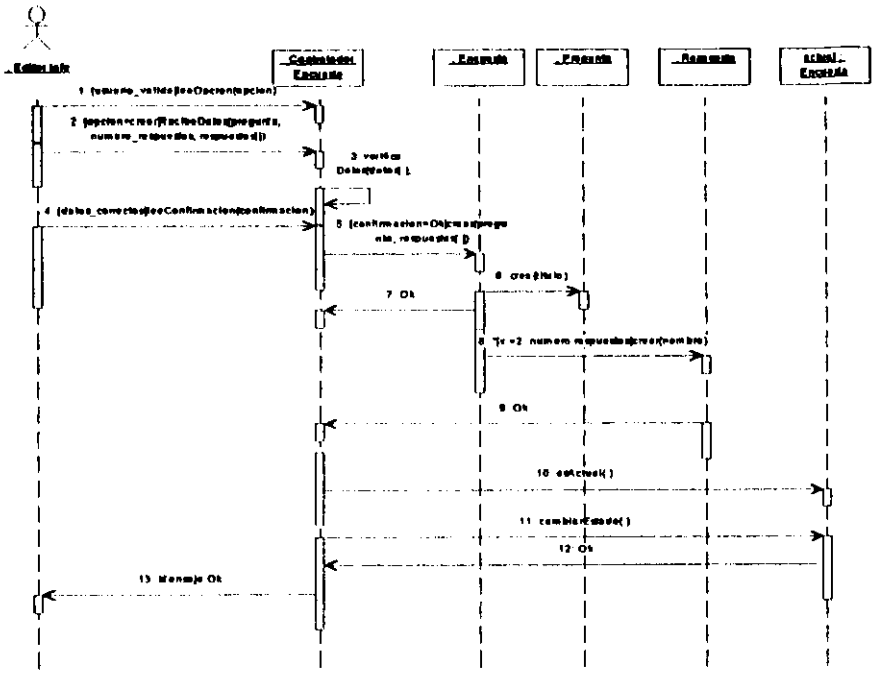


Diagrama 3.8 Inicio Foro

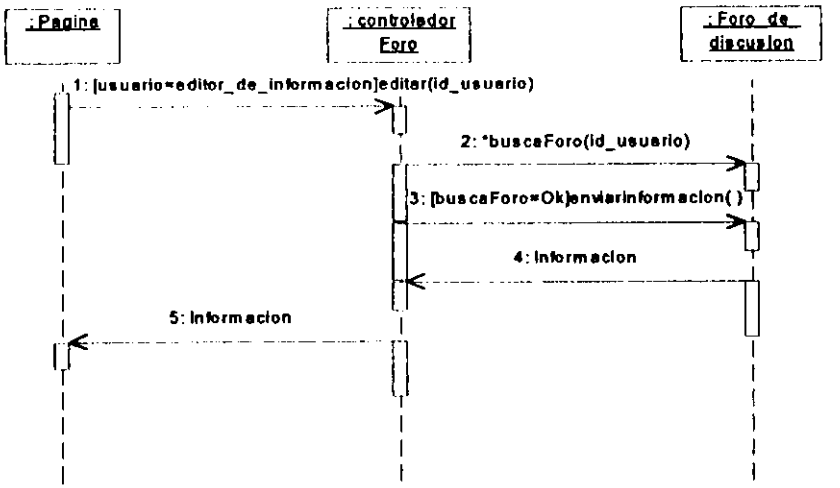


Diagrama 3.9 Creación de foro

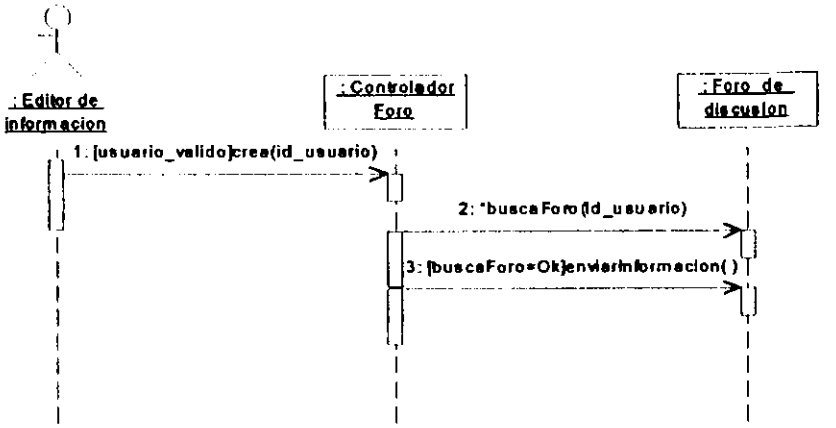


Diagrama 3.10 Eliminación de foro

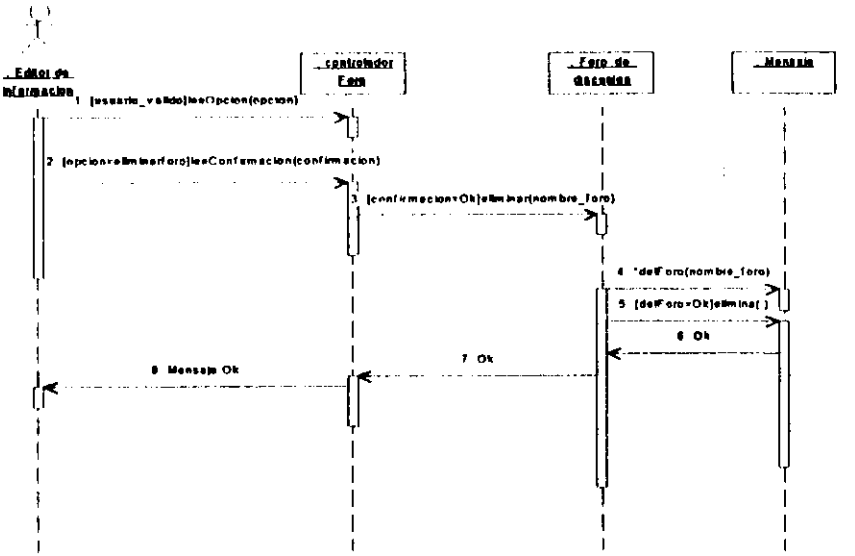


Diagrama 3.11 Publicar mensajes

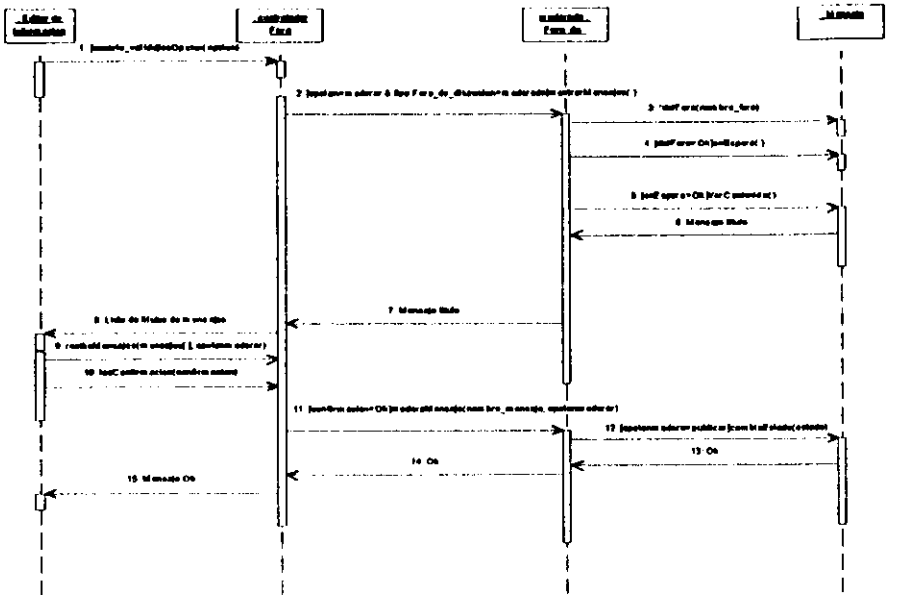


Diagrama 3.12 Bloquear mensajes

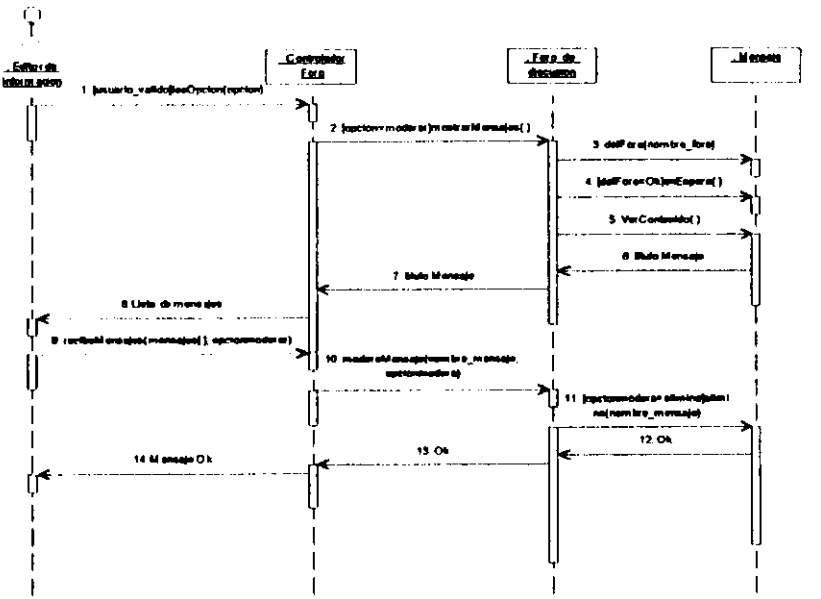


Diagrama 3.13 Inicio noticias

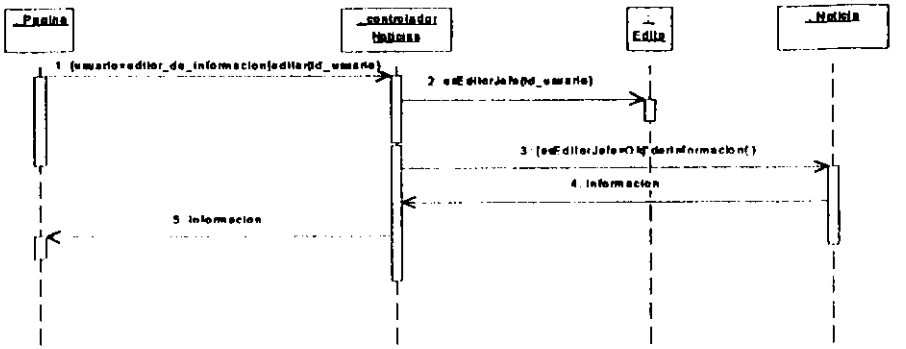


Diagrama 3.14 Editar noticias

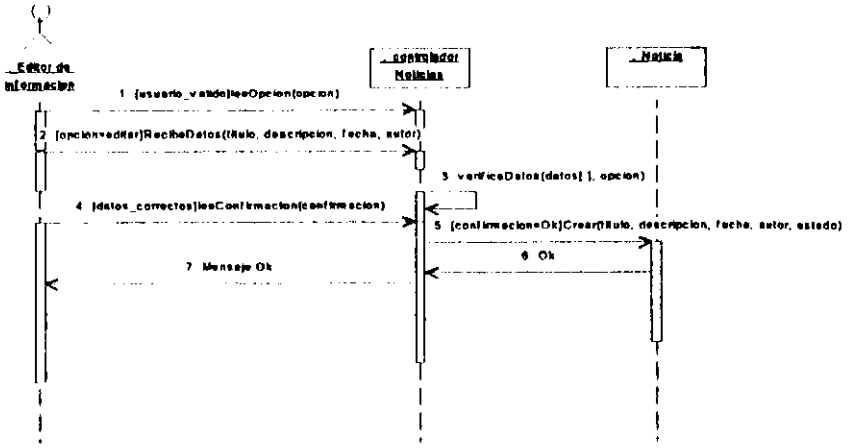


Diagrama 3.15 Publicar noticias

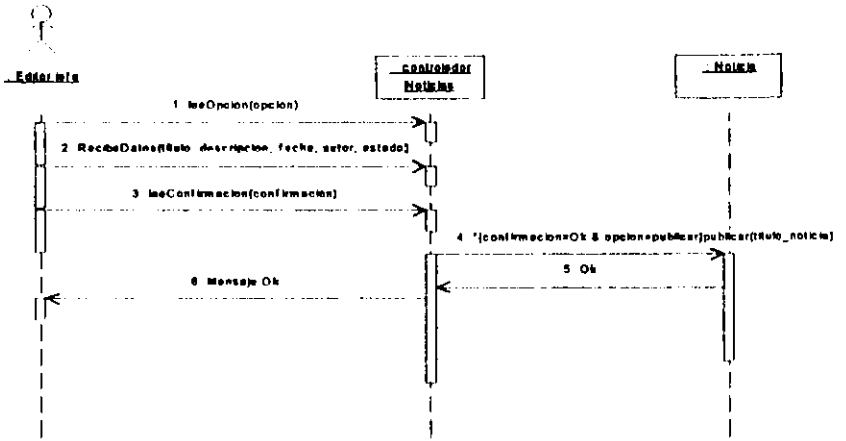
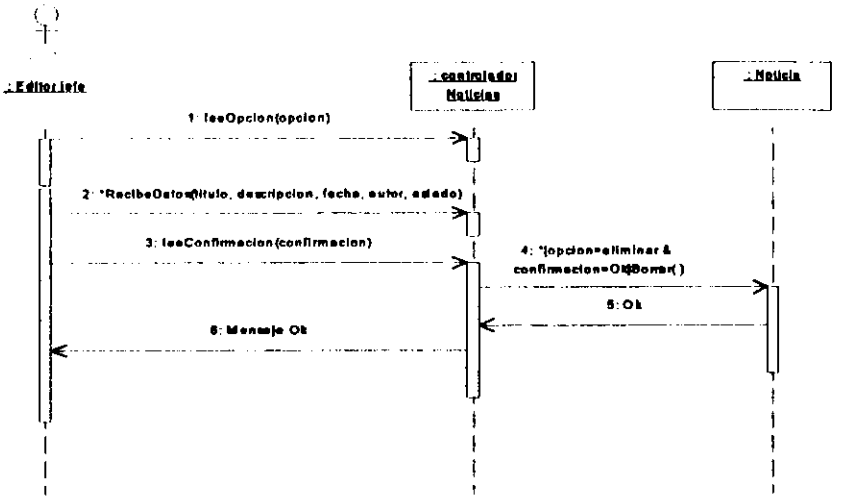


Diagrama 3.16 Eliminar noticias



Bloque: VISITAS AL SITIO

Diagrama 4.1 Registro de miembros

1: El "Visitante" invoca el método leeOpción( ) del objeto Controlador visitante
2: Si la opción es igual a registro, entonces se llama al método RecibeDatos( ) del objeto Controlador visitante
3: Controlador visitante verifica los datos enviados por el "visitante" por medio de su método verificaDatos( )
4: Si los datos son correctos, el método leeConfirmación( ) es llamado.
5: Si "visitante" confirma su opción, se llama al método crea( ) del objeto visitante_activo
6: Envía aviso.
7: Se notifica al actor "visitante" el resultado de su operación.

Diagrama 4.4 Enviar mensajes

1: Si el usuario es "visitante_activo", se invoca el método del objeto Controlador Foro.
2: Si la opción es enviar mensajes, se invoca el método recibeDatos( )
3: El objeto ControladorForo invoca a su propio método verificaDatos()
4: Se hace una instancia de la clase Mensaje
5: Envío de aviso
6: Se envía a "Visitante activo" el resultado de su operación.

Diagrama 4.6 Ver documentos

1: El "visitante" llama al método del objeto ControladorDocumentos, muestraDocumento( )
2: El objeto ControladorDocumentos invoca al método darInformación() del objeto Documento.
3: El objeto Documento devuelve información de los valores de sus atributos
4: El objeto ControladorDocumentos devuelve información desplegada al "visitante".

