

1.1. INTRODUCCIÓN

Internet es considerada como una “Red de Redes” porque está hecha a base de unir muchas redes locales de ordenadores, por esta red mundial circulan constantemente cantidades increíbles de información, por este motivo se le llama también “La Autopista de la Información”, hay millones de personas que navegan por Internet en todo el mundo. Uno de los servicios más importantes y más usados de Internet en la actualidad es el World Wide Web, éste ha evolucionado a pasos agigantados, primero incorporando en el contenido de las paginas diferentes tipos de letras, imágenes, sonidos, etc. y posteriormente, por medio del CGI, lenguajes script y otros métodos, el usuario interactúa cada vez más con el servidor Web, desde el envío de opiniones hasta el acceso y administración de grandes bases de datos.

Es una verdad que la información se ha convertido en un activo importante de las empresas e instituciones. No se conciben las actividades de la vida moderna sin información y ésta no sólo es importante en su cantidad y calidad sino en la rapidez con la que se obtiene. No sería posible obtener actualmente información con las características requeridas si no se contara con la tecnología informática representada principalmente con las computadoras y los programas que las hacen funcionar.

Con los avances informáticos de hoy, otra realidad que no se puede soslayar es el hecho de que en la administración pública de nuestro país es difícil encontrar sistemas de información de gran alcance o tan elaborados como los que existen en el sector privado. Casi cualquier empresa privada de regular tamaño cuenta con su sistema de información que automatiza sus funciones principales, controla sus operaciones y apoya sus decisiones. La Web se ha convertido en un escaparate de alcance mundial, donde las organizaciones pueden mostrar sus objetivos, funciones y servicios a prácticamente cualquier persona en el mundo a un costo muy bajo.

Esta realidad se hace evidente en el sector de salud pública de nuestro departamento, puesto que los ciudadanos se encuentran pobremente informados sobre los requisitos, beneficios y prestaciones a los cuales tienen acceso. Así mismo la administración de

los hospitales aún hace uso de precarios servicios de comunicación para hacer llegar su documentación e informes entre la red de hospitales del departamento.

Es así que para dar solución a la problemática planteada hacer uso de la tecnología informática disponible en nuestro medio, tanto para el modelado como para la implementación del proyecto. Para el modelado de sistema se utilizó la metodología de desarrollo “Proceso Unificado” y el lenguaje de modelado Unificado UML para diseñar una solución acorde a las necesidades del Hospital San Juan de Dios de Entre Ríos. Si bien el alcance del proyecto no abarca la implementación del sistema, la propuesta fue diseñada tomando en cuenta el paradigma orientado a objetos, arquitectura cliente-servidor, el patrón vista-controlador y conceptos fundamentales de base de datos, para asegurar una arquitectura lo suficientemente robusta como para soportar los requerimientos del sistema.

La solución planteada consiste en modelar un **“Sistema Web de Información aplicado al Hospital San Juan de Dios de Entre Ríos”**, mediante el uso de una arquitectura, paradigmas y herramientas involucradas en el desarrollo de aplicaciones Web. Con la finalidad de apoyar las funciones que tiene este sector, mejorando así la comunicación, rendimiento y uso de la información, ampliando la capacidad del procesamiento y de recuperación de la información, así como un mayor control en cuanto a la integridad, confidencialidad y disponibilidad de los datos.

1.2. ANTECEDENTES

La satisfacción de los pacientes y trabajadores con los servicios de salud constituyen un elemento importante, ya que contribuye a la propia salud individual dado que la persona satisfecha tiene más probabilidades de participar en la atención de una forma más efectiva, hablamos de la calidad de la atención recibida y sus resultados.

En el proceso de perfeccionamiento del Sistema Nacional de Salud que enfrenta nuestro país en los últimos años y con el fin de mantener los avances alcanzados en la salud de la población se pretende unir todos los esfuerzos y recursos para ir mejorando la calidad de los servicios que se ofrecen teniendo en cuenta que los

servicios de salud deben tener una orientación a la población que logre la participación social óptima alcanzando la promoción, conservación y recuperación de la salud de la población.

El Sector Salud ha implementado diversos sistemas Web, destinados a difundir información y promocionar los servicios que ofrece a la población en general, mencionaremos a continuación algunos de estos sistemas y una breve descripción de sus funcionalidades:

SIGEH (<http://www.cossmil.gov.bo>) Sistema que permite el control y Gestión de los Asegurados y beneficiarios del Hospital Militar Central.

SIR (<http://www.rib.gov.bo>) El Sistema de Intercomunicación de Riberalta fue creado con el fin de apoyar a resolver los problemas de comunicación entre las unidades de Salud de Riberalta.

1.3. PLANTEAMIENTO DEL PROBLEMA

En los últimos años ha evidenciado un crecimiento en el volumen de información que se genera en el Hospital San Juan de Dios de Entre Ríos, debido al crecimiento poblacional, a los programas y las prestaciones que éste ofrece, sin embargo el tratamiento de dicha información es manual, produciendo como resultado retardo en las búsquedas de información, sistemas de comunicación precarios y desinformación en la población, lo cual pone en manifiesto la necesidad de contar con un sistema de información dinámico que se ajuste a los requerimientos del centro de salud de Entre Ríos y de la población en general.

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

Analizar y Diseñar un “**Sistema Web de Información para el Hospital San Juan de Dios de Entre Ríos**” que ofrezca información acerca de los servicios y programas sociales existentes y que establezca un vínculo de comunicación entre centro de salud

y la población, a través de la aplicación de metodologías, paradigmas, estándares y herramientas involucradas en el desarrollo de aplicaciones Web.