Diagrama 4.1Registro de miembros

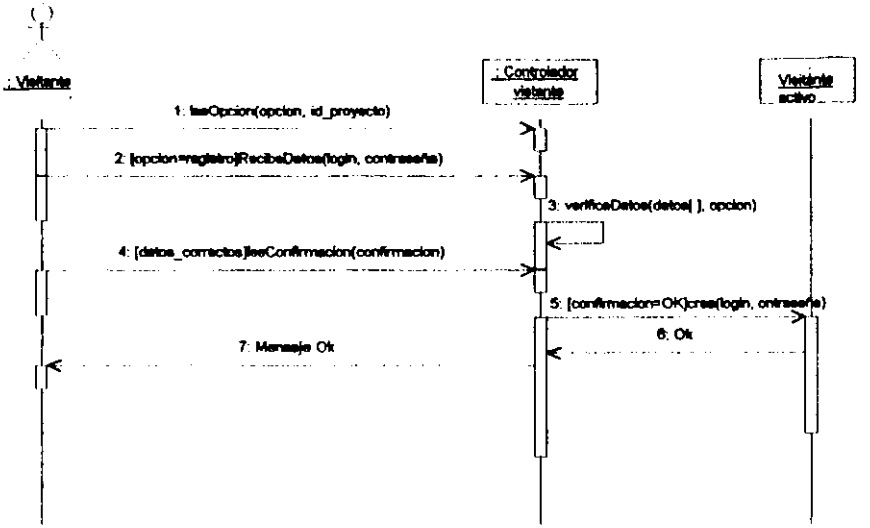


Diagrama 4.2 Responder encuesta

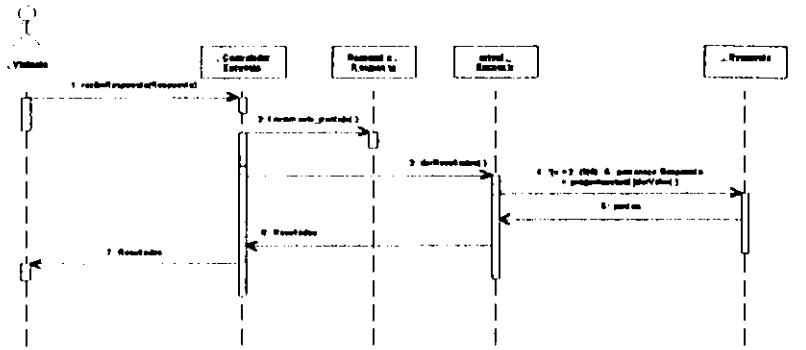


Diagrama 4.3 Consulta de mensaje

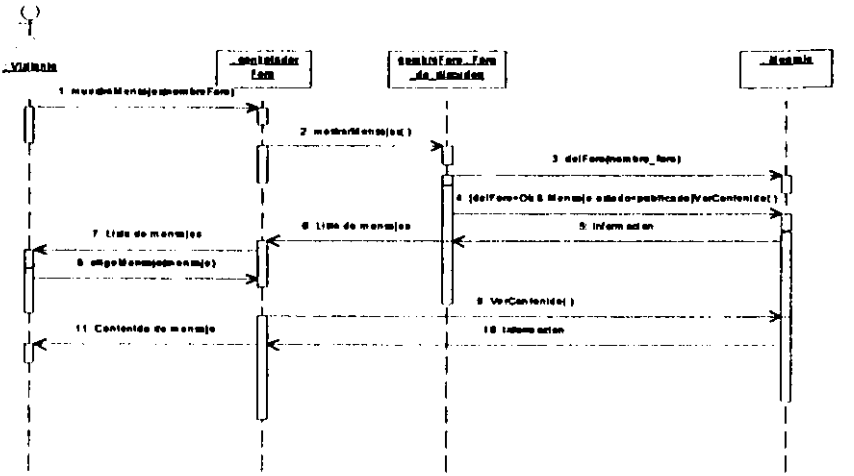


Diagrama 4.4 Enviar mensajes

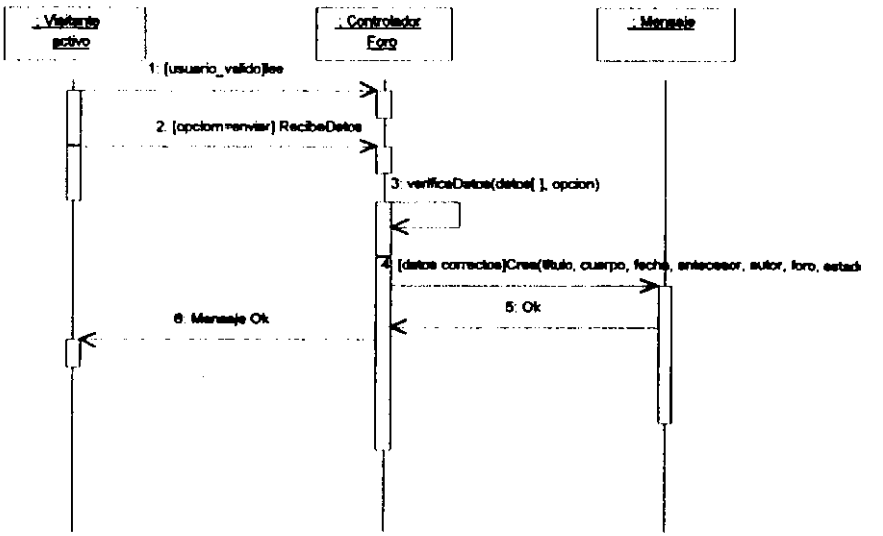


Diagrama 4.5 Ver documentos

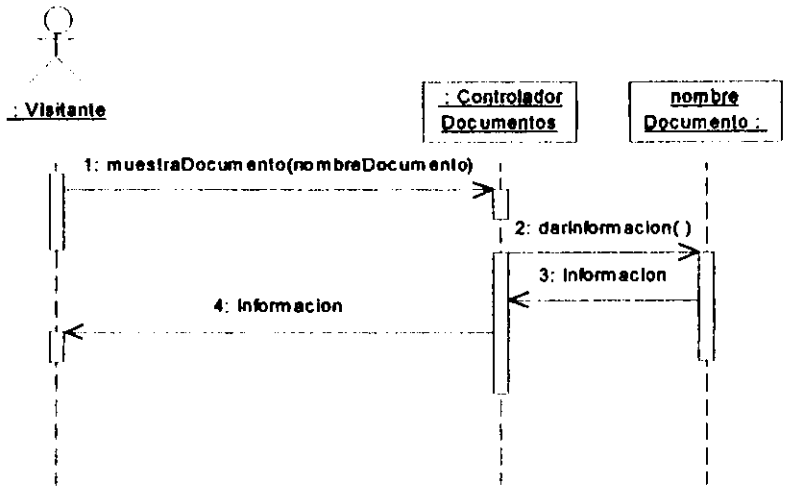
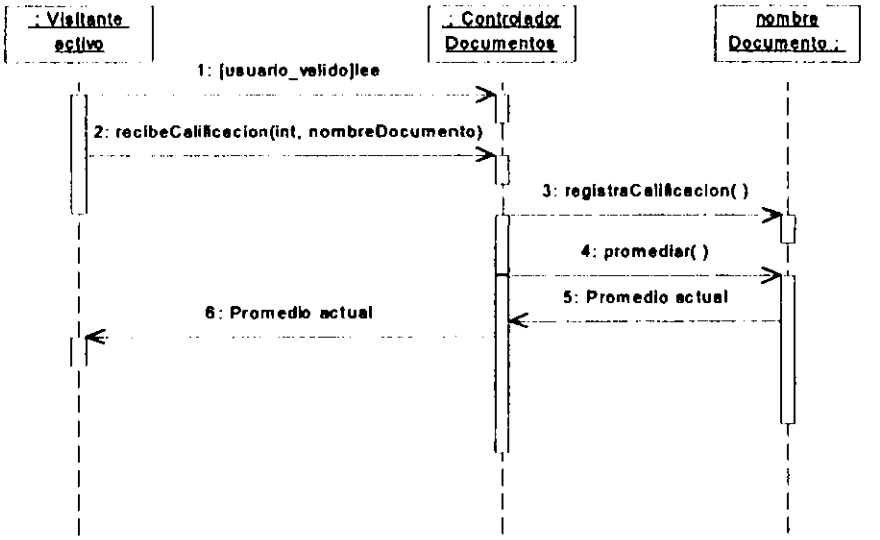


Diagrama 4.6 Calificar documento



Diagramas de colaboración

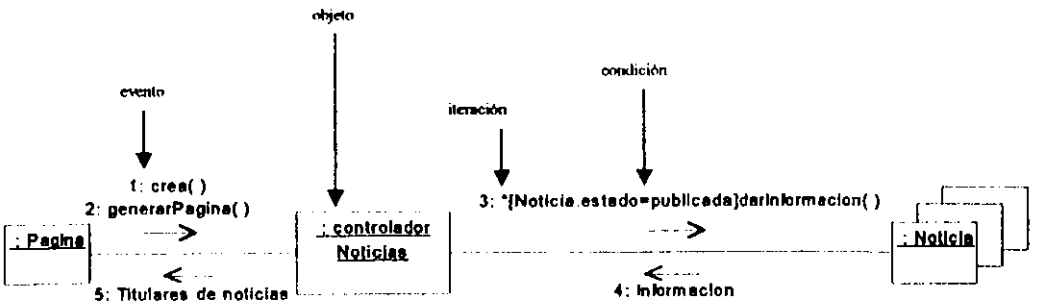
Una de las tareas más importantes dentro de la etapa del diseño es determinar las funciones de cada clase que ya hayan sido identificadas a través de los diagramas de secuencia. Para presentar el resultado de dicha tarea se ha hecho uso de la presentación de diagramas de colaboración, que explican en un formato basado en grafos, las interacciones entre objetos. Estos diagramas muestran objetos, métodos (incluidos sus parámetros), precondiciones y secuencias de métodos, necesarios para poder cumplir con la realización de los casos de uso del sistema.

La especificación de funciones de cada clase en este diseño se basó en patrones GRASP[16]. Los patrones GRASP, son principios generales que guían el desarrollo del software, permiten una acertada toma de decisiones en la asignación de responsabilidades a cada una de las clases que forman parte del sistema. Si el diseño orientado a objetos, se apega a dichos patrones, es altamente probable que, las interacciones entre los objetos del sistema cuenten con la calidad que los sistemas requieren y poder tener componentes fáciles de entender, reutilizar y de mantener.

En los diagramas de colaboración participan objetos y actores. Los objetos están representados por un rectángulo con su nombre subrayado e incluido dentro de él. Por su parte los actores, se representan por medio de una figura humana. Los eventos son generados tanto por actores como por objetos y estos eventos desencadenan llamadas a diversos métodos de los objetos.

Las relaciones entre objetos se grafican con una línea; existen situaciones en las que aparece una línea relacionando a un objeto consigo mismo,, lo que significa que un objeto ha mandado a llamar a un método propio. Cuando la ejecución de un método está condicionada, la condición se escribe dentro de paréntesis cuadrados y solo si es verdadera se ejecutará el método, si no fuera así, entonces continua con la ejecución del siguiente método que esté inmediatamente después en la numeración. Dentro de las condiciones existen los siguientes símbolos, cuyo significado es:

& que todas las condiciones sean verdaderas para que un método se lleve a cabo
| que por lo menos una condición sea verdadera para que un método se lleve a cabo
* indica iteración



Notación de los diagramas de colaboración

[16] General Responsability Asignment Software Patterns (patrones generales de software para asignar responsabilidades)

Dado que se está desarrollando este sistema dentro de un ciclo de vida iterativo, se elaboró el diseño básico de pantallas a la par de la creación de los diagramas de colaboración, por lo que, los siguientes gráficos presentan, conjuntos a los diagramas de colaboración, las pantallas fundamentales del sistema.

En la mayoría de los diagramas, el lector encontrará un objeto cuyo nombre incluye la palabra *controlador*. Uno de los patrones GRASP, es el ***patrón controlador***, que sugiere la inclusión de objetos de interfaz encargados del manejo de los eventos, los cuales se dan desde la capa de presentación, y no interfieren en la responsabilidad de las operaciones de objetos que manejan información importante del negocio o dominio y la ventaja se obtiene al aumentar el potencial de la reutilización de componentes. De esta manera se tienen a los objetos controladores y a las pantallas de despliegue formando parte de la capa de presentación del sistema y no dentro de la capa del dominio.

Es necesario señalar que para que los procesos de inicio se lleven a cabo se requiere que una persona haya tecleado su clave y contraseña, y el sistema lo haya reconocido como usuario o no del sistema[17]; como respuesta a esto, se activa el método MuestraPagina( ) del objeto Pagina y a su vez este método invoca a los métodos llamados crea( ) de los distintos objetos controladores y se originan los procesos de inicio. Por esta razón aparece el objeto Pagina como iniciador en los diagramas de inicio.

A diferencia de los demás diagramas, en los de inicio se muestra la pantalla que desplegará al terminar el proceso y que representa en cada diagrama, una parte de la página que devuelve el objeto Pagina en el proceso de *Reconocimiento del usuario*. Por lo que respecta al resto de los diagramas, se omitió la pantalla de salida, ya que en todos los casos, ésta siempre será un mensaje de notificación al usuario (éxito o fracaso en la operación solicitada), y a partir de aquí decidir si intenta de nuevo la operación o navega hacia otra página.

[17] Ver el diagrama reconocimiento del usuario

Bloque: INICIO DEL SISTEMA

Cada vez que una persona acceda al sitio, se estará invocando al método constructor[18] crear( ), que se encargará de la creación de una instancia de la clase Pagina y una vez creado, se invocará el método PaginaInicio( ) cuya función es el despliegue de la pagina principal en el browser del visitante; para lograr esto, se requiere que los procesos que se muestran en los siguientes diagramas se lleven a cabo, a excepción del proceso inicio, todos los demás se realizan debido a que el método PaginaInicio( ) los origina.

Diagrama 1.1 Inicio del sistema

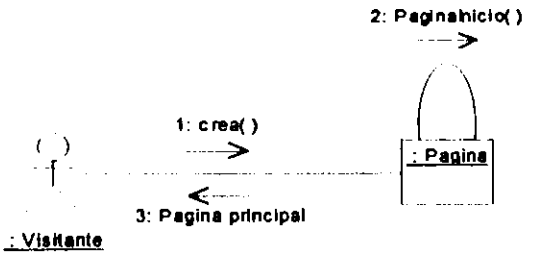


Diagrama 1.2 Encuesta

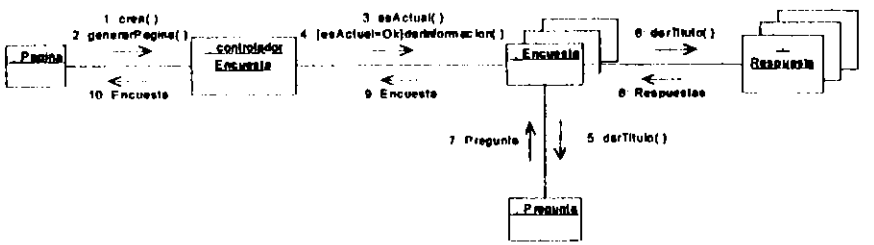
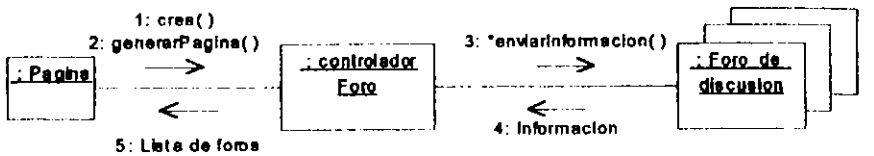


Diagrama 1.3 Foros



[18] Un método constructor es aquel que crea una instancia de una clase.

Diagrama 1.4 Proyectos

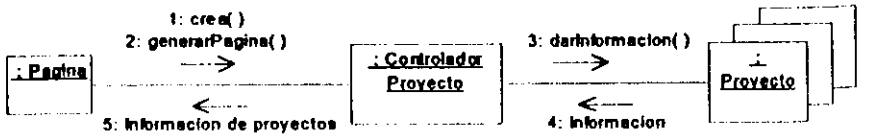


Diagrama 1.5 Noticias

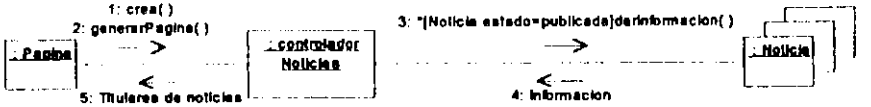


Diagrama 1.6 Documentos

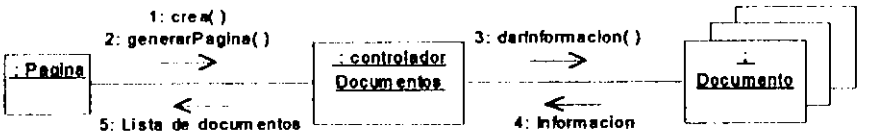


Diagrama 1.7 Calendario

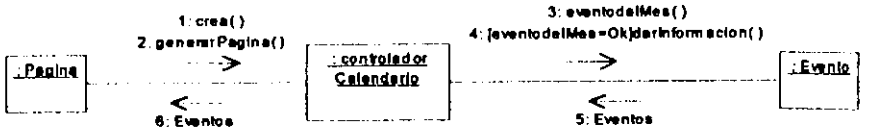
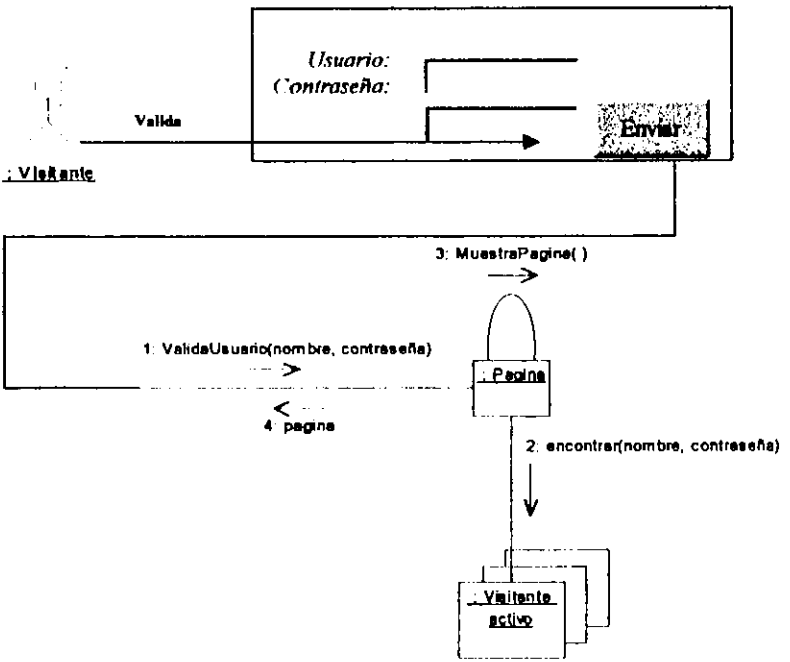


Diagrama 1.8 Reconocimiento de usuario



Bloque: **PROYECTOS**

Diagrama 2.1 Inicio

La pantalla de salida, desplegará únicamente aquellos proyectos que sean propiedad del líder de proyecto, permitiéndole realizar las operaciones de eliminación, modificación o registrar uno nuevo. Si el líder de proyecto, aún no ha registrado ninguno, simplemente aparecerá la operación de registrar uno nuevo.

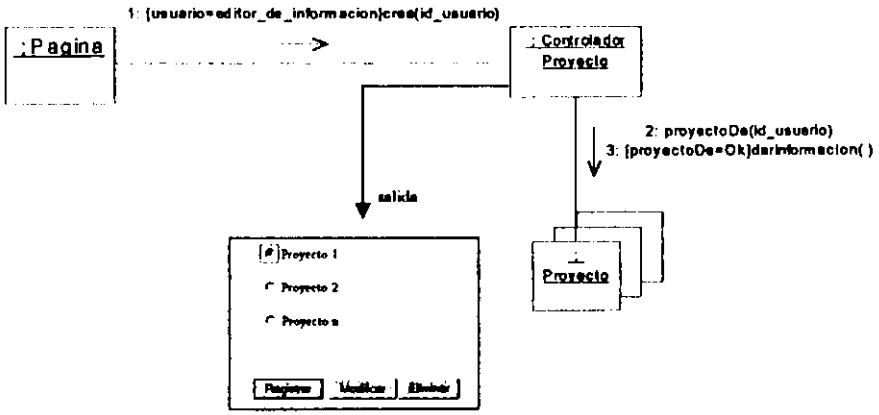


Diagrama 2.2 Registro de proyectos

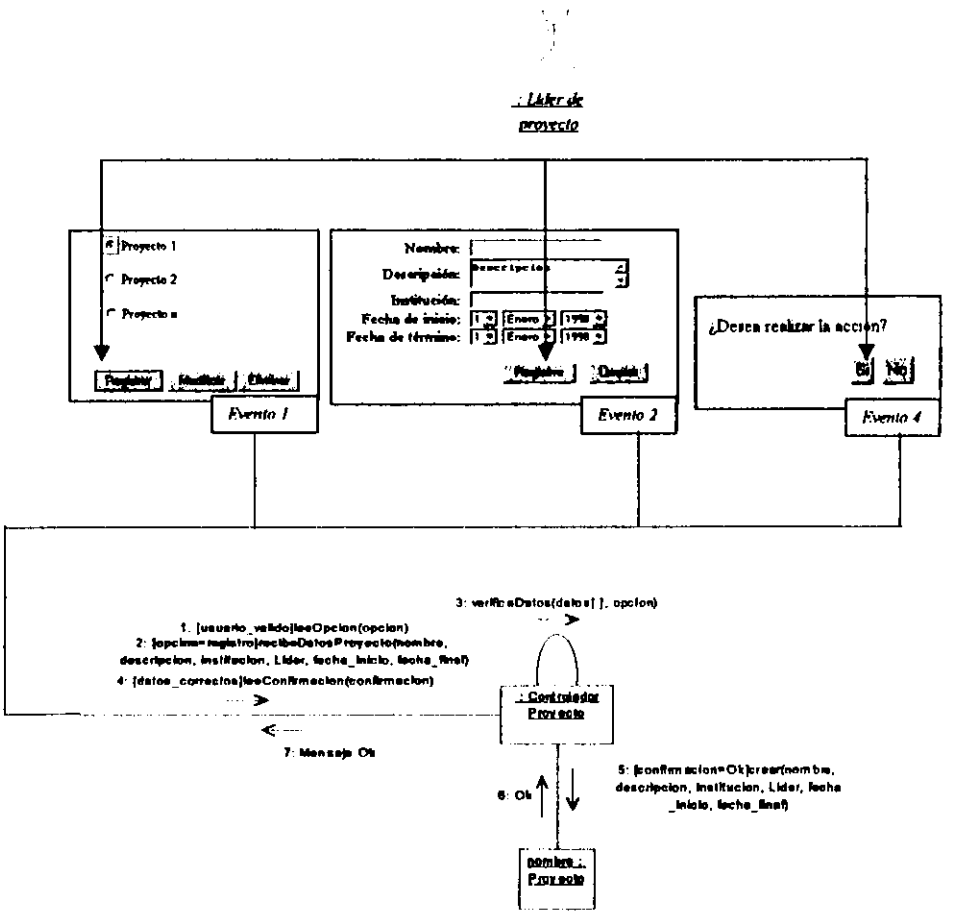


Diagrama 2.3 Modificación de proyectos

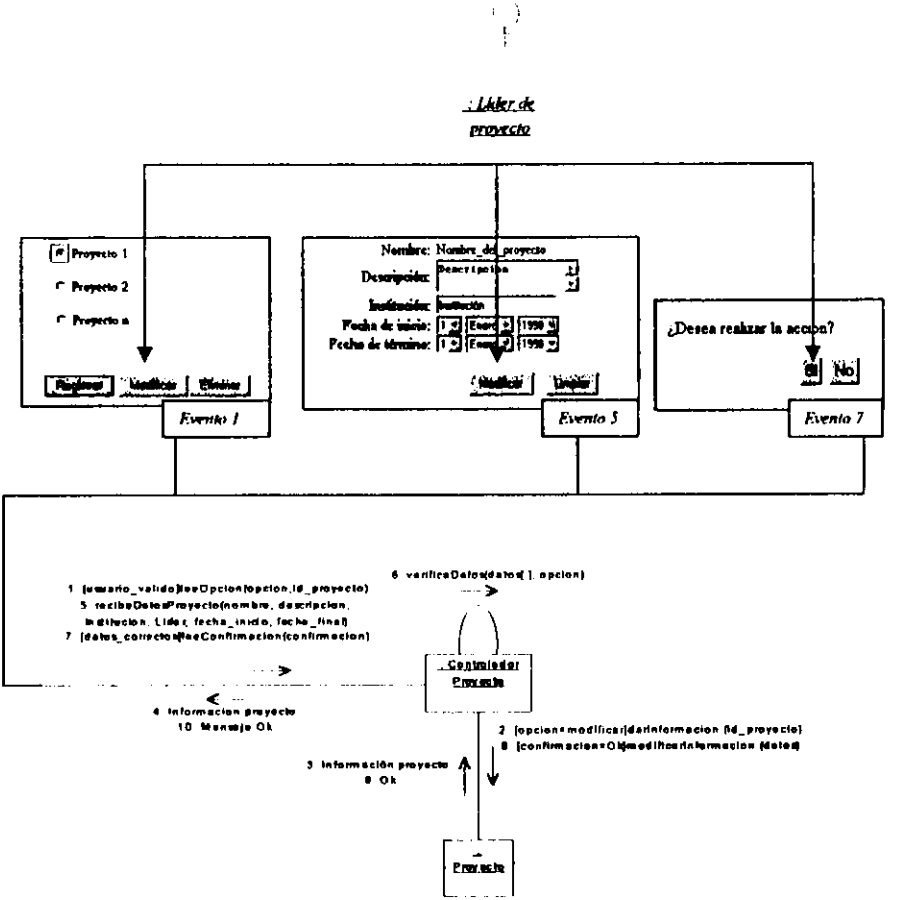


Diagrama 2.4 Eliminar información

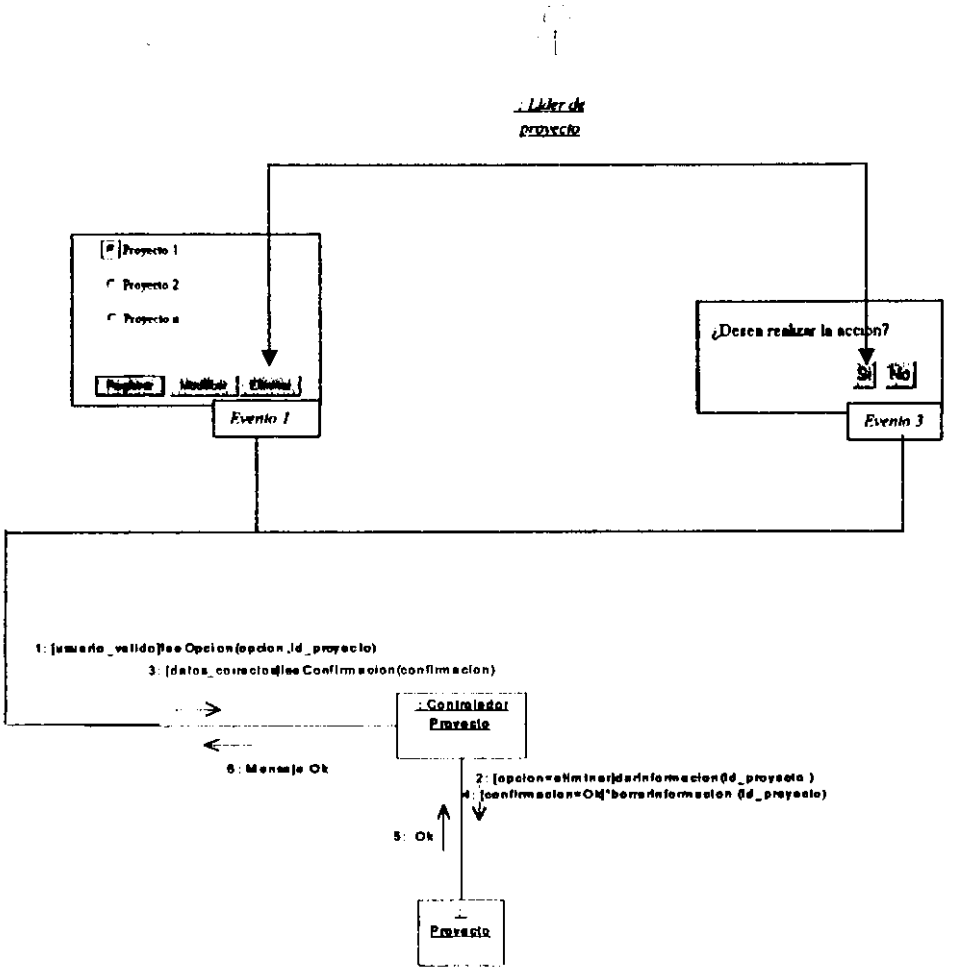
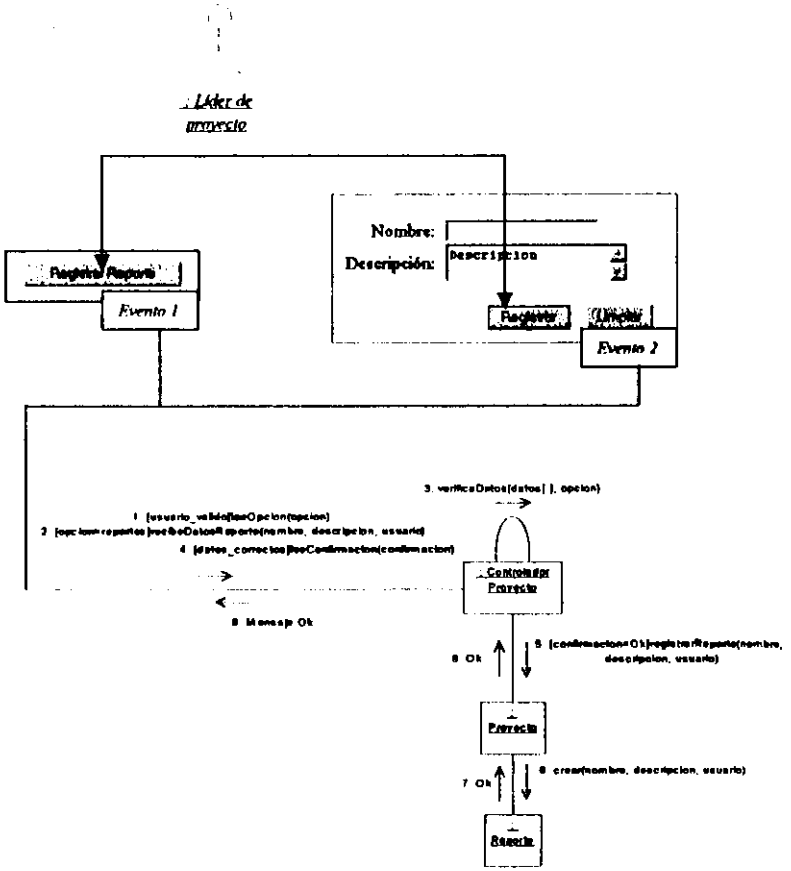


Diagrama 2.5 Registrar reportes



Bloque: CONTENIDOS DE INFORMACIÓN

Diagrama 3.1 Inicio calendario

El objeto, controlador Calendario, identificará al editor de información y le devolverá una pantalla con las opciones: registrar un nuevo evento y eliminar los eventos que él haya registrado anteriormente.

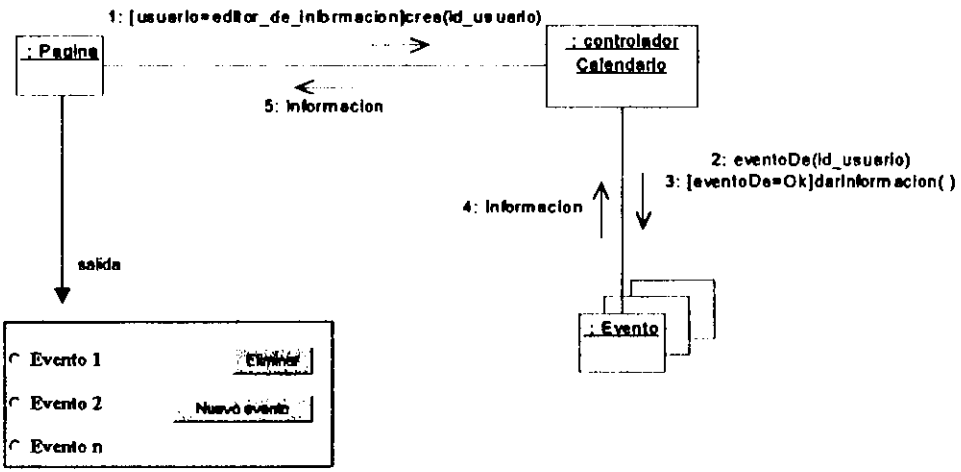


Diagrama 3.2 Calendarizar evento

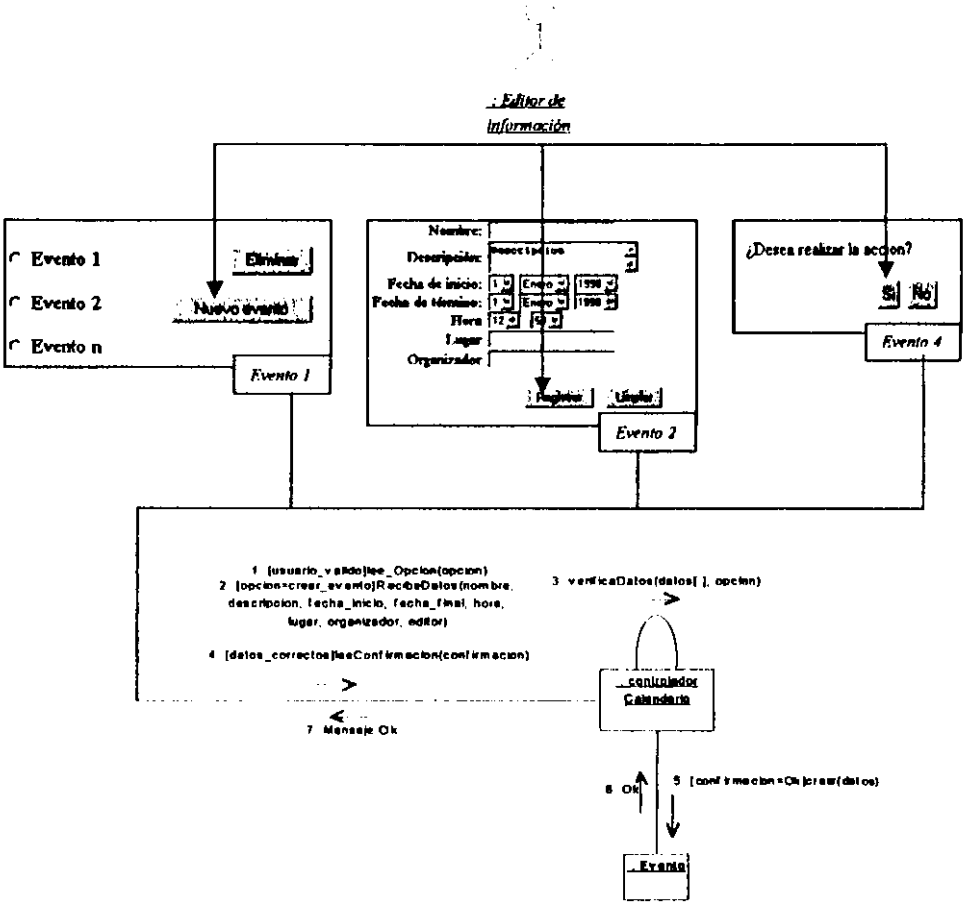


Diagrama 3.3 Cancelar eventos

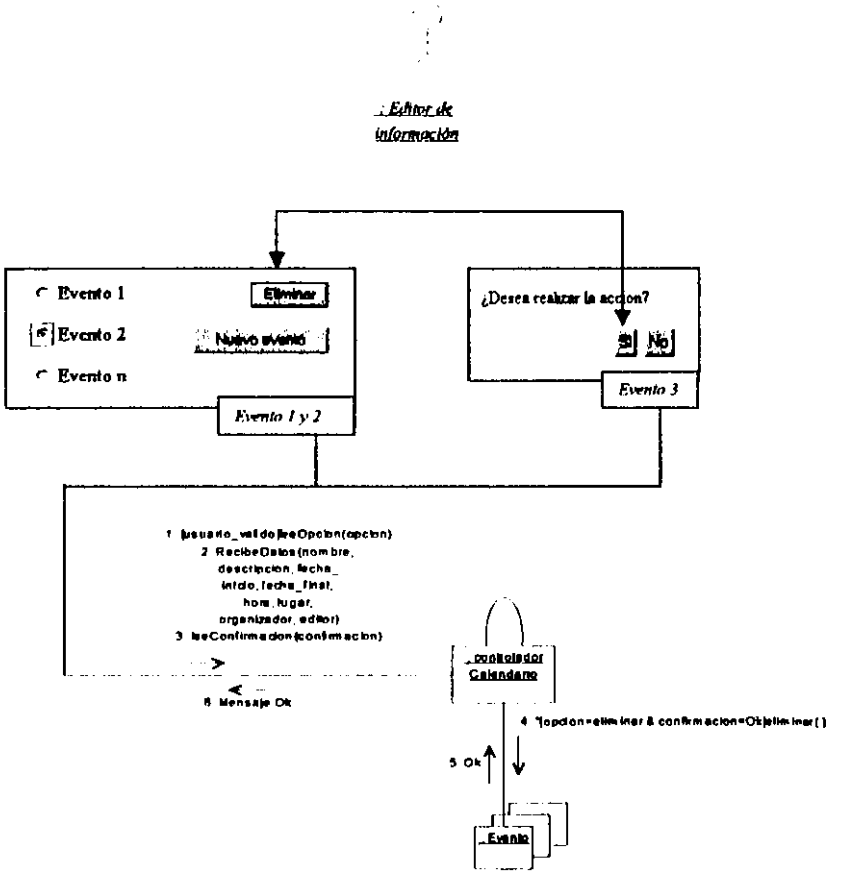


Diagrama 3.4 Inicio documentos

El objeto, controladorDocumentos, identificará al editor de información y le devolverá una pantalla con las opciones: registrar nuevos documentos y eliminar los documentos que él haya transferido anteriormente.