1.4.2. OBJETIVOS ESPECÍFICOS

- Modelar los requisitos del Sistema para identificar las necesidades reales de la institución, haciendo uso del modelo de casos de uso del lenguaje de modelado unificado.
- Realizar el análisis y diseño del sistema de acuerdo a las actividades especificadas por el proceso de desarrollo de software Proceso Unificado (UP) en sus fases de inicio y elaboración para garantizar una calidad aceptable del producto.
- Modelar los flujos de trabajos fundamentales de UP, utilizando los diagramas de UML Lenguaje Unificado de Modelado para describir y especificar el software.
- Diseñar una interfaz amigable e intuitiva, a través de la implementación de principios de usabilidad, accesibilidad y diseño de la experiencia del usuario
- Diseñar una Base de Datos aplicando el modelo relacional para la creación y acceso a la misma, que garantice la integridad y seguridad de los datos.
- Diseñar una arquitectura de software robusta capaz de soportar los requerimientos del sistema, aplicando el paradigma orientado a objetos y la arquitectura cliente servidor.

1.5. JUSTIFICACIÓN

1.5.1. JUSTIFICACIÓN ACADÉMICA

El desarrollo del Sector Salud está muy ligado al desarrollo de la informática. El manejo de la información de modo eficiente constituye una de las principales preocupaciones dentro de los diferentes programas y servicios de salud.

Por ese motivo es importante garantizar la integridad, disponibilidad y privacidad del volumen de información que se maneja, ya que de ello podría depender, en gran medida, la contribución que pueda hacer este sector al desarrollo de la sociedad.

El motivo fundamental que me lleva a realizar el “**Sistema Web de Información para el Hospital San Juan de Dios de Entre Ríos**”, radica en que los mecanismos actuales de información y comunicación no se ajustan a los requerimientos del Hospital y de la población.

La investigación realizada sobre metodologías de análisis, diseño, estándares de diseño de interfaces y políticas de seguridad que se utilizan en un sistema Web de estas características puede servir de apoyo y guía para los futuros profesionales.

1.5.2. JUSTIFICACIÓN TECNOLÓGICA

La justificación tecnológica radica en la existencia de herramientas de Software Libre que ofrece muchos beneficios, entre ellos el acceso al código fuente, por tanto su funcionamiento está bajo el escrutinio de toda una comunidad de usuarios alrededor del mundo; si se detecta algún error o problema de seguridad en algún programa, éstos son corregidos en un tiempo muy corto; y los requerimientos de Hardware son relativamente bajos. Además de la libertad y flexibilidad que se logra usando Software Libre, otro factor de peso que orienta a la elección de este tipo de software es que la mayoría de programas desarrollados bajo este paradigma son gratuitos, por lo que al momento de realizar la instalación de ellos en una computadora no hay que preocuparse por los costos de licencia de uso.

Debido a que la informática y las comunicaciones están relacionadas, los países en general van evolucionando. Tecnológicamente Bolivia no se queda al margen, existiendo cada vez más los medios para dotar al Sector Salud de equipos tecnológicos. Los cuales son los medios necesarios para el desarrollo del “**Sistema Web de Información para el Hospital San Juan de Dios de Entre Ríos**”.

1.5.3. JUSTIFICACIÓN SOCIAL

El “**Sistema Web de Información para el Hospital San Juan de Dios de Entre Ríos**”, es justificable socialmente en gran medida porque beneficiará directamente a las distintas dependencias del Hospital y a las personas que requieran un servicio.

1.6. ALCANCE DEL TRABAJO (VISTA DEL ADMINISTRADOR Y VISTA DEL USUARIO)

- Analizar y diseñar los procesos que componen el Sistema.
- Diseñar el modelo navegacional.
- Diseñar la Base de Datos.
- Diseñar interfaces.
- Diseñar el modelo lógico.
- Diseñar el modelo de datos.

2.1. CONCEPTOS, TECNOLOGÍAS Y HERRAMIENTAS NECESARIAS

2.1.1. INTERNET

Internet es considerada como una “Red de Redes” porque está hecha a base de unir muchas redes locales de ordenadores, por ésta red mundial circulan constantemente cantidades increíbles de información, por este motivo se le llama también “La Autopista de la Información” Hay millones de personas que "navegan" por Internet en todo el Mundo.

Uno de los servicios más importantes y más usados de Internet en la actualidad es el World Wide Web, éste ha evolucionado a pasos agigantados, primero incorporando en el contenido de las páginas diferentes tipos de letras, imágenes, sonidos etc. y posteriormente, por medio de lenguajes script y otros métodos, el usuario interactúa cada vez más con el servidor Web, desde el envío de opiniones hasta el acceso y administración de grandes bases de datos.

2.1.2. WEB AUTOGESTIONABLE

Con un Sitio Web Autogestionable ofrece la información siempre actualizada sin depender de terceros, con lo que la Web está siempre actualizada, ofreciendo la opción de modificar el contenido existente actualmente, añadir nuevo contenido e incluso eliminarlo. A través de la administración se permite gestionar los contenidos: registrar, modificar eliminar sin depender de ningún servicio de mantenimiento.

Características más destacadas de la Web Autogestionable:

- Administración protegida con nombre de usuario y contraseña.
- Modifica, borra o añade contenidos a la web cuándo y dónde sea requerido.
- Diseño único y exclusivo realizado para la Web, permitiendo la personalización a través de temas CSS.
- Generación de páginas web dinámicas, que se generan con información almacenada en la base de datos.

2.1.3. PÁGINA WEB DINÁMICA ¹

Se conoce con el nombre de página web dinámica a aquella, cuyo contenido se genera a partir de lo que un usuario introduce en un web o formulario. El contenido de la página no está incluido en un archivo HTML como en el caso de las páginas web estáticas.

Una página Web dinámica tiene las siguientes características:

- Gran número de posibilidades en su diseño y desarrollo.
- El visitante puede alterar el diseño, contenidos o presentación de la página a su gusto.
- En su realización se utilizan diversos lenguajes y técnicas de programación.
- El proceso de actualización es sumamente sencillo, sin necesidad de entrar en el servidor.
- Permite un gran número de funcionalidades tales como bases de datos, foros, contenido dinámico, etc.
- Pueden realizarse íntegramente con software de libre distribución.
- Existe una amplia comunidad de programadores que brinda apoyo desinteresado.
- Cuenta con un gran número de soluciones prediseñadas de libre disposición.

2.1.4. USABILIDAD

2.1.4.1. CONCEPTO ²

La usabilidad (dentro del campo del desarrollo web) es la disciplina que estudia la forma de diseñar sitios web para que los usuarios puedan interactuar con ellos de la forma más fácil, cómoda e intuitiva posible.

¹ <http://www.manuales.dgsca.unam.mx/webdina/concepto.htm>

² http://www.nosolousabilidad.com/articulos/introduccion_usabilidad.htm

La mejor forma de crear un sitio web usable es realizando un diseño centrado en el usuario, diseñando para y por el usuario, en contraposición a lo que podría ser un diseño centrado en la tecnología o uno centrado en la creatividad u originalidad.

Alrededor del concepto de usabilidad existe una gran cantidad de conceptos relacionados semánticamente, que a veces son confundidos entre sí.

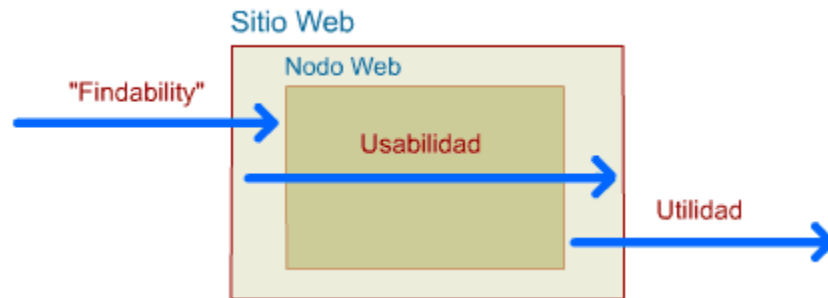


Figura 1: Usabilidad

Findability ("recuperabilidad", "encontrabilidad") es la posibilidad de que determinada información sea encontrada, localizada, o recuperada. O lo que es lo mismo, la "accesibilidad" de un sitio o de un nodo web (no confundir con la Accesibilidad de los contenidos). En la "findability" intervienen el posicionamiento en motores e índices de búsqueda, la predisponibilidad de nuestro sitio a ser recuperado por motores de búsqueda (search engine optimization), la promoción que hayamos realizado (por correo electrónico, concursos, banners), etc.

Pero la "findability" también abarca los medios de que disponga nuestro sitio web para encontrar información dentro de este, es decir, buscadores locales, correcta arquitectura de la información y otros.

La Usabilidad de un sitio web está estrechamente relacionada con la "findability", ya que cuanto más sencillo resulte para el usuario navegar por un sitio web, mayores posibilidades tendrán de encontrar lo que busca.

Otro concepto íntimamente relacionado con la Usabilidad es el de la Accesibilidad, cuyo objetivo es conseguir que el diseño del sitio permita que personas discapacitadas también puedan acceder a sus contenidos. Así mismo, otro requisito para que una web

sea accesible es que cualquier persona pueda acceder a sus contenidos independientemente del dispositivo que use (hardware o software).

2.1.4.2. EVALUACIÓN DE LA USABILIDAD

Evaluar la usabilidad de nuestro sitio web puede ser algo verdaderamente útil, ya que descubrir qué errores de diseño tiene nuestra web es el primer paso para poder corregirlos. En cuanto a en qué momento del proyecto es más recomendable evaluar el sitio web, se debe seguir la siguiente regla: Cuanto más tarde peor, ya que será más costoso rediseñar todo un sitio ya acabado, que reconducir la línea de desarrollo por mejores caminos. Hay varias formas de evaluación:

- **Expertos**

Una opción bastante recomendable es encargar a un experto que evalúe (evaluación por criterios o heurística) nuestra web. En Internet hay muchos profesionales independientes y empresas que se dedican a esta tarea.

- **Encuestas**

Otra opción es utilizar encuestas para comprobar la usabilidad de nuestro sitio web. La encuesta debería ser diseñada por un experto y realizada sobre usuarios actuales o potenciales de nuestro sitio web. Este sistema, en realidad, no es de los más útiles.

- **Pruebas de usabilidad**

Una prueba o test de usabilidad es contar con unos 5 usuarios en una sala, que realizarán una navegación "asistida" por el sitio web a probar. El encargado de la prueba tomará nota de qué problemas encuentran los usuarios para realizar las tareas que se les hayan indicado, y así conocer qué errores de diseño tiene el sitio web.

- **Otras técnicas y herramientas**

Existen otras técnicas más complejas que se sirven de herramientas automatizadas para evaluar la usabilidad. Una de estas herramientas son los sistemas de seguimiento visual (eye tracking).

Otra herramienta es el software especializado en análisis de ficheros "log" de visitas, ficheros que nos proporcionan información sobre qué visitan nuestros usuarios y qué camino han seguido en su navegación. Este tipo de análisis no sólo nos será útil desde el punto de vista de la usabilidad.

2.1.5. TEORÍA DEL COLOR

2.1.5.1. CONCEPTO ³

Los colores de los objetos nos transmiten emociones muy fuertes, pese a que, generalmente no nos damos cuenta de ello. Los hospitales carecen de color rojo, pues éste es asociado a la sangre, y en las oficinas suele predominar el blanco que denota orden y pulcritud (en las BOFH Room predomina el color verde placa base). Por ejemplo trabajar en un lugar con paredes rojas, sería demasiado estresante, pues es un color de un fuerte impacto visual y que tiene mucha fuerza y energía. Esto son claros ejemplos de como en la actualidad estamos usando los colores para nuestro beneficio, para trabajar mejor, para estar más tranquilos, etc. Algunas de las cosas más importantes que debemos saber del color son:

- El color y su expansión
- Armonía y contraste
- Significado del color

2.1.5.2. EL COLOR Y SU EXPANSIÓN

Probablemente alguna vez os hayáis preguntado por qué la mayoría de las páginas web que hay en internet tienen un fondo claro y no negro.

³ <http://www.cristalab.com/tutoriales/disenio-web-fundamentos-basicos-c63l/>

De hecho, las páginas que tienen un fondo negro suelen quedar peor. Esto se debe al carácter expansivo de los colores, no se verá igual un texto sobre fondo negro que blanco, el texto sobre fondo blanco es más legible y por lo tanto el lector debe forzar menos la vista, sintiéndose agradecido. Esto también lo podemos ver en colores claros y oscuros, no es algo que suceda únicamente en blanco y negro.