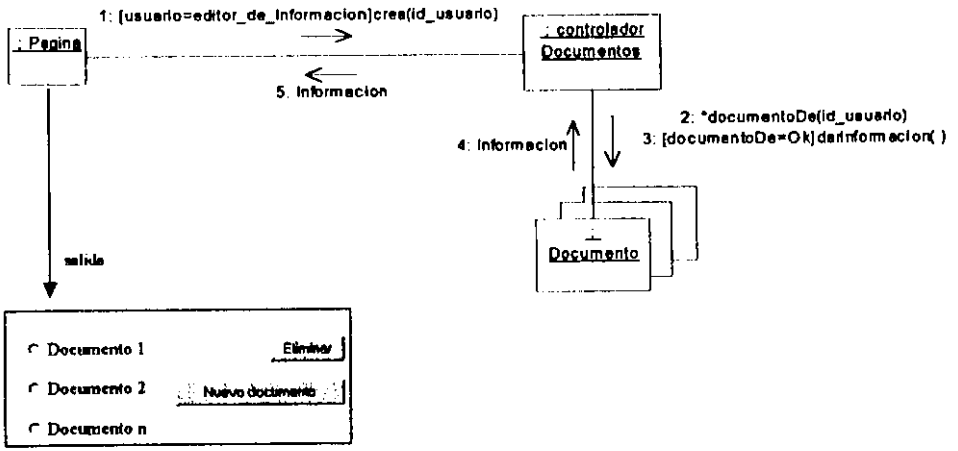


Diagrama 3.5 Recopilar documentos

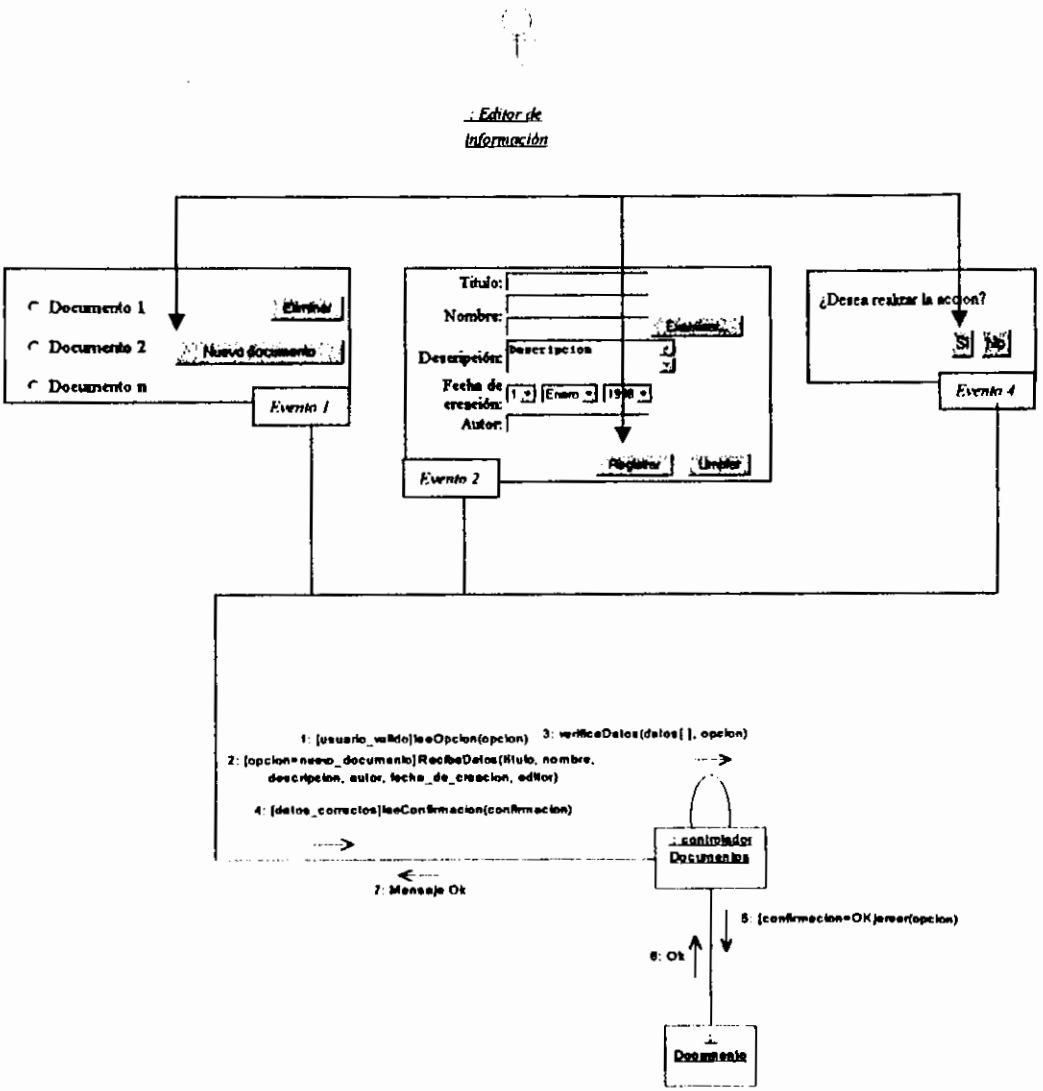


Diagrama 3.6 Eliminar documentos

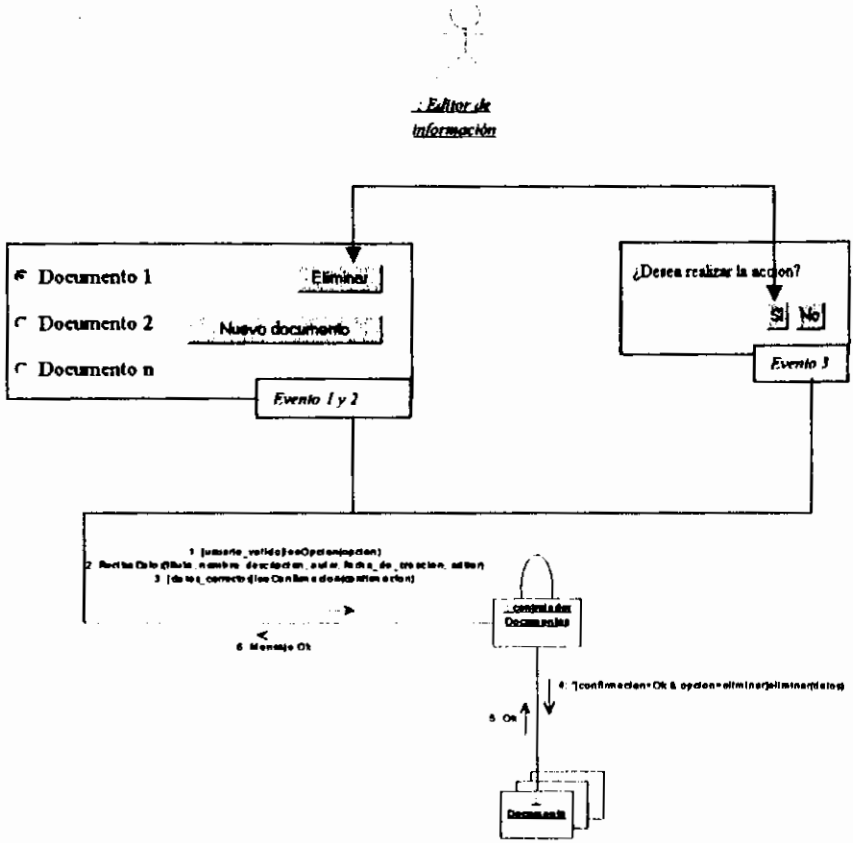


Diagrama 3.7 Captura Encuesta

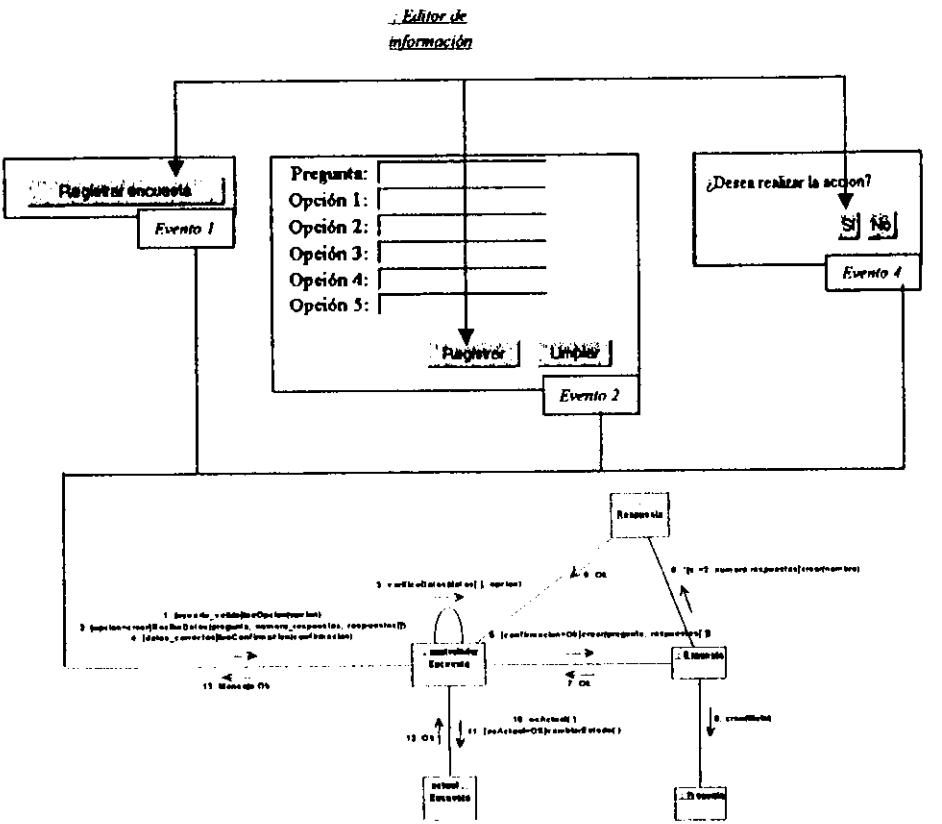


Diagrama 3.8 Inicio foro

El objeto controladorForo, devolverá una pantalla que contendrá las listas de los foros (moderados y no moderados) que el editor de información haya creado. Las opciones que tiene son: eliminación de foros, moderación (en caso de que el foro sea moderado) y la creación de un foro.

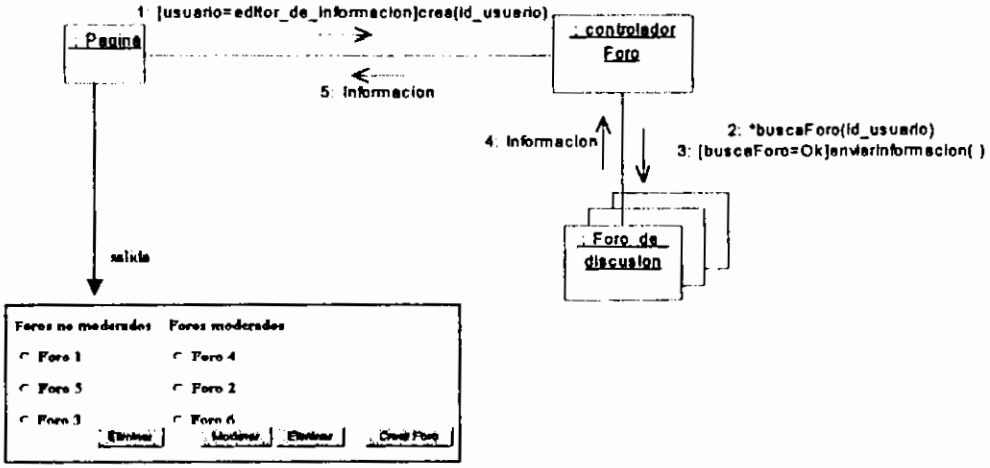


Diagrama 3.9 Creación de foro

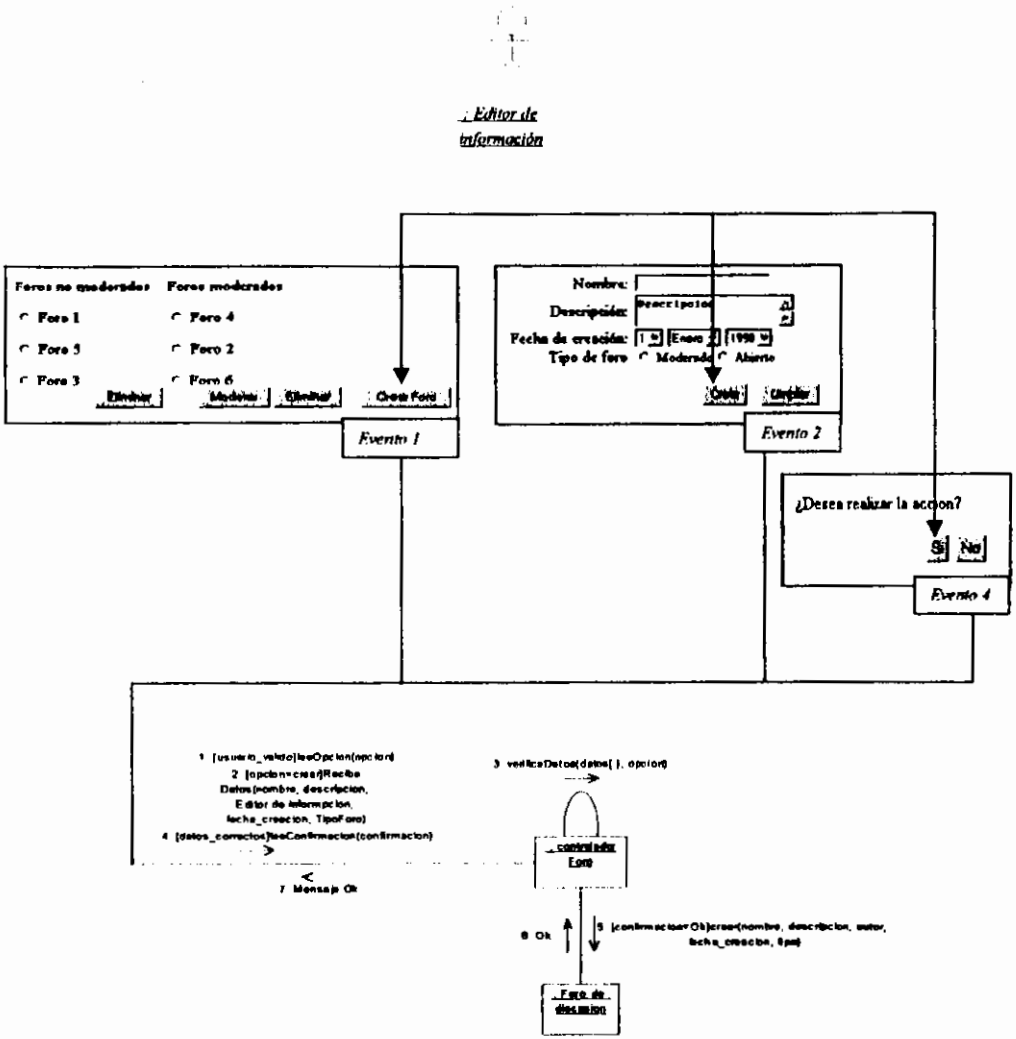


Diagrama 3.10 Eliminación de foro

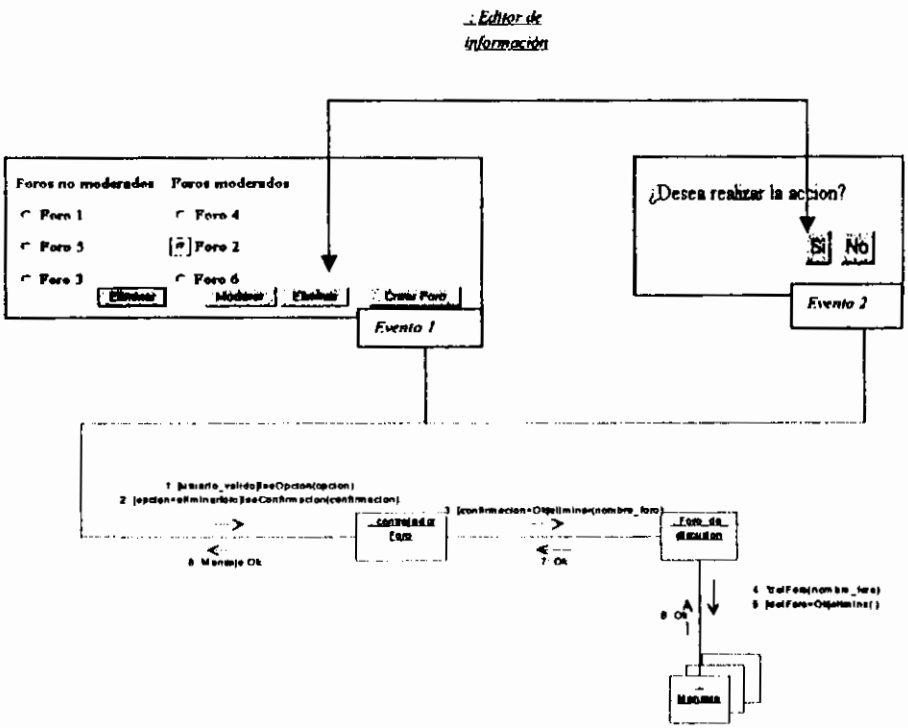


Diagrama 3.11 Publicar mensajes

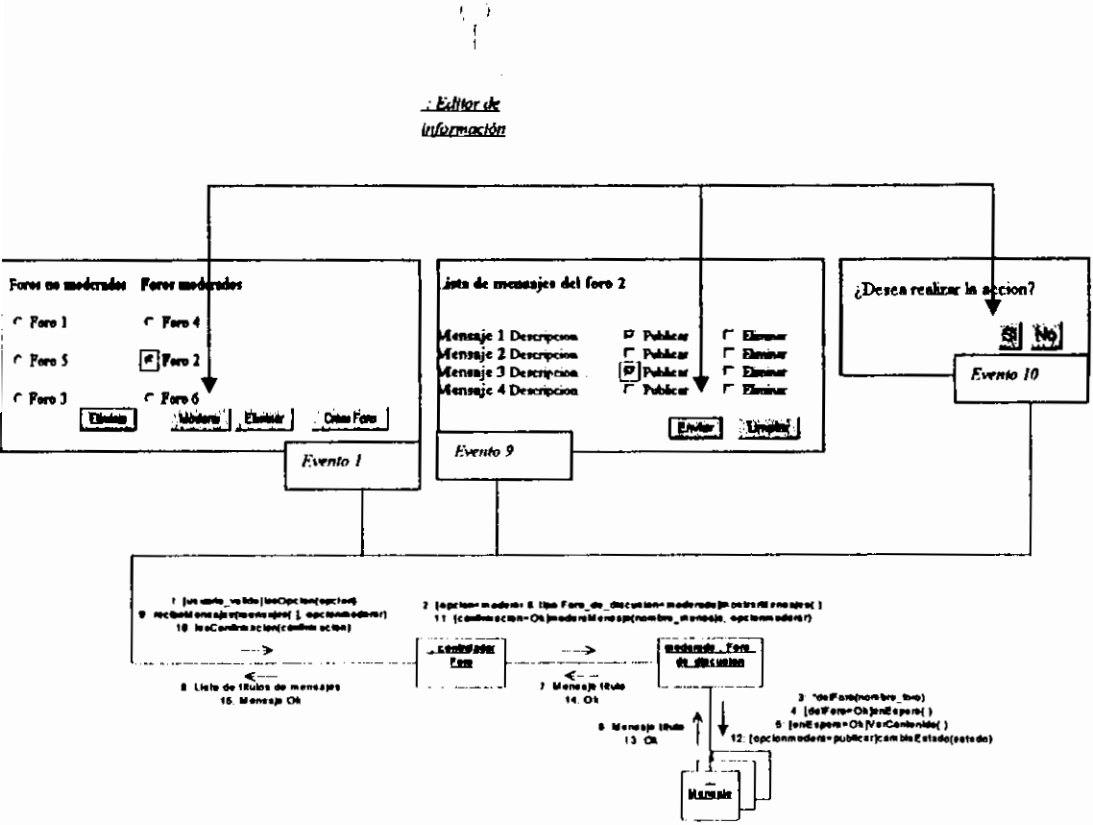


Diagrama 3.12 Bloquear mensaje

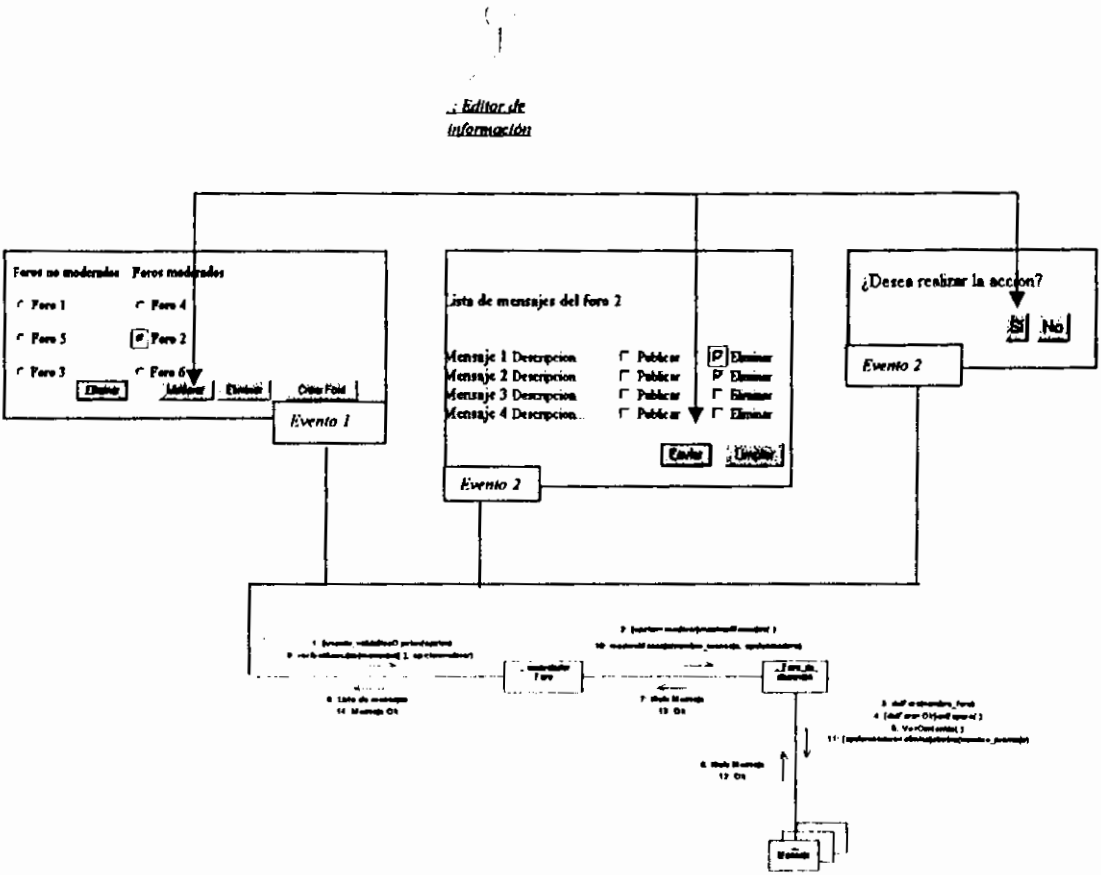


Diagrama 3.13 Inicio noticias

Cuando el controladorNoticias, reconoce al editor de información, tiene que verificar si el editor, además es editor jefe. Para el editor jefe, la pantalla que se le devolverá, contendrá las noticias, que están en espera de ser publicadas o eliminadas, junto con la opción de editar noticias. Para el editor de información, sólo se le desplegará la opción de editar noticias.

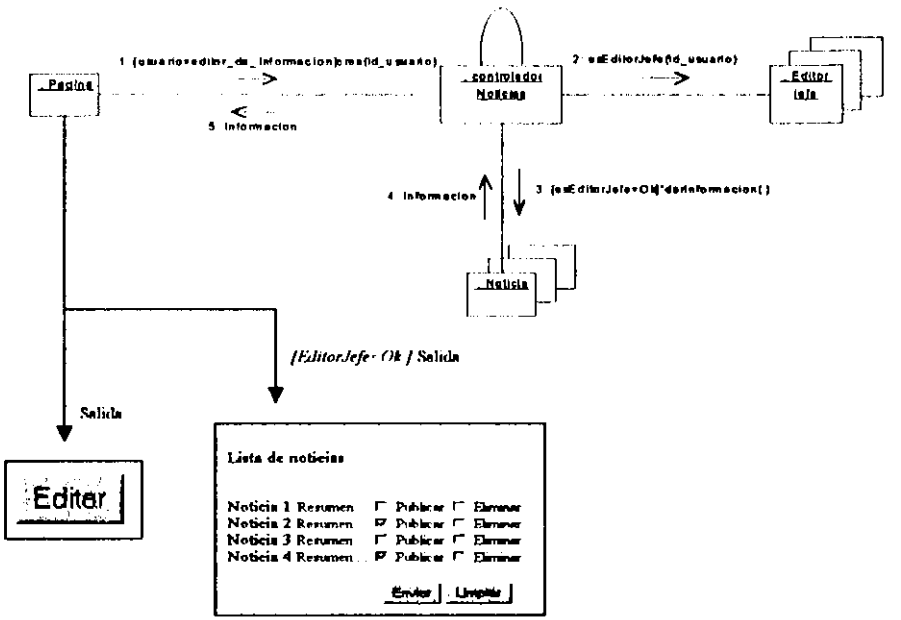


Diagrama 3.14 Editar noticias

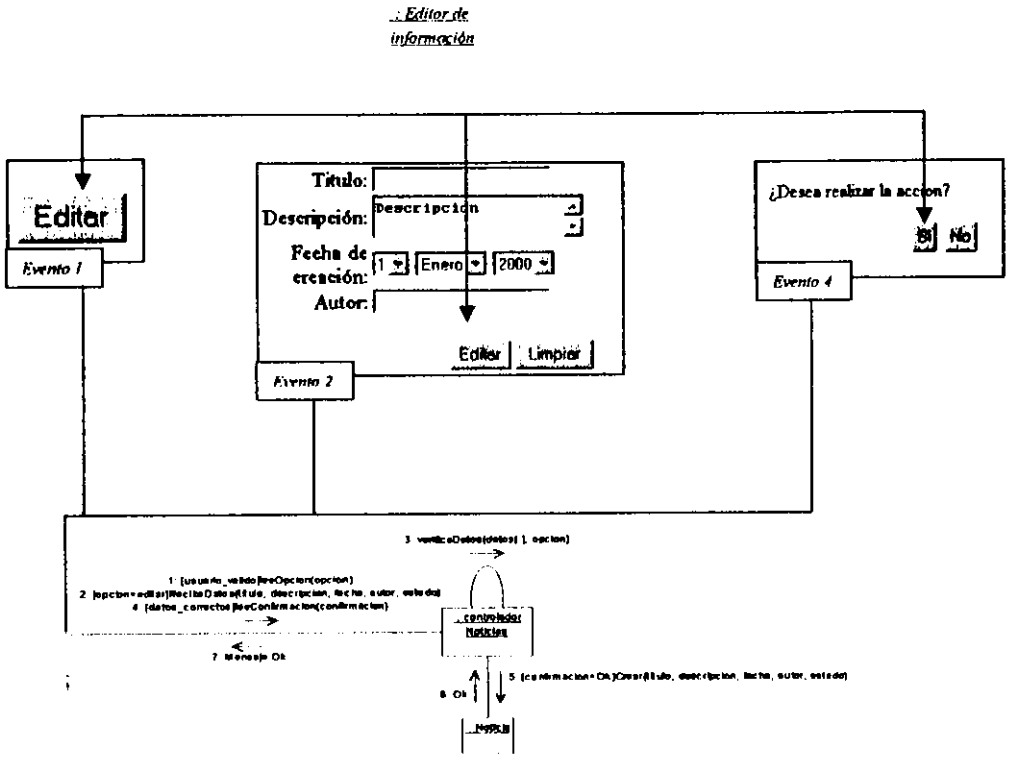


Diagrama 3.15 Publicar noticias

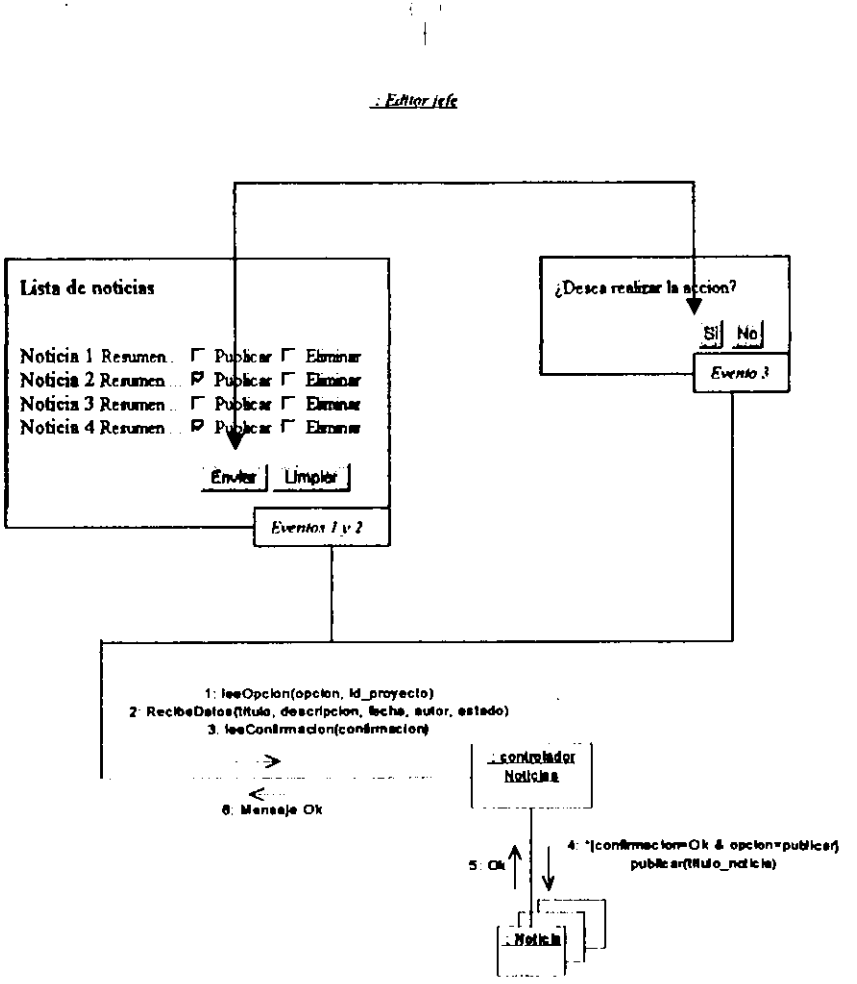
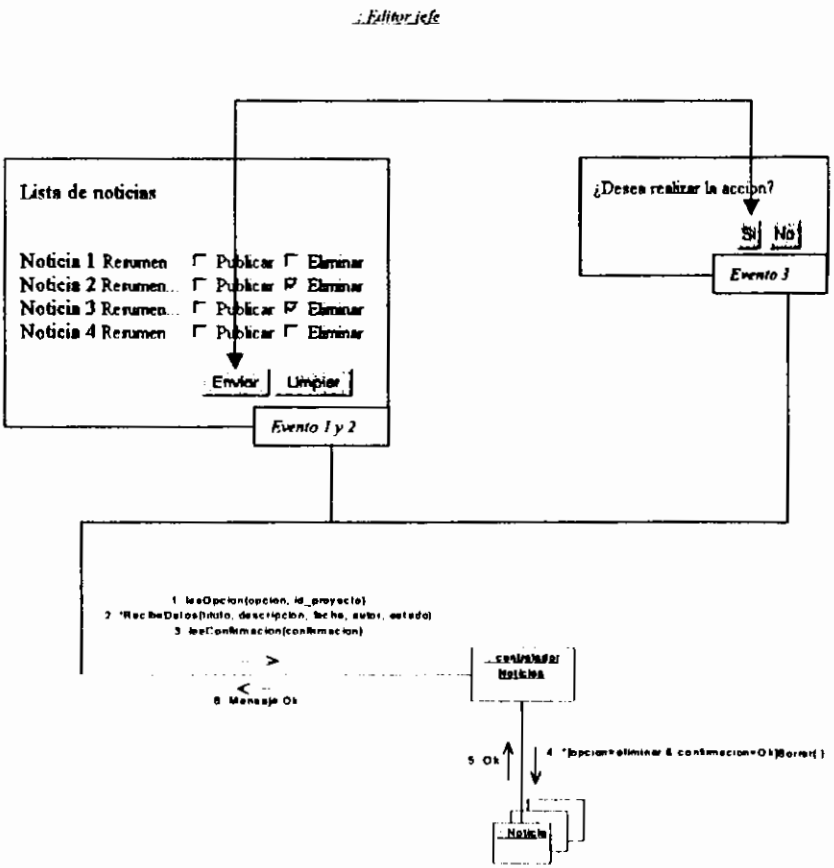