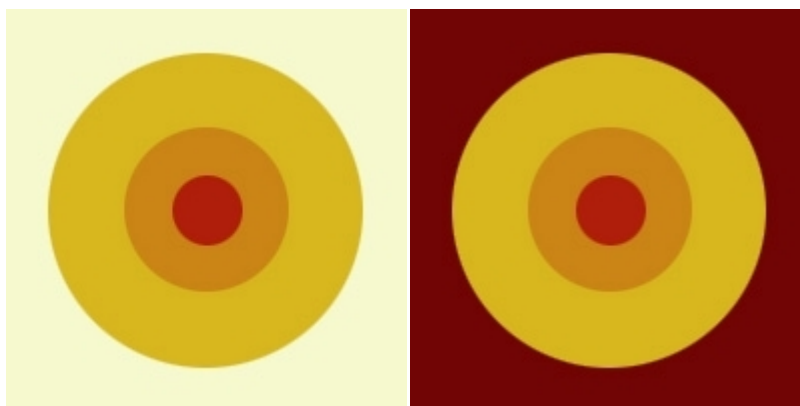


Figura 2: Expansión del Color

En estos dos dibujos parece que el de la izquierda tienda a expandirse y a hacerse más grande, todo lo contrario que en el de la derecha. Eso es debido a que el color oscuro comprime la figura del medio. Estas técnicas las usaremos cuando queramos que una parte de nuestra web quede cerrada o abierta.

2.1.5.3. ARMONÍA Y CONTRASTE

Armonía: Crear una gama de colores para nuestra web que esté compuesta por colores de la misma gama o tono.



Figura 3: Armonía del Color

Contraste: Combinar diferentes colores para crear una gama como, por ejemplo, claros y oscuros, cálidos y fríos, etc.



Figura 4: Contraste del Color

En esta imagen cada casilla nos está expresando una idea diferente.

2.1.5.4. SIGNIFICADO DEL COLOR

El color tiene su significado, aunque estos significados varían un poco según las culturas, ya que por tradición es posible que el amarillo signifique mala suerte en algunas partes y en otros no. Sin embargo existen estándares que de manera general describen el significado de los colores:

- **Blanco**

Se halla en el extremo de la gama de los grises. Es un color latente por su capacidad de potenciar los otros colores vecinos. El blanco puede expresar paz, soleado, feliz, activo, puro e inocente. El blanco es el fondo universal de la comunicación gráfica.

- **Negro**

Al igual que el blanco, también se encuentra en el extremo de la gama de grises. Es el símbolo del silencio, del misterio y, en ocasiones, puede significar impuro y maligno. Confiere nobleza y elegancia, sobre todo cuando es brillante.

- **Gris**

Simboliza la indecisión y la ausencia de energía, expresa duda y melancolía. Los colores metálicos tienen una imagen lustrosa, adoptando las cualidades de los metales que representan. Dan impresión de frialdad metálica, pero

también dan sensación de brillantez, lujo, elegancia, por su asociación con los metales preciosos.

- **Amarillo**

Es el color más luminoso, más cálido, ardiente y expansivo. Es el color del sol, de la luz y del oro, y como tal es violento, intenso y agudo. Suelen interpretarse como animados, joviales, excitantes, afectivos e impulsivos. No es recomendable usarlo como color principal de nuestra página, pues tiene demasiada fuerza y tiende a cansar al visitante.

- **Naranja**

Posee fuerza activa, radiante y expansiva. Tiene un carácter acogedor, cálido, estimulante y una cualidad dinámica muy positiva y energética.

- **Rojo**

Significa la vitalidad, es el color de la sangre, de la pasión, de la fuerza bruta y del fuego. Está ligado al principio de la vida, expresa la sensualidad, la virilidad, la energía; es exultante y agresivo. El rojo es el símbolo de la pasión ardiente y desbordada, de la sexualidad y el erotismo.

- **Azul**

Es el símbolo de la profundidad. Inmaterial y frío, suscita una predisposición favorable. La sensación de placidez que provoca el azul es distinta de la calma o reposo terrestres, propios del verde. Expresa armonía, amistad, fidelidad, serenidad, sosiego... y posee la virtud de crear la ilusión óptica de retroceder. Este color se asocia con el cielo, el mar y el aire. El azul claro puede sugerir optimismo. Cuanto más se clarifica más pierde atracción y se vuelve indiferente y vacío. Cuanto más se oscurece más atrae hacia el infinito.

- **Violeta**

Es el color de la templanza, de la lucidez y de la reflexión. Es místico, melancólico y podría representar también la introversión. En sus tonos más claros se vuelve un poco triste, en los más oscuros representa grandeza.

- **Verde**

Es el color más tranquilo y sedante. Evoca la vegetación, el frescor y la naturaleza. Es el color de la calma indiferente: no transmite alegría, tristeza o pasión. Cuando algo reverdece suscita la esperanza de una vida renovada.

- **Marrón**

Es un color masculino, severo, confortable. Es evocador del ambiente local otoñal y da la impresión de gravedad y equilibrio. Es el color realista, tal vez porque es el color de la tierra que pisamos.

2.1.6. APLICACIONES Y SISTEMAS WEB⁴

En la ingeniería de software se denomina aplicación web o sistemas web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los webmails, wikis, weblogs, tiendas en línea y la propia Wikipedia que son ejemplos bien conocidos de aplicaciones web.

⁴ http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web

Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

2.1.7. USUARIOS Y CONTRASEÑAS

Una contraseña o clave (en inglés password) es una forma de autenticación que utiliza información secreta para controlar el acceso hacia algún recurso. La contraseña normalmente debe mantenerse en secreto ante aquellos a quien no se le permite el acceso. Aquellos (usuarios) que desean acceder a la información se les solicitan una clave; si conocen o no conocen la contraseña, se concede o se niega el acceso a la información según sea el caso.

Para prevenir posibles vulnerabilidades en nuestro servidor, es una medida eficaz encriptar las contraseñas. Si se tiene un sistema de usuarios y queremos proteger las contraseñas para prevenir posibles vulnerabilidades en nuestro servidor, es una medida eficaz encriptar las contraseñas, de manera que si alguien puede acceder a ellas no pueda ver la contraseña si no su encriptación.