Diagrama 3.16 Eliminar noticias



Bloque: **VISITAS AL SITIO**

Diagrama 4.1Registro de miembro

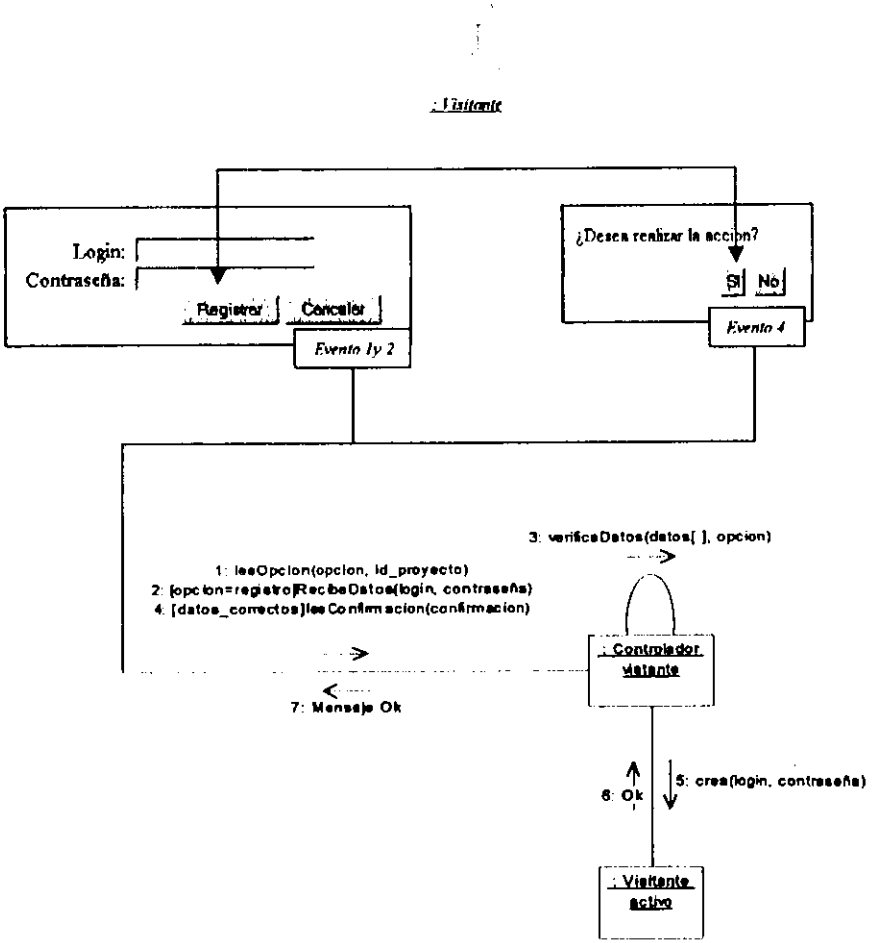


Diagrama 4.2 Responder encuesta

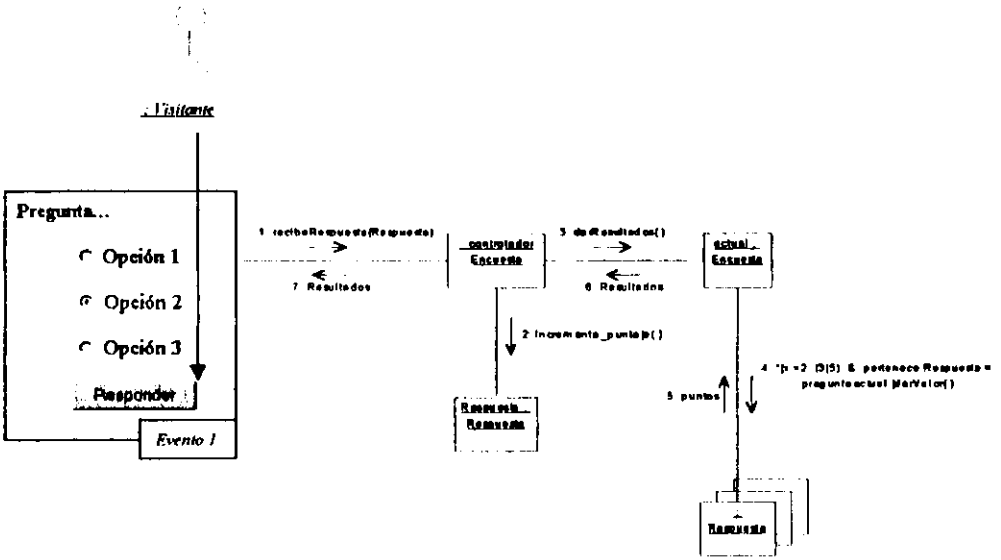


Diagrama 4.3 Consulta de mensaje

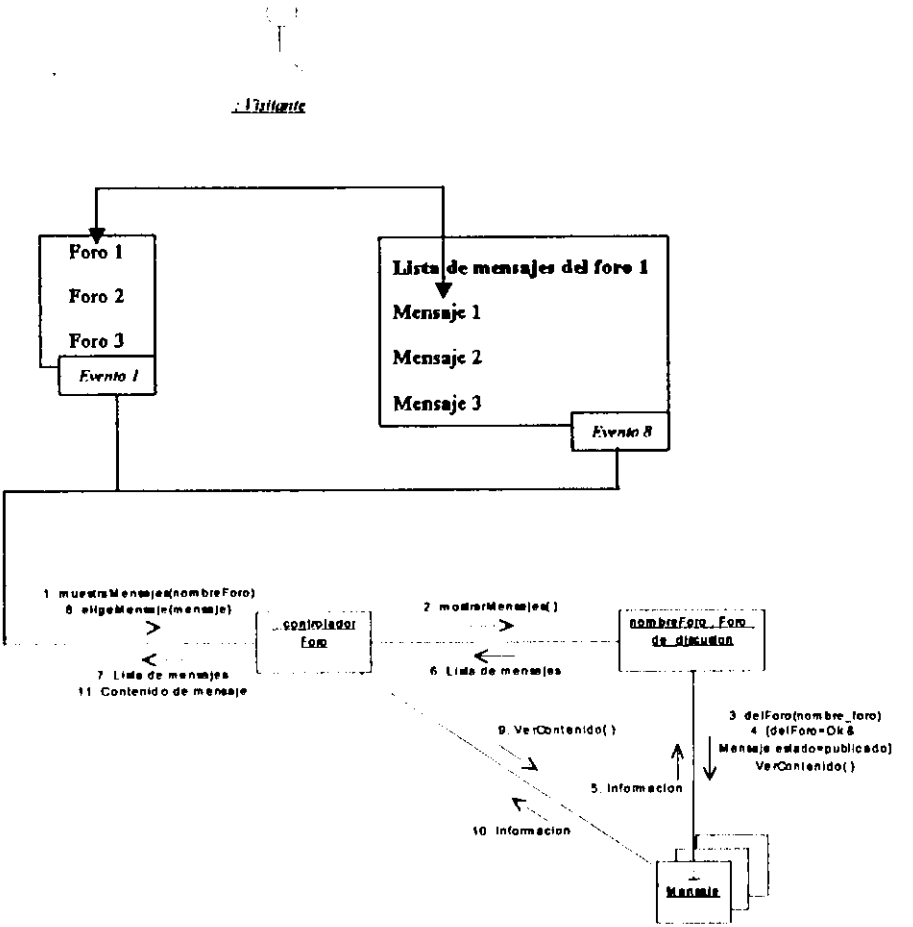


Diagrama 4.4 Enviar mensajes

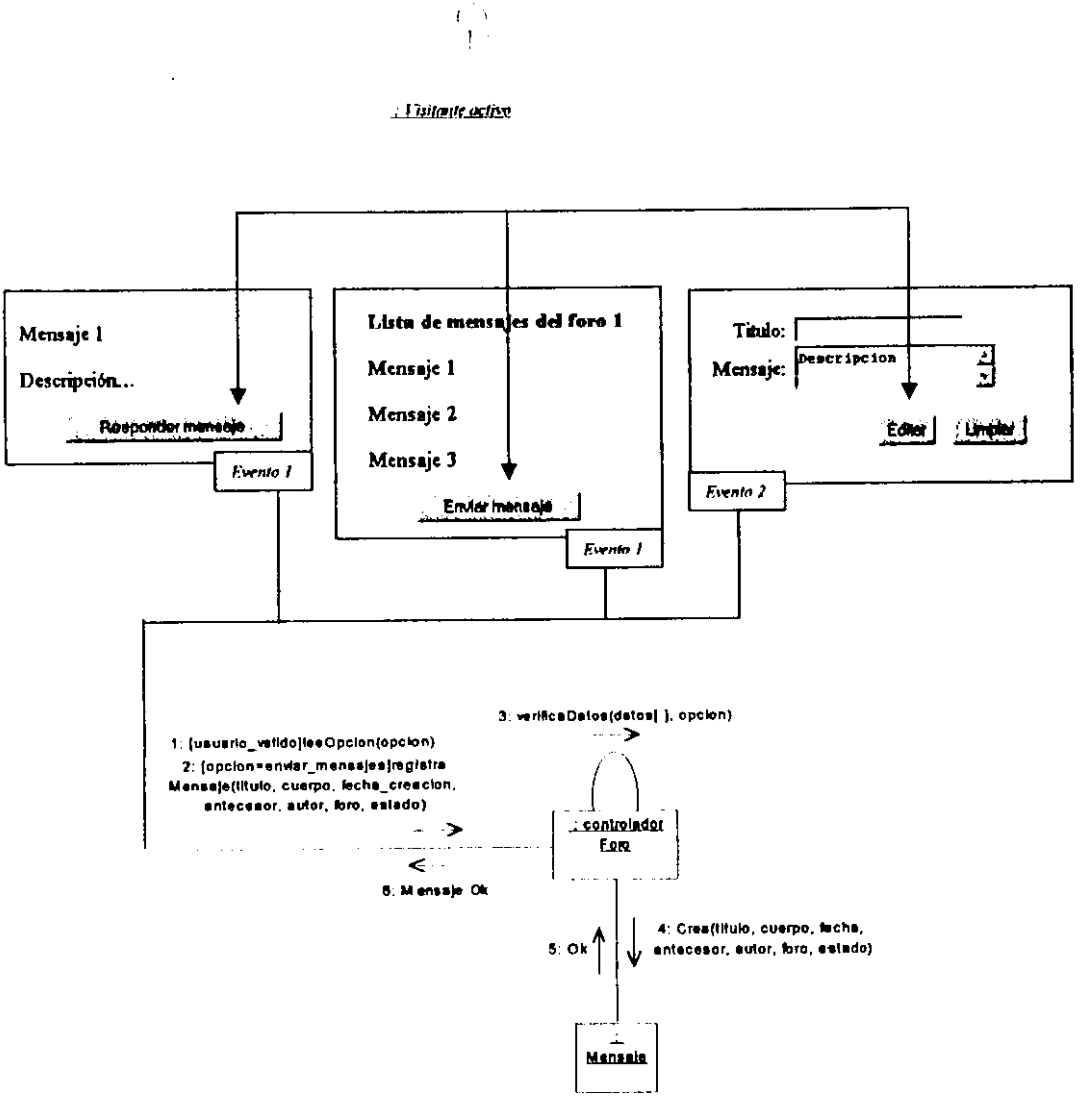


Diagrama 4.5 Ver documentos

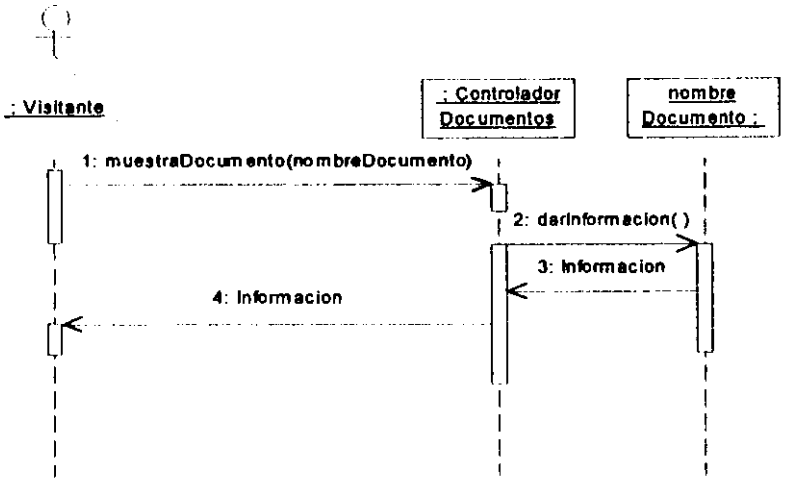


Diagrama 4.6 Calificar documentos

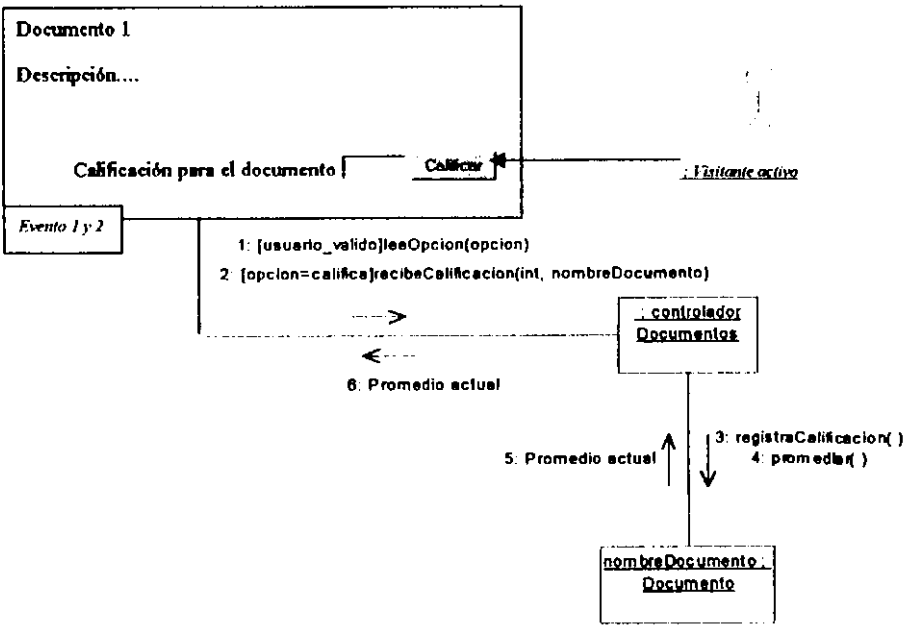


Diagrama de clases

Con la elaboración de los diagramas de secuencia y de colaboración se completó el diagrama de clases. Un diagrama de clases proporciona una notación gráfica formal para modelar clases y las relaciones entre unas y otras. Una clase se representa a través de un rectángulo que se divide en tres partes, indicando en la parte superior el nombre de la clase, en la parte central sus atributos, y en la parte inferior sus métodos. Cada una de estas partes se divide con una línea pintada.

Atributos
Cada nombre de un atributo puede ser seguido por detalles opcionales, tales como el tipo de dato y el valor por omisión, pero esto es dependiente del nivel de detalle deseado en el modelo de clases.

Métodos
Cada nombre de método puede ser seguido por detalles opcionales, tales como una lista de argumentos y el tipo de resultado que devuelven. Una lista de argumentos se escribe dentro de unos paréntesis seguidos del nombre del método; los argumentos son separados por una coma. El nombre y el tipo de los argumentos pueden ser especificados. El tipo de resultado que devuelve un método, se especifica después de los paréntesis y precedido por dos puntos. De la misma manera que en los atributos, esto depende del grado de detalle que se requiere.

Relaciones
Las relaciones entre clases, se grafican con una línea que une a las clases relacionadas. Es posible indicar cuantas instancias de una clase pueden relacionarse con cada instancia de la otra clase, a esto se le conoce como multiplicidad y dentro del diagrama de clases se indica justo arriba en un extremo de la línea de relación. La multiplicidad puede referirse a un número o a un rango; si describe un rango, entonces se escribe el límite inferior seguido de dos puntos y el límite superior. Para especificar los límites del rango se utilizan números o un asterisco que significa 1 o muchos.

Los tipos de relaciones que se encuentran en el diagrama son:

Agregación: Es una relación "parte-todo" en la cual una clase representa ser componente de una clase que representa ser el todo. Esta relación se muestra con una línea que contiene un diamante en uno de sus extremos.

Herencia: Es una relación de generalización o especialización entre clases, en la cual una clase nombra a otra como su padre[19]. Cuando una clase es hija de otra, se dice que la clase hija "heredo" todas las propiedades y operaciones de la clase padre, con la posibilidad de agregarle más funcionalidad. Para indicar que una clase hereda de otra se dibuja una flecha que apunta a la clase padre.

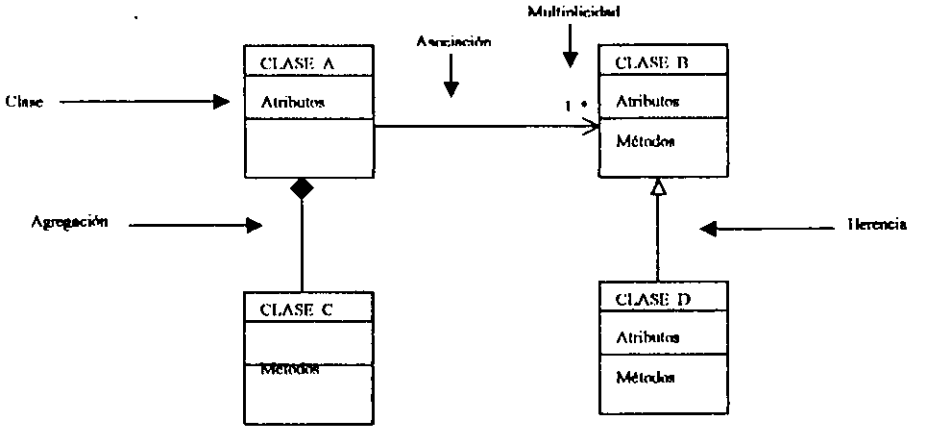
Asociación: Es una relación que representa un uso o conocimiento entre clases. Se dibuja con una línea.

De la siguiente figura se pueden entender las siguientes situaciones:

- CLASE A se asocia con CLASE B.
- Una instancia de CLASE A se puede asociar con 1 o muchas instancias de CLASE B.
- CLASE A no posee métodos
- CLASE A contiene a CLASE C
- CLASE C es parte de CLASE A

[19] Este término también es conocido como superclase.

- CLASE B es padre o superclase de CLASE D
- CLASE D es subclase o hija de CLASE B
- CLASE D heredará todos los atributos y métodos de CLASE B y pude tener los propios
- CLASE C no tiene atributos



Notación utilizada en un diagrama de clases

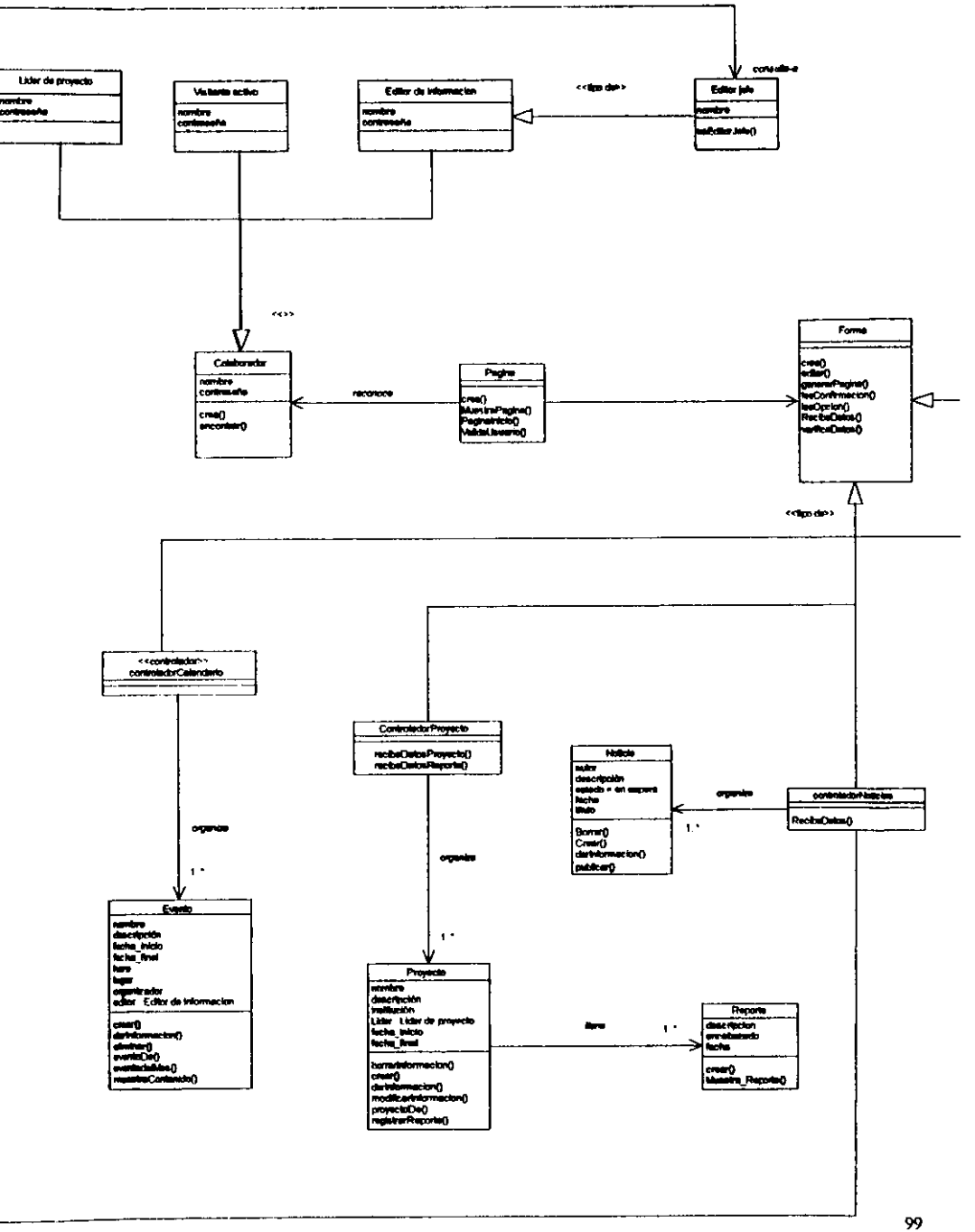
Diccionario de datos

El diagrama de clases es de mucha utilidad para el entendimiento en términos de software del sistema, sin embargo no es suficiente para indicar con detalle cada componente, por lo que en este trabajo se presenta también el diccionario de datos, tan bien conocido como glosario de clases.

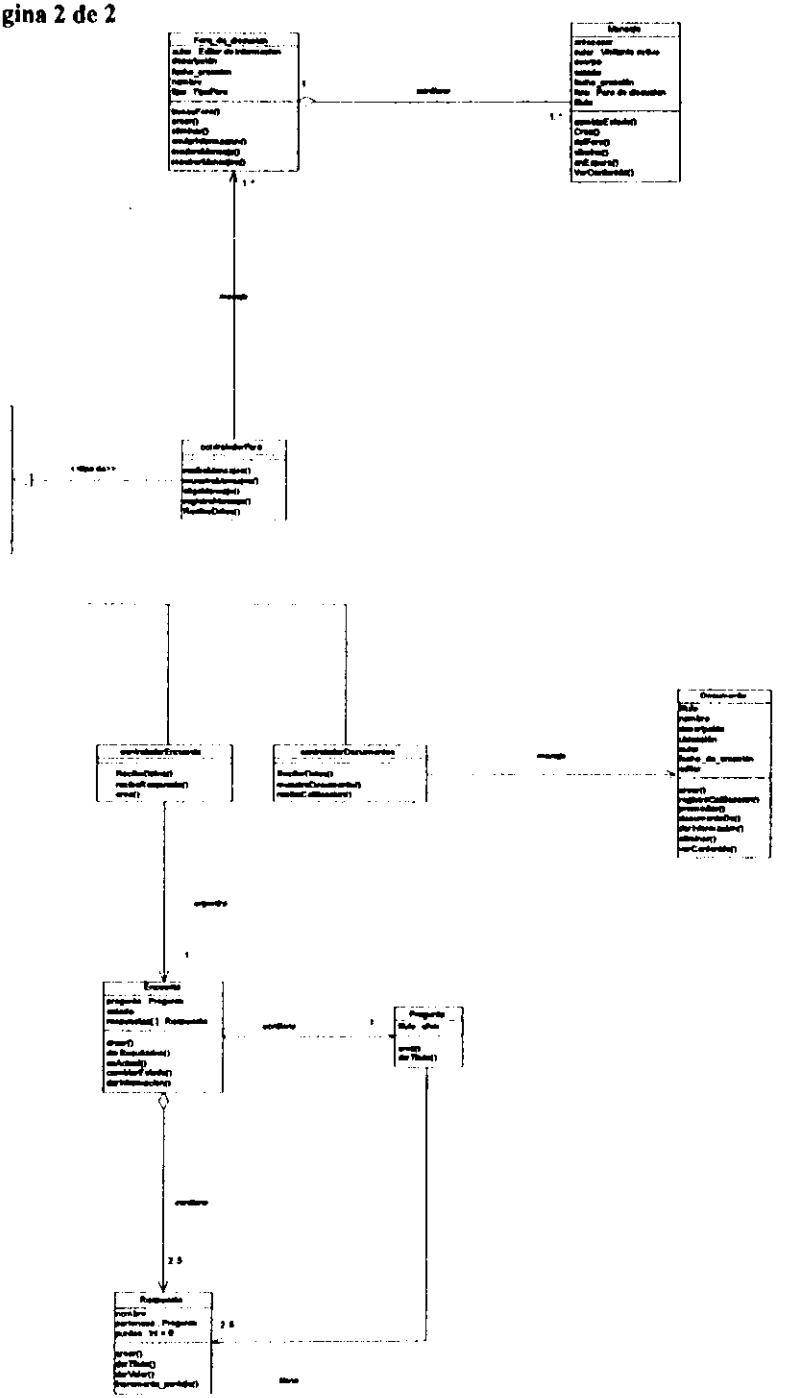
El diccionario de datos es un documento que posee información referente a las descripciones de las clases, atributos y métodos de las mismas. No existe un formato especial que sea requerido para elaborar el diccionario de datos, por lo que en este proyecto el diccionario describe en orden alfabético cada clase que aparece en el diagrama y debajo de su descripción, los atributos y métodos que la componen. A diferencia de los atributos, los métodos están escritos seguidos de paréntesis, y las clases se diferencian debido a que su nombre está resaltado del resto del texto.