Los algoritmos recomendados para encriptar contraseñas son:

- MD5 es un algoritmo de generación de resúmenes de mensajes de longitud arbitraria que genera salidas (resúmenes o firmas) de 128 bits.
- SHA1 es otro algoritmo de generación de resúmenes de mensajes de longitud inferior a 2^{64} bits que genera salidas de 160 bits.

2.2. ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

Análisis y diseño orientado a objetos (ADOO) es un enfoque de la ingeniería de software que modela un sistema como un grupo de objetos que interactúan entre sí. Este enfoque representa un dominio en términos de conceptos compuestos por verbos y sustantivos, clasificados de acuerdo a su dependencia funcional.

En este método de análisis y diseño se crea un conjunto de modelos utilizando una notación acordada como, por ejemplo, el lenguaje unificado de modelado (UML). ADOO aplica técnicas de modelado de objetos para analizar los requerimientos para un contexto - por ejemplo, un sistema de negocio, un conjunto de módulos de software - y para diseñar una solución para mejorar los procesos involucrados. No está restringido al diseño de programas de computadora, sino que cubre sistemas enteros de distinto tipo. Las metodologías de análisis y diseño más modernas son casos de uso guiados a través de requerimientos, diseño, implementación, pruebas, y despliegue.

2.2.1. DEFINICIÓN DE OBJETO

Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos) los mismos que consecuentemente reaccionan a eventos.

Se corresponde con los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa). Es una instancia de a una clase.

2.2.2. DEFINICIÓN DE CLASE

Definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.

2.2.3. DEFINICIÓN MÉTODOS

Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

2.2.4. DEFINICIÓN PROPIEDAD Y ATRIBUTO

Contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.

2.2.5. CARACTERÍSTICAS DE PROGRAMACIÓN ORIENTADA A OBJETOS (POO)

2.2.5.1. ABSTRACCIÓN

Denota las características esenciales de un objeto, donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar *cómo* se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos y cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción. El proceso de abstracción permite seleccionar las características relevantes dentro de un conjunto e identificar comportamientos comunes para definir nuevos tipos de entidades en el mundo real. La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiere atacar.

2.2.5.2. ENCAPSULAMIENTO

Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción.

Esto permite aumentar la **cohesión** de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.

2.2.5.3. POLIMORFISMO

Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia produce el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama *asignación tardía* o *asignación dinámica*.

2.2.5.4. HERENCIA

Es la facilidad mediante la cual la clase D hereda en ella cada uno de los atributos y operaciones de C, como si esos atributos y operaciones hubiesen sido definidos por la misma D. Por lo tanto, puede usar los mismos métodos y variables públicas declaradas en C. Los componentes registrados como "privados" (private) también se heredan, pero como no pertenecen a la clase, se mantienen escondidos al programador y sólo pueden ser accedidos a través de otros métodos públicos.

2.3. MODELO DE DATOS

2.3.1. MODELO ENTIDAD RELACIÓN

El modelo ER está basado en una percepción del mundo real consistente en objetos básicos llamados entidades y de relación entre estos objetos. Se desarrolló para facilitar el diseño de base de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica completa de una base de datos.

El modelo E-R es extremadamente útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual. Debido a esta utilidad, muchas herramientas de diseño de bases de datos se basan en los conceptos del modelo E-R.

2.3.2. BASE DE DATOS RELACIONALES (EL MODELO RELACIONAL)

Es un Base de Datos relacional los datos se encuentran organizados y almacenados en tablas llamadas afinidades, cada una de las cuales es implementada como un registro. Cada tabla del modelo relacional es en dos dimensiones: filas y columnas, cada fila contiene datos que pertenecen a alguna cosa o parte de alguna cosa y cada columna contiene un atributo.

El objetivo del diseño de las bases de datos relacionales es la generación de un conjunto de esquemas relacionales que nos permita almacenar la información sin redundancia innecesaria, pero también nos permitan recuperar fácilmente esa información. Un enfoque es el diseño de esquemas que se hallen en una formas normales deseables hacen falta información adicional sobre la empresa real que se está modelando con las base de datos.

2.4. EL PATRÓN DE DISEÑO MODELO-VISTA-CONTROLADOR

Durante toda la década del setenta, SmallTalk y algunos otros lenguajes como Simula I, fueron construyendo gradualmente el paradigma de programación orientada a objetos y estableciendo conceptos tales como objetos, clases, encapsulación, herencia y polimorfismo. Si bien dichos lenguajes no son usados actualmente para implementar aplicaciones comerciales, los conceptos que dejaron en el mundo del desarrollo de software están vigentes en la actualidad y son la base de lenguajes modernos como C++, Java o C#.

SmallTalk también fue el primer lenguaje de programación que permitió diseñar interfaces de usuario con múltiples “ventanas” desplegadas en una misma pantalla, concepto que después fue aplicado por GEMS, Macintosh, X11, Windows y otras interfaces gráficas de usuario modernas. El concepto central detrás de las librerías de interfaz de usuario provistas por SmallTalk está basado en el patrón de diseño MVC, creado por el profesor Trygve Reenskaug.

MVC es un patrón de diseño que considera dividir una aplicación en tres módulos claramente identificables y con funcionalidad bien definida: El Modelo, las Vistas y el Controlador.

2.5. EL MODELO

El modelo es un conjunto de clases que representan la información del mundo real que el sistema debe procesar, así por ejemplo un sistema de administración de datos climatológicos tendrá un modelo que representará la temperatura, la humedad ambiental, el estado del tiempo esperado, etc. sin tomar en cuenta ni la forma en la que esa información va a ser mostrada ni los mecanismos que hacen que esos datos estén dentro del modelo, es decir, sin tener relación con ninguna otra entidad dentro de la aplicación.

El modelo desconoce la existencia de las vistas y del controlador. Ese enfoque suena interesante, pero en la práctica no es aplicable pues deben existir interfaces que permitan a los módulos comunicarse entre sí, por lo que SmallTalk sugiere que el modelo en realidad esté formado por dos sub módulos: El modelo del dominio y el modelo de la aplicación.