DIAGRAMA DE CLASES (completo)

Diagrama de clases
Página 1 de 2



Página 2 de 2



Diccionario de datos

Colaborador
Clase que representa a cualquier persona que colabore en la alimentación de información del sitio.

nombre
Clave del colaborador.

contraseña
Palabra que servirá para la autentificación del colaborador.

crea( )
Crea una instancia de la clase.

encontrar( )
Verifica que la clave y contraseña que le son pasados como parámetros, correspondan al valor de sus atributos. Devuelve un booleano.

controladorCalendario
Clase que se encarga de administrar a la clase Calendario, manejando las interfaces adecuadas para cada tipo de usuario.

RecibeDatos( )
Recibe como parámetros los datos del evento; nombre, descripción, fecha de inicio y término, lugar y hora en que se llevará a cabo, el nombre del organizador y el editor de información que lo está registrando.

controladorDocumento
Clase que se encarga de administrar a la clase Documento, manejando las interfaces adecuadas para cada tipo de usuario.

muestraDocumento( )
Solicita a objetos del tipo documento información de sí mismos y devuelve una lista con los nombres de los documentos que se están publicando.

recibeCalificacion( )
Recibe dos parámetros, la calificación otorgada y el nombre del documento que se está calificando. Verifica que los parámetros sean validos y llama al objeto correspondiente para registrar la calificación que ha otorgado el visitante.

recibeDatos( )
Recibe los datos del documento que se desea registrar; titulo, nombre, descripción, autor, fecha de creación y el editor está realizando la operación.

controladorEncuesta
Clase que se encarga de administrar a la clase Encuesta, manejando las interfaces adecuadas para cada tipo de usuario.

recibeRespuesta
Recibe como parámetro la respuesta que ha seleccionado el visitante y la envía al objeto Respuesta con el fin de que éste incremente su puntuación.

recibeDatos
Recibe los datos para que la encuesta sea creada; pregunta, número de respuestas y opciones de respuesta.

controladorForo
Clase que se encarga de administrar a la clase Foro, manejando las interfaces adecuadas para cada tipo de usuario.

eligeMensaje( )
Recibe el nombre del mensaje que desea consultar un visitante. Y solicita al objeto mensaje le proporcione información de sí mismo para desplegarla al visitante.

muestraMensajes( )
Llama al foro que recibe como parámetro, para pedirle información de los mensajes que lo integran y devuelve la lista de estos mensajes.

recibeDatos( )
Recibe los datos necesarios para la creación de un foro; nombre, descripción y fecha de creación del foro, tipo del foro (moderado o abierto) y el editor de información que lo está creando.

recibeMensajes( )
En el proceso de moderación de mensajes, recibe el nombre de uno o más mensajes seleccionados para ser publicados o eliminados.

registraMensaje( )
Cuando un visitante activo participa en un foro de discusión, envía mensajes, y éste método tiene por función recibir el titulo, cuerpo, fecha de creación, nombre del mensaje antecesor (en el caso de que lo tenga) del mensaje, así como también el nombre del visitante que esté enviando el mensaje, el foro al que pertenece y el estado (publicado o en espera) del mensaje.

controladorNoticias
Clase que se encarga de administrar a la clase Noticia, manejando las interfaces adecuadas para cada tipo de usuario.

RecibeDatos( )
Se encarga de recibir el titulo, la descripción, la fecha, el estado(en espera o publicada) y el autor de una noticia.

controladorProyecto
Clase que se encarga de administrar a la clase Proyecto, manejando las interfaces adecuadas para cada tipo de usuario.

recibeDatosProyecto( )
Se encarga de recibir los datos correspondientes para el registro o modificación de un proyecto. Recibe como parámetros los siguientes datos: nombre, descripción, institución y líder del proyecto; fechas en que inicia y concluye el desarrollo del proyecto.

recibeDatosReporte( )
Captura todos los datos que se envían a través de la forma para registrar un reporte de algún proyecto. Recibe como parámetros los siguientes datos: el nombre y descripción correspondientes al reporte y el nombre del usuario que lo registra.

Documento
Clase que representa al documento publicado en el sitio.

autor
El creador del documento.

descripcion
Descripción del documento.

editor
El editor de información que publicó el documento.

fecha_de_creación
Fecha de publicación en el sitio.

nombre
Nombre del archivo que contiene el documento

ubicacion
Ruta absoluta en donde reside el archivo dentro del servidor.

titulo
Encabezado del documento

crear( )
Método constructor. Crea una instancia de la clase Documento.

darInformacion( )
Devuelve el nombre y descripción de un documento.

documentoDe( )
Verifica si un documento fue hecho por un determinado editor de información. Recibe como parámetro el identificador del editor de información y devuelve un booleano.

eliminar( )
Destruye una instancia de la clase Documento.

promediar( )
Calcula el promedio de las calificaciones que le han sido dadas a un documento.

registrarCalificacion( )
Recibe como parámetro un número y lo registra como calificación que se le otorga a un documento.

verContenido( )
Devuelve toda la información referente a un documento.

Editor de información
Clase que representa al editor de información, dentro del sistema.

nombre
Clave el editor de información.

contraseña
Palabra que servirá para la autentificación del editor de información.

Editor jefe
Clase que es una especialización del editor de información y representa al editor jefe, dentro del sistema.

nombre
Clave del editor jefe.

contraseña
Palabra que servirá para la autentificación del editor jefe.

esEditorJefe( )
Tiene por objetivo reconocer si la clave que esta recibiendo como parámetro corresponde con una clave de editor jefe. Devuelve un booleano.

Encuesta
Clase que representa la encuesta.

estado
Indica si actualmente está publicada o no.

pregunta
Pregunta de la encuesta.

respuestas
Contiene las opciones para responder la encuesta.

cambiarEstado( )
Modifica el valor del atributo estado de la encuesta.

crear( )
Método constructor. Crea una instancia de la clase Encuesta.

darResultados( )
Obtiene la puntuación de cada opción de respuesta y presenta los resultados totales de la encuesta.

esActual( )
Indica si el objeto corresponde a la encuesta que actualmente está publicada.

darInformacion( )
Obtiene la pregunta y respuestas de la encuesta publicada actualmente.

Evento
Clase que representa un evento que se publicará en el calendario, a fin de invitar a participar en él a los visitantes del sitio.

descripcion
Descripción del evento.
editor
El editor de información que publicó el evento.

fecha_inicio y fecha_final
Dia, mes y año en que comenzará y terminara el evento.

hora
Horario en que se llevara a cabo el evento

lugar
Dirección del lugar donde se llevará a cabo el evento.

nombre
Nombre del evento.

organizador
Nombre de la persona u organización que organice el evento.

crear( )
Método constructor. Crea una instancia de la clase Evento.

darInformacion( )
Devuelve información general de un evento.

eliminar( )
Destruye una instancia de la clase Evento.

eventoDe( )
Comprueba que un evento haya sido registrado por el editor de información, cuya clave recibe como parámetro.

eventodelMes( )
Indica si el evento se llevara a cabo en el mes actual. Devuelve un booleano.

muestraContenido( )
Devuelve información completa de un evento.

Forms
Clase padre de las aquellas cuya función es administrar a otras clases; maneja las interfaces adecuadas para cada tipo de usuario.

crea( )
Genera una instancia de la clase.

editar( )
Solicita información a una determinada clase para que esta sea modificada, por su propietario, el cual reconoce a través del identificador que recibe como parámetro. Devuelve una página con esta información y las opciones que se permiten para manejar dicha información.

generarPagina( )
Solicita información a las clases para que sea desplegada en forma de páginas web.

leeConfirmacion( )
Envía un mensaje para que el usuario confirme o cancele, la operación que se solicito realizar.

leeOpcion( )
Recibe como parámetro la operación que el usuario desea realizar, y dependiendo de ésta será lo que se despliegue en pantalla.

RecibeDatos( )
Captura los datos que le son enviados a través de un formulario, y dispone de ellos.

verificaDatos( )
Comprueba que los datos que ha recibido están completos y son validos. Devuelve un booleano.

Foro de discusión
Clase que representa un foro de discusión virtual.

autor
Persona que creó el foro.

descripcion
Descripción del foro de discusión.

fecha_creacion
Día, mes y año en que fue creado el foro.

nombre
Titulo del foro de discusión.

tipo
Indica si el foro es moderado o abierto.

buscaForo( )
Indica si el valor de su atributo autor es igual al que se le envía como parámetro. Devuelve un booleano.

crear( )
Crea una instancia de la clase Foro.

eliminar( )
Destruye una instancia de la clase Foro de discusión

enviarInformacion( )
Devuelve información general del objeto foro de discusión.

moderaMensaje( )
Se comunica con un objeto de la clase Mensaje y le indica que operación realizar. Recibe como parámetros, el mensaje con el cual se debe comunicar y que operación indicarle.

mostrarMensajes( )
Identifica que mensajes le corresponden al foro y devuelve una lista de ellos.

Líder de proyecto
Clase que representa al líder de proyecto, dentro del sistema.

nombre
Clave del líder de proyecto.

contraseña
Palabra que servirá para la autentificación del líder de proyecto.

Mensaje
Clase que representa un mensaje del foro de discusión.

antecesor
Indica el mensaje al cual se está respondiendo.

autor
Persona que está enviando el mensaje.

cuerpo
Contenido del mensaje.

estado
Indica si el mensaje esta en espera de ser publicado o ya está publicado en el sitio.

fecha_creación
Día, mes y año en que se envió el mensaje.

foro
Foro al que pertenece el mensaje.

titulo
Encabezado del mensaje.

cambiaEstado( )
Cambia el valor del atributo estado, permitiendo la publicación del mensaje. Se utiliza cuando el foro de discusión al que pertenece el mensaje es moderado.

Crea( )
Genera una instancia de la clase Mensaje.

delForo( )
Indica si pertenece o no a un foro.

elimina( )
Borra al propio mensaje.

enEspera( )
Indica si el mensaje está en estado de espera, es decir que sea necesario la autorización del editor de información para su publicación en el sitio.

VerContenido( )
Muestra información completa del mensaje.

Noticia
Clase que representa una noticia del sitio.

autor
Persona que editó la noticia.

descripción
Contenido de la noticia.

estado
Indica si la noticia esta en estado de espera para su publicación
fecha
Día, mes y año de edición de noticia.

titulo
Encabezado de la noticia.

Borrar( )
Elimina una noticia.

Crear( )
Genera una instancia de la clase Noticia.

darInformacion( )
Da detalles de la noticia.

publicar( )
Cambiar el valor del atributo estado, permitiendo su publicación en el sitio.

Pagina
Clase que representa la página principal que controla a las demás páginas.

crea( )
Genera una instancia del objeto Pagina. Es el primer método que se invoca cuando se visita el sitio.

MuestraPagina( )
Devuelve la página correspondiente a un determinado usuario.

PaginaInicio( )
Tiene por objetivo mostrar la pagina de inicio del sistema, y lo hace invocando a los métodos generarPagina de las siguientes clases: controladorProyecto, controladorForo, controladorCalendario, controladorDocumentos, controladorNoticias y controfadorEncuesta.

ValidaUsuario( )
Captura la clave y contraseña que un visitante envía.

Pregunta
Clase que representa la pregunta en una encuesta.

titulo
Contenido de la pregunta.

crea( )
Crea una instancia de la clase Pregunta.

darTitulo( )
Devuelve el valor del único atributo que posee el objeto.

Proyecto
Clase que representa un proyecto de Internet2.

Nombre
Nombre del proyecto.

descripción
Descripción del proyecto.

institución
Nombre de la institución que respalda el proyecto.

Líder
Persona responsable del proyecto.

fecha_inicio
Día, mes y año en que comienza el desarrollo del proyecto.

fecha_final
Día, mes y año en que termina el proyecto.

crear( )
Crea una instancia a la clase Proyecto.

borrarInformacion( )
Elimina una instancia de la clase proyecto.

darInformacion( )
Proporciona información del proyecto.

modificarInformacion( )
Modifica el valor de algunos atributos de la clase proyecto.

proyectoDe( )
Verifica si el proyecto es de un líder de proyecto determinado.

registrarReporte( )
Captura los datos que se requieren para generar un reporte del proyecto.

Reporte
Clase que Representa un reporte de un proyecto.

descripción
Descripción del reporte.

encabezado
Titulo del reporte

fecha
Día, mes y año en que se realizó el reporte.

crear( )
Genera una instancia de la clase reporte.

Muestra_Reporte( )
Muestra el contenido de un reporte.

Respuesta
Clase que representa una respuesta de la encuesta.

nombre
La respuesta.

pertenece
La encuesta a la que pertenece.

puntos
Número que corresponde al número de veces que ha sido seleccionada esta respuesta.

crear( )
Crea una instancia de la clase Respuesta.

darTitulo( )
Devuelve el valor del atributo nombre.

darValor( )
Devuelve el valor del atributo puntos del objeto.

IncrementaPuntaje( )
Incrementa en uno el valor del atributo puntos del objeto.

Visitante activo
Clase que representa el visitante que se ha inscrito y podrá participar en los contenidos del sitio, tales como los foros de discusión y la calificación de documentos.

nombre
Clave del visitante activo.

contraseña
Palabra que servirá para la autentificación del visitante activo.

Arquitectura del software

La arquitectura lógica del software del sistema está compuesta por las siguientes tres capas:

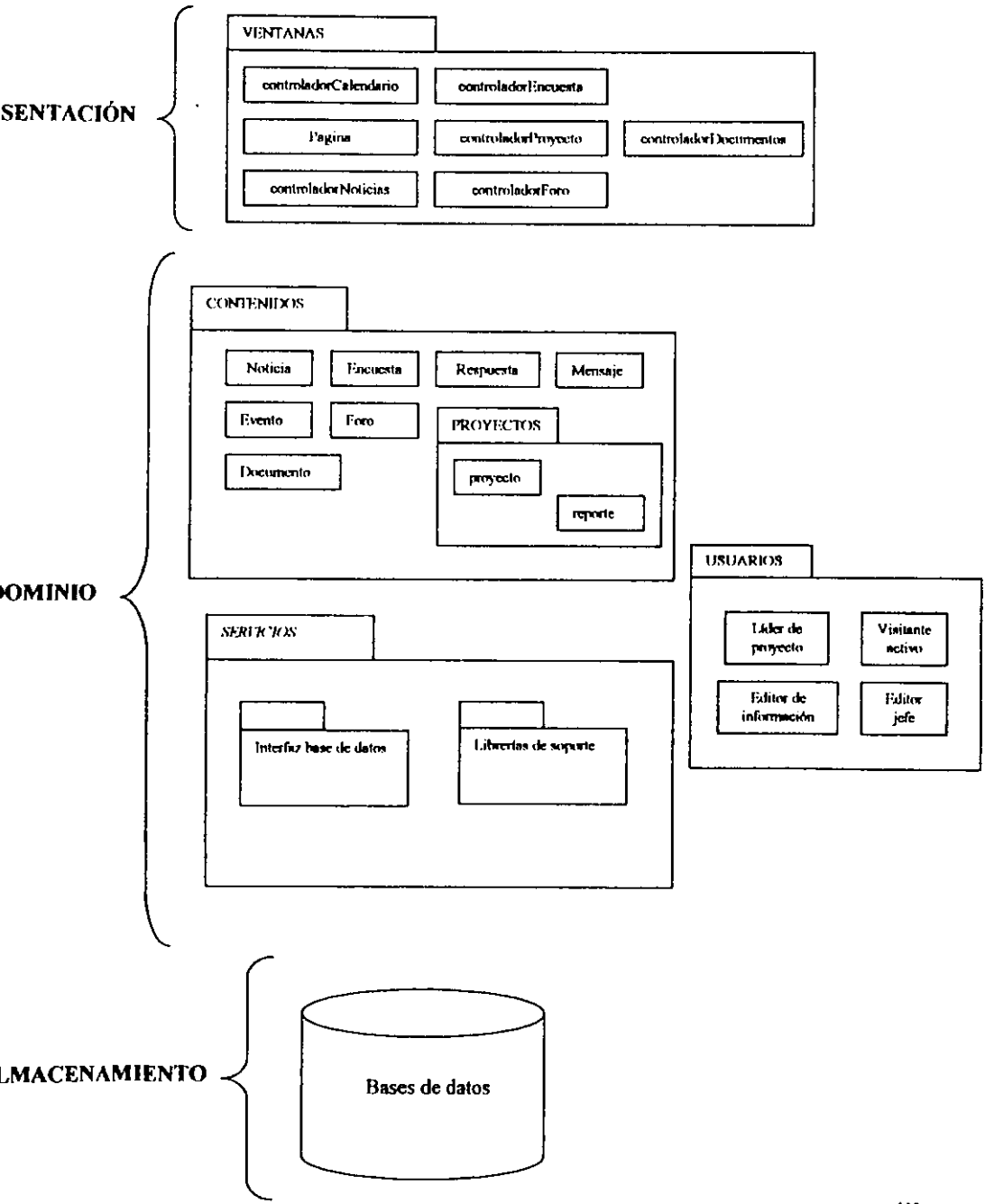
- **Presentación**: compuesta por todas las clases del tipo controlador, encargadas de la administración de ventanas y eventos, que utiliza y genera el usuario.
- **Dominio**: conocida también como lógica de aplicaciones, contempla todos los objetos del dominio y las clases involucradas con la prestación de servicios tales como la conexión a la base de datos y la utilización de librerías estándar que proporcionan funcionalidad bien definida a algunos componentes del sistema.
- **Almacenamiento**: se integra del medio de almacenamiento que se utilice, en este caso de una base de datos relacional y surge debido a la necesidad de guardar permanentemente el valor de los atributos de objetos persistentes[20].

El patrón *Separación Modelo-Vista*, establece que los objetos modelo (dominio) no deberían conocer directamente los objetos vista (presentación) ni estar directamente acoplados a ellos. Guiándose por este patrón, en esta arquitectura reitera que los mensajes, que comunican a las capas de presentación y del dominio, surgen de peticiones que objetos de la vista solicitan a objetos del dominio y no de manera inversa. De esta manera, la capa de presentación se encarga solamente de la entrada y salida de los datos, sin que los conserven, y sin ofrecer directamente la funcionalidad de la capa del dominio. En síntesis, las clases de los paquetes de la capa de presentación tienen visibilidad respecto las clases de los paquetes de la capa del dominio y puede enviarles mensajes, pero la capa de dominio no conoce la existencia de la capa de presentación.

Reducir al mínimo el impacto que los cambios de la interfaz tienen en la capa del dominio; conectar fácilmente otras interfaces a la capa del dominio, sin que esto lo afecte; transportar fácilmente la capa del modelo hacia otras vistas, son las razones por las cuales se ha empleado, en el diseño de esta arquitectura, el patrón *Modelo-Vista*.

[20] Son objetos que requieren de almacenamiento perdurable.

Arquitectura del Software



Arquitectura del Hardware

Esta arquitectura describe a gran escala, la estructura de la tecnología usada para implementar el sistema y consiste en componentes de hardware. Su definición se basa en la tecnología con la que actualmente se cuenta

Para la representación gráfica de esta arquitectura se utilizó el siguiente diagrama de despliegue, en donde cada caja representa un nodo y la línea representa la liga que existe entre estos nodos. La multiplicidad entre ambos nodos es de 1 a muchos, representada por el número uno y el asterisco, y se lee que un servidor atenderá las peticiones de 1 o varios clientes.

La maquina que funcione como servidor, es una estación de trabajo SUN Sparc, y los clientes que accedan al sistema, serán computadoras conectadas a Internet y capaces de ejecutar un navegador de páginas web. En relación a los procesos, estos se llevan a cabo en el servidor y envía los resultados en forma de páginas web, al browser del cliente.

Diagrama de despliegue

Almacenamiento de objetos persistentes

La necesidad de almacenar y recuperar alguno de los objetos que formarán parte de este sistema, hace imprescindible la toma de decisiones en relación al mecanismo de almacenamiento que se utilizará.

El diseño de base de datos relacionales tiene algunas semejanzas con el desarrollo orientado a objetos, tales como, que su diseño suele llevarse a cabo a través de un proceso incremental e iterativo; en ambos desarrollos los diseñadores, constantemente, se encuentran haciendo elecciones de manera paralela, entre el diseño lógico y el físico, una vez que se aproxima la implementación; a su vez la forma en que se describen los elementos en una base de datos es muy parecida a la forma en que se describen los objetos en una aplicación que utilice la metodología orientada a objetos. Asimismo el hacer uso de la tecnología de los sistemas manejadores de bases de datos relacionales dentro de una arquitectura orientada a objetos reduce el riesgo de desarrollo, ya que dicha tecnología está más madura y se encuentra disponible en distintas plataformas. Tomando en cuenta estos aspectos y considerando que actualmente el cliente ya tiene cuenta con personal y herramientas para el soporte de un modelo relacional, se ha decidido que para el desarrollo de este sistema se utilizará una arquitectura híbrida que se construirá con una cubierta orienta a objetos sobre una base de datos relacional tradicional, tomando ventaja de ambos paradigmas.

Conversión a tablas.

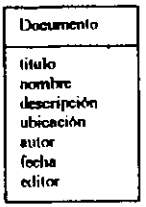
Se ha utilizado el patrón Representación de Objetos como Tablas, que propone definir una tabla a partir de los atributos de una clase, y a esta actividad se le conoce como mapeo. En este caso el mapeo se realizó de manera horizontal, lo que significa que cada atributo de una clase será una columna perteneciente a una tabla cuyo nombre sea el de la clase que se está mapeando.

Los objetos persistentes del sistema son:

- Documentos
- Evento
- Foro de discusión
- Mensaje
- Noticia
- Pregunta
- Proyecto
- Reporte
- Respuesta
- Colaborador[21]

[21] Presenta un caso de generalización por lo que el mapeo es distinto a los anteriores.

Modelo de clases

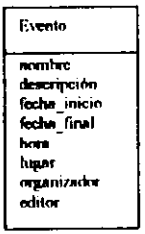


Modelo de tablas

Nombre columna	Tipo	Nulo
id_documento	int	no
nombre	string	no
descripción	string	no
ubicación	string	no
autor	string	si
fecha	date	si
id_editor	int	no

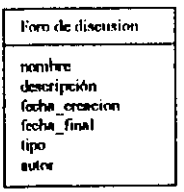
Código SQL

```
Create table Documento(
  id_ documento int not null,
  nombre string not null,
  descripción string not null,
  ubicación string not null,
  autor string,
  fecha date,
  id_editor int not null)
```



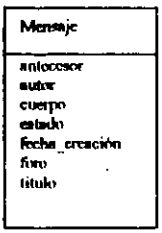
Nombre columna	Tipo	Nulo
id_evento	int	no
nombre	string	no
descripción	string	no
fecha_inicio	date	no
fecha_final	date	si
hora	time	si
lugar	string	no
organizador	string	si
id_editor	int	no

```
Create table Evento(
  id_ evento int not null,
  nombre string not null,
  descripción string not null,
  fecha_inicio date not null,
  fecha_final  date,
  hora time,
  lugar string not null,
  organizador,
  id_editor int not null)
```



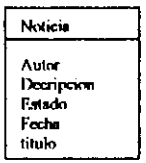
Nombre columna	Tipo	Nulo
id_foro	int	no
nombre	string	no
descripción	string	no
fecha_creacion	date	si
tipo	string	no
id_editor	int	no

```
Create table Foro(
  id_ foro int not null,
  nombre string not null,
  descripción string not null,
  fecha_creacion date,
  tipo string not null,
  id_editor int not null)
```



Nombre columna	Tipo	Nulo
id_mensaje	int	no
id_antecesor	int	no
id_autor	int	no
cuerpo	string	no
estado	string	no
fecha_creación	date	si
id_foro	int	no
titulo	string	no

```
Create table Mensaje(
  id_ mensaje int not null,
  id_antecesor int not null,
  autor int not null,
  cuerpo string not null,
  estado string not null,
  fecha_creacion date,
  id_foro int not null,
  titulo string not null)
```



Nombre columna	Tipo	Nulo
id_noticia	int	no
descripcion	string	no
estado	string	no
fecha	date	no
titulo	string	no

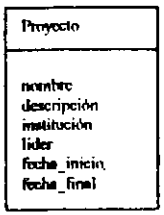
```
Create table Noticia(
  Id_noticia int not null,
  Descripcion string not null,
  Estado string not null,
   Fecha date not null,
  Titulo string not null)
```

Nombre columna	Tipo	Nulo
id_pregunta	int	no
titulo	string	no

```
Create table Pregunta(
  id_ pregunta int not null,
 titulo string not null)
```

Nombre columna	Tipo	Nulo
id_respuesta	int	no
nombre	string	no
id_pregunta	int	No
puntos	int	si

```
Create table  Respuesta(
  id_respuesta int not null,
  nombre string not null,
  id_pregunta int not null,
  puntos int)
```



Nombre columna	Tipo	Nulo
id_proyecto	int	no
nombre	string	no
descripcion	string	no
institucion	string	si
id_lider	int	no
fecha_inicio	date	no
fecha_final	date	si

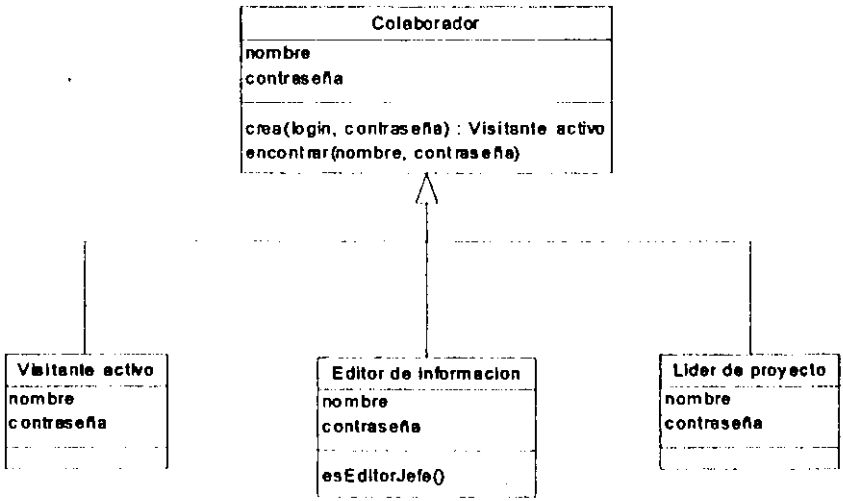
```
Create table Proyecto(
   id_proyecto int not null,
   nombre string not null,
   descripcion string not null,
   institucion string not null,
   id_lider int not null,
   fecha_inicio date not null,
    fecha_final date)
```

Nombre columna	Tipo	Nulo
id_reporte	int	no
descripcion	string	no
encabezado	string	no
fecha	date	no

```
Create table Reporte(
   id_ reporte int not null,
   descripcion string not null,
   encabezado string not null,
   fecha date not null)
```

Mapeo del caso de generalización

Para modelar la generalización que se presenta en este caso, se decidió por utilizar una sola tabla que diferenciara los tipos por medio de un campo cuyo nombre es *tipo*.



Modelo de tablas

Nombre columna	Tipo	Nulo
id_colaborador	int	no
nombre	string	no
contraseña	string	no
tipo[22]	string	no

Código SQL

```
Create table colaborador(
    id_colaborador int not null,
    nombre string not null,
    contraseña string not null,
     tipo string not null)
```

[22] Tipo={visitante_activo | lider_de_proyecto | editor | editor_jefe}

Hacia la implementación...

En esta última parte se muestran las herramientas de software y hardware necesarias para la construcción y funcionamiento del sistema. Para tomar decisiones en este aspecto, fue necesario considerar los siguientes puntos:

1. Se dispone de una máquina modelo SUN Ultra 5 Workstation, destinada a dar alojamiento al sistema.
2. Se requiere que el servidor que tenga instalados, un servidor web, un manejador de bases de datos relacionales y un interprete o compilador de un lenguaje de programación que soporte el paradigma orientado a objetos.
3. Dado que el solicitar licencias de software implica dinero y tiempo, se optó por la utilización de software libre para la implementación del sistema.
4. Es necesaria conexión con la red Internet.

También se tomaron en cuenta las arquitecturas del hardware y del software, propuestas en anteriores páginas de este diseño, para finalmente llegar a las siguientes decisiones:

Características del servidor

Modelo: Ultra 5 Workstation
Procesador: UltraSPARC-IIi
Número de procesadores: 1
Velocidad: 400 Mhz
Capacidad de memoria: RAM 128 MB
Capacidad de disco duro: 20 GB
Número de discos duros: 1
Sistema operativo: Solaris versión 7
Servidor web: Apache Server versión 1.3.19
Manejador de bases de datos relacionales: Mysql versión 3.29
Interprete o compilador del lenguaje de programación: PHP
Conexión a Internet

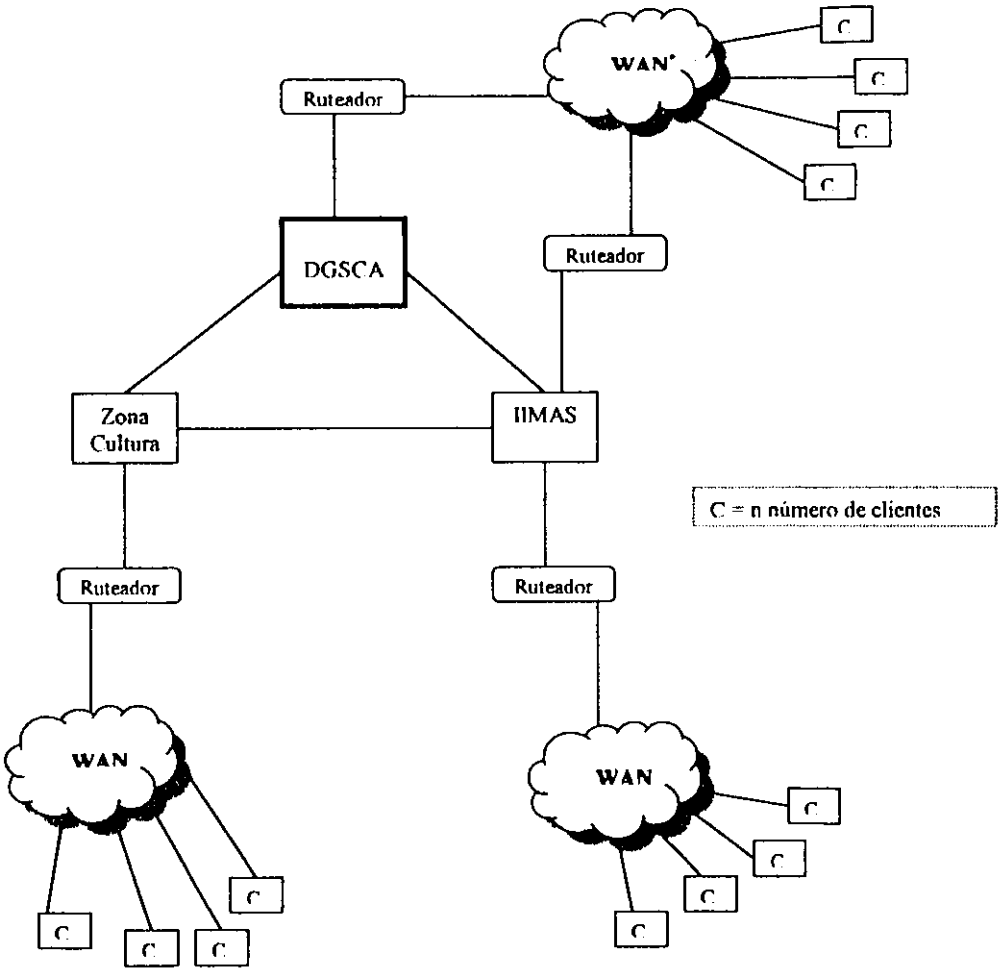
Características del cliente

Una computadora que tenga instalado un visualizador de páginas web.
Conexión a Internet

La conexión a Internet del servidor

El servidor donde residirá el sistema, se encuentra ubicado dentro de las instalaciones de la DGSCA en ciudad Universitaria. A partir de este punto serán enviadas páginas web hacia diversos clientes conectados a Internet.

El siguiente diagrama muestra de manera general la conexión del servidor a Internet. Existen tres puntos básicos, DGSCA, el IIMAS[23] y la zona cultural, conectados a ruteadores que permiten el vínculo entre la red de la UNAM y el resto del mundo conectado a Internet.



[23] Instituto de Investigaciones de Matemáticas Aplicadas en Sistemas

* Red de área amplia, por sus siglas Wide Area Network

CONCLUSIONES

- Aunque en nuestros días la tecnología esta cada vez más al alcance de muchas personas, en Internet se presenta a menudo la dependencia de los interesados en publicar ciertos contenidos de información con las personas con los conocimientos necesarios para que dichos contenidos se publiquen.

- En este trabajo se propuso el diseño de un sistema que rompiera justamente con esa dependencia, ya que los expertos de la información tienen a su alcance el medio para poder publicarla; los interesados y no necesariamente expertos, tienen un espacio para dar a conocer sus opiniones y de esta manera sugerir la dirección de las publicaciones. Por otra parte existe el usuario que, accidentalmente o no, visita el sitio y posiblemente esta visita resulte el inicio de una comunicación e intercambios de ideas.

- En la introducción de este documento se habla someramente de los medios de comunicación y de la poca participación de los receptores. Con la propuesta de este sistema se persigue incrementar la participación de los visitantes del sitio y ya que se utiliza Internet como medio de comunicación y las probabilidades de conseguirlo son mayores.

- Las personas con conocimientos técnicos, realmente se preocuparán por que las herramientas que se utilizan para publicar información en el sitio funcionen adecuadamente, ya que a final de cuentas por la información y su presentación se preocuparán las personas que realmente conozcan del tema, es decir que, tanto investigadores, desarrolladores, consultores y gente involucrada en los proyectos de Internet2 serán los indicados para iniciar a alimentar este sitio.

- Ahora la responsabilidad de los contenidos de información que se presentan en el sitio, no es de un intermediario o de alguien que conoce de programación para el web, sino de personas que desean que se publique porque conocen de ella. Para lograr esto se puso especial cuidado en analizar los procesos utilizados y en los que se podrían proponer. Era definitivamente importante mantener una comunicación clara que fuera entendida por todos los involucrados y para ello, la utilización de un lenguaje de modelado y una metodología orientada a objetos, facilitó lograr esta meta.

- El sistema aquí propuesto presenta una funcionalidad que no está definida por el tipo de información que maneja, sino por el tipo de procesos que automatiza y que en muchos casos son muy similares a los que se presentan en la actividad de administrar y actualizar datos en los sitios de Internet de diversas áreas como son la científica (como es este caso), educativa y comercial, entre otras. En otras palabras, este sistema es reutilizable y la consecuencia más benéfica de esto, será un ahorro en el tiempo dedicado a la búsqueda de posibles soluciones.

REFERENCIAS

Bibliografía

LARMAN, Craig UML y Patrones (Introducción al análisis y diseño orientado a objetos). México. 2000. Prentice Hall

OLSEN, Andy Object-Oriented Análisis and Design Using the Unified Modeling Language (UML) 1998. Oracle Corporation.

Sitios en internet

http://www.internet2.edu/

http://www.internet2.edu.mx/

http://websearch.about.com/internet/websearch/library/weekly/aa050799.htm

http://iamwww.unibe.ch/~scg/OOinfo/FAQ/

http://www.joopmag.com/html/from_pages/article.asp?id=189

http://www.cyberdyne-object-sys.com/oofaq2/body/general.htm#S3.7

http://www.hsr.ch/div/Booch/BoochReference/

Disco compacto

Inside the Unified Modeling Language. Creado por Rational Company en 1999.