2.5.1. MODELO DEL DOMINIO

Se podría decir que el modelo del dominio (o el modelo propiamente dicho) es el conjunto de clases que un ingeniero de software modela al analizar el problema que desea resolver; así, pertenecerían al modelo del dominio: El cliente, la factura, la temperatura, la hora, etc. El modelo del dominio no debería tener relación con nada externo a la información que contiene.

2.5.1.1. MODELO DE LA APLICACIÓN

El modelo de la aplicación es un conjunto de clases que se relacionan con el modelo del dominio, que tienen conocimiento de las vistas y que implementan los mecanismos necesarios para notificar a éstas últimas sobre los cambios que se pudieren dar en el modelo del dominio.

El modelo de la aplicación es llamado también coordinador de la aplicación.

2.5.1.2. LAS VISTAS

Las vistas son el conjunto de clases que se encargan de mostrar al usuario la información contenida en el modelo. Una vista está asociada a un modelo, pudiendo existir varias vistas asociadas al mismo modelo; así por ejemplo, se puede tener una vista mostrando la hora del sistema como un reloj analógico y otra vista mostrando la misma información como un reloj digital.

Una vista obtiene del modelo solamente la información que necesita para desplegar y se actualiza cada vez que el modelo del dominio cambia por medio de notificaciones generadas por el modelo de la aplicación.

2.5.2. EL CONTROLADOR

El controlador es un objeto que se encarga de dirigir el flujo del control de la aplicación debido a mensajes externos, como datos introducidos por el usuario u opciones del menú seleccionadas por él. A partir de estos mensajes, el controlador se encarga de modificar el modelo o de abrir y cerrar vistas. El controlador tiene acceso al modelo y a las vistas, pero las vistas y el modelo no conocen de la existencia del controlador.

MVC es un patrón de diseño que considera dividir una aplicación en tres módulos claramente identificables y con funcionalidad bien definida: El Modelo, las Vistas y el Controlador.

2.5.3. EL MODELO

El modelo es un conjunto de clases que representan la información del mundo real que el sistema debe procesar, así por ejemplo un sistema de administración de datos climatológicos tendrá un modelo que representará la temperatura, la humedad ambiental, el estado del tiempo esperado, etc. sin tomar en cuenta ni la forma en la que esa información va a ser mostrada ni los mecanismos que hacen que esos datos

estén dentro del modelo, es decir, sin tener relación con ninguna otra entidad dentro de la aplicación.

2.6. RUP (PROCESO UNIFICADO DEL DESARROLLO DE SOFTWARE)⁵

En primer lugar, el Proceso Unificado es un proceso de desarrollo de software. Un *proceso* de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario de un sistema software (**Figura 5**). Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

El Proceso Unificado esta basado en *componentes*, lo cual quiere decir que el sistema software en construcción está formado por **componentes** software interconectados a través de **interfaces** bien definidas.

El Proceso Unificado utiliza el *Lenguaje Unificado de Modelado* (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software. De hecho, UML es una parte esencial del Proceso. No obstante, los verdaderos aspectos definitorios del Proceso Unificado se resumen en tres fases clave: Dirigido por casos de uso, Centrado en la arquitectura, Iterativo e incremental.

Esto es lo que hace único al Proceso Unificado. En las tres secciones siguientes describiremos estas tres fases claves. Después, daremos una breve visión general del proceso: su ciclo de vida, fases, versiones, iteraciones, flujos de trabajo, y artefactos.



Figura 5: Un Proceso de Desarrollo de Software

⁵ “El proceso unificado de desarrollo de software” Ivar Jacobson, Grady Booch, James Rumbaugh. Madrid, 2000.

2.6.1. EL PROCESO UNIFICADO ESTÁ DIRIGIDO POR CASOS DE USO

Un sistema software ve a la luz para dar servicio a sus usuarios. Por tanto, para construir un sistema con éxito debemos conocer lo que sus futuros usuarios necesitan y desean.

El termino *usuario* no solo hace referencia a usuarios humanos sino a otros sistemas. En este sentido, el término *usuario* representa alguien o algo (como otro sistema fuera del sistema en consideración) que interactúa con el sistema que estamos desarrollando. Un ejemplo de interacción sería una persona que utiliza un cajero automático. Él (o ella) inserta la tarjeta de plástico, responde a las preguntas que le hace la maquina en su pantalla, y recibe una suma de dinero. En respuesta a la tarjeta del usuario ya sus contestaciones, el sistema lleva a cabo una secuencia de **acciones** que proporcionan al usuario un resultado importante, en este caso, la retirada del efectivo.

Una interacción de este tipo es un **caso de uso**. Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales. Todos los casos de uso juntos constituyen el **modelo de casos de uso**, el cual describe la funcionalidad total del sistema. Puede decirse que una especificación funcional contesta a la pregunta: ¿Qué debe hacer el sistema? La estrategia de los casos de uso puede describirse añadiendo tres palabras al final de esta pregunta: *¿Para cada usuario?* Estas tres palabras albergan una implicación importante. Nos fuerzan a pensar en términos de importancia para el usuario y no sólo en términos de funciones que seria bueno tener. Sin embargo, los casos de uso no son solo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación, y prueba; esto es, *guían el proceso de desarrollo*. Basándose en el modelo de casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que llevan a cabo los casos de uso. Los desarrolladores revisan cada uno de los sucesivos modelos para que sean conformes al modelo de casos de uso. Los ingenieros de prueba prueban la implementación para garantizar que los componentes del modelo de implementación implementan correctamente los casos de uso. De este modo, los

casos de uso no sólo inician el proceso de desarrollo sino que le proporciona un hilo al conductor.

Dirigido por casos de uso quiere decir que el proceso de desarrollo sigue un hilo, avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso se especifican, se diseñan, y los casos de uso finales son la fuente a partir de la cual los ingenieros de prueba construyen sus casos de prueba.

Aunque es cierto que los casos de uso guían el proceso, no se desarrollan aisladamente. Se desarrollan a la vez que la arquitectura del sistema. Es decir, los casos de uso guían la arquitectura del sistema y la arquitectura del sistema influye en la selección de los casos de uso. Por tanto, tanto la arquitectura del sistema como los casos de uso maduran según avanza el ciclo de desarrollo.

2.6.2. EL PROCESO UNIFICADO ESTÁ CENTRADO EN LA ARQUITECTURA

El papel de la arquitectura software es parecido al papel que juega la arquitectura en la construcción de edificios. El edificio se contempla desde varios puntos de vista: estructura, servicios, conducción de la calefacción, fontanería, electricidad, etc. Esto permite a un constructor ver una imagen completa antes que comience la construcción. Análogamente, la arquitectura en un sistema software se describe mediante diferentes vistas del sistema en construcción.

El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, como las perciben los usuarios y los inversores, y se refleja en los caso de uso. Sin embargo, también se ve influida por otros muchos factores, como la plataforma en la que tiene que funcionar el software (arquitectura hardware, sistema operativo, sistema de gestión de base de datos, protocolos para comunicaciones en red), los bloques de construcción reutilizables de que se dispone (por ejemplo, rendimiento, fiabilidad). La arquitectura es una vista del diseño completo con las características más completas resaltadas, dejando los detalles de lado. Debido a que lo que es significativo depende

en parte de una valoración, que a su vez, se adquiere con la experiencia, el valor de una arquitectura depende de las personas que se hayan responsabilizado de su creación.

No obstante, el proceso ayuda al arquitecto a centrarse en los objetivos adecuados, como la comprensibilidad, la capacidad de adaptación al cambio, y la reutilización.

¿Cómo se relacionan los casos de uso y la arquitectura? Cada producto tiene tanto una función como una forma. Ninguna es suficiente por sí misma. Estas dos fuerzas deben equilibrarse para obtener un producto con éxito. En esta situación, la función corresponde a los casos de uso y la forma a la arquitectura. Debe haber interacción entre los casos de uso y la arquitectura. Es un problema del tipo “el huevo y la gallina”. Por un lado, los casos de uso deben encajar en la arquitectura cuando se llevan a cabo. Por otro lado, la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, ahora y en el futuro. En realidad, tanto la arquitectura como los casos de uso deben evolucionar en paralelo.

Por tanto los arquitectos moldean el sistema para darle una *forma*. Es esta forma, la arquitectura, la que debe diseñarse para permitir que el sistema evolucione, no sólo en su desarrollo inicial, sino también a lo largo de las futuras generaciones. Para encontrar esa forma, los arquitectos deben trabajar sobre la comprensión general de las funciones clave, es decir, sobre los casos de uso claves del sistema. Estos casos de uso clave pueden suponer solamente entre el 5 y el 10 por ciento de todos los casos de uso, pero son los significativos, los que constituyen las funciones fundamentales del sistema.

De manera resumida, podemos decir que el arquitecto:

- Crea un esquema en borrador de la arquitectura, comenzando por la parte de la arquitectura que no es específica de los casos de uso (por ejemplo, la plataforma). Aunque esta parte de la arquitectura es independiente de los casos de uso, el arquitecto debe poseer una comprensión general de los casos de uso antes de comenzar la creación del esquema arquitectónico.

- A continuación, el arquitecto trabaja con un subconjunto de los casos de uso especificados, con aquellos que representen las funciones clave del sistema en desarrollo. Cada caso de uso seleccionado se especifica en detalle y se realiza en términos de subsistemas, clases y componentes.
- A medida que los casos de uso se especifican y maduran, se descubre más de la arquitectura. Esto, a su vez, lleva a la maduración de más casos de uso.

Este proceso continua hasta que se considere que la arquitectura es estable.

2.6.3. EL PROCESO UNIFICADO ES ITERATIVO E INCREMENTAL

El desarrollo de un producto software comercial supone un gran esfuerzo que puede durar entre varios meses hasta posiblemente un año o más. Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento.

Las iteraciones hacen referencia a pasos en flujo de trabajo, y los incrementos, al crecimiento del producto. Para una efectividad máxima, las iteraciones deben estar controladas; esto es, deben seleccionarse y ejecutarse de una forma planificada. Es por esto que son mini proyectos.

Los desarrolladores basan la selección de lo que se implementará en una iteración en dos factores. En primer lugar, aliteración trata de un grupo de casos de uso que juntos amplían la utilidad del producto desarrollado hasta ahora. En segundo lugar, la iteración trata de los riegos más importantes. Las iteraciones sucesivas se construyen sobre los artefactos de desarrollo tal como quedaron al final de la última iteración. Al ser mini proyectos, comienzan con los casos de uso y continúan a través del atrabajo de desarrollo subsiguiente—análisis, diseño, implementación y prueba—, que termina convirtiendo en código ejecutable los casos de uso que se desarrollan en la iteración. Por supuesto, un incremento no necesariamente es aditivo. Especialmente en las primeras fases del ciclo de vida, los desarrolladores pueden tener que reemplazar un diseño superficial por uno más detallado o sofisticado. En fases posteriores, los incrementos son típicamente aditivos.

En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes, y verifican que los componentes satisfacen los casos de uso.

Si una iteración cumple con sus objetivos—como suele suceder—el desarrollo continúa con la siguiente iteración. Cuando una iteración no cumple sus objetivos, los desarrolladores deben revisar sus decisiones previas y probar con un nuevo enfoque.

Para alcanzar el mayor grado de economía en el desarrollo, un equipo de proyecto intentara seleccionar solo las iteraciones en un orden lógico. Un proyecto con éxito se ejecutara de una forma directa, solo con pequeñas desviaciones del curso que los desarrolladores planificaron inicialmente. Por supuesto, en la medida en que se añadan iteraciones o se alteren el orden de las mismas por problemas inesperados, el proceso de desarrollo consumirá más esfuerzo y tiempo. Uno de los objetivos de la reducción del riesgo es minimizar los problemas inesperados.

Son muchos los beneficios de un proceso iterativo controlado:

- La iteración controlada reduce el coste de riesgo a los costes de un solo incremento. Si los desarrolladores tienen que repetir la iteración, la organización solo pierde el esfuerzo mal empleado de la iteración, no el valor del producto entero.
- La iteración controlada reduce el riesgo de no sacar al mercado el producto en el calendario previsto. Mediante la identificación de riesgos en fases tempranas del desarrollo, el tiempo que se gasta en resolverlos se emplea al principio de la planificación, cuando la gente está menos presionada por cumplir los plazos. En el método “tradicional”, en el cual los problemas complicados se revelan por primera vez en la prueba del sistema, el tiempo necesario para resolverlos normalmente es mayor que el tiempo que queda en la planificación, y casi siempre obliga a retrasar la entrega.

- La iteración controlada acelera el ritmo del esfuerzo de desarrollo en su totalidad debido a que los desarrolladores trabajan de manera más eficiente para obtener resultados claros a corto plazo, en lugar de tener un calendario largo, que se prolonga eternamente.
- La iteración controlada reconoce una realidad que a menudo se ignora que las necesidades del usuario y sus correspondientes requisitos no pueden definirse completamente al principio. Típicamente, se refina en sucesivas iteraciones. Esta forma de operar hace más fácil la adaptación a los requisitos cambiantes.

Estos conceptos: los de desarrollo dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental, son de igual importancia. La arquitectura proporciona la estructura sobre la cual guiar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración. La eliminación de una de las tres ideas reduciría drásticamente el valor del Proceso Unificado. Es como un taburete de tres patas. Sin una de ellas, el taburete se cae. Ahora que hemos presentado los tres conceptos clave, es el momento de echar un vistazo al proceso en su totalidad, su ciclo de vida, artefactos, flujos de trabajo, fases e iteraciones.

2.6.4. LA VIDA DEL PROCESO UNIFICADO

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema, como se muestra en la **(Figura 6)**. Cada ciclo concluye con una **versión** del producto para los clientes.

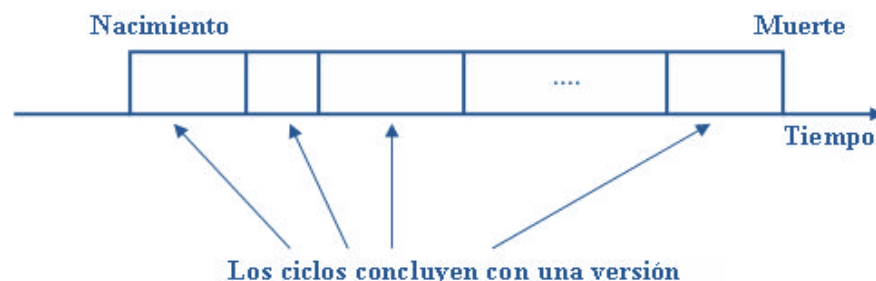


Figura 6: La Vida de un Proceso Consta de Ciclos desde su Nacimiento Hasta su Muerte.

Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición. Cada fase se subdivide a su vez en iteraciones, como se ha dicho anteriormente. Como se muestra en la **(Figura 7)**.

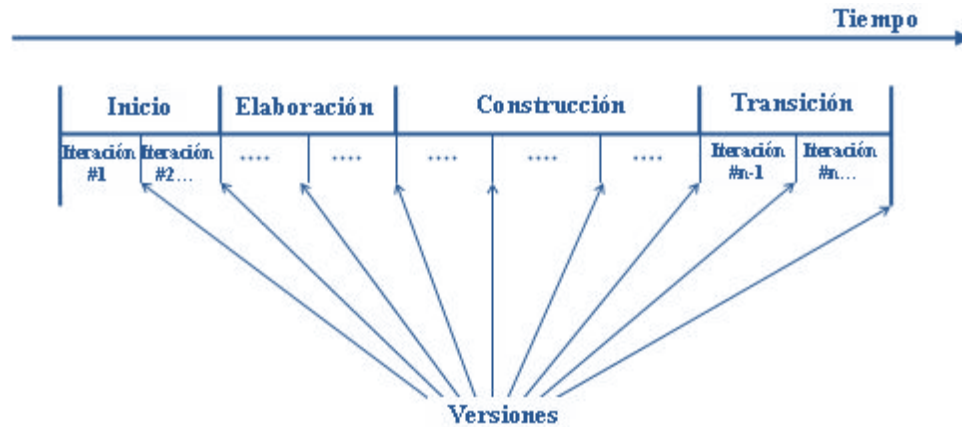


Figura 7: Un ciclo con sus fases e iteraciones.

2.6.5. FASES DENTRO DE UN CICLO

Cada ciclo se desarrolla a lo largo del tiempo. Este tiempo, a su vez, se divide en cuatro fases, como se muestra en la Figura 4. A través de una secuencia de modelos, los implicados visualizan lo que está sucediendo en esas fases. Dentro de cada fase, los directores o los desarrolladores pueden descomponer adicionalmente el trabajo, en iteraciones con sus incrementos resultantes. Cada fase termina con un **hito**. Cada hito se determina por la disponibilidad de un conjunto de artefactos; es decir, ciertos modelos o documentos han sido desarrollados para alcanzar un estado predefinido. Los hitos tienen muchos objetivos. El más crítico es que los directores deben tomar ciertas decisiones cruciales antes de que el trabajo pueda continuar con la siguiente fase. Los hitos también permiten a la dirección, y a los mismos desarrolladores, controlar el progreso del trabajo según pasa por esos cuatro puntos clave. Al final, se obtiene un conjunto de datos a partir del seguimiento del tiempo y los recursos humanos para otros proyectos, en la asignación de recursos durante el tiempo que dura el proyecto, y en el control del proceso contrastado con las planificaciones.

La **(Figura 8)** muestra en la columna izquierda los flujos de trabajo, requisitos, análisis, diseño, implementación y prueba. Las curvas son una aproximación (no deben considerarse muy literalmente) de hasta donde se llevan a cabo los flujos de trabajo en cada fase. Recuérdese que cada fase se divide normalmente en iteraciones o mini-proyectos.

Una iteración típica pasa por lo cinco flujos de trabajo, como se muestra en la Figura 4 para una iteración en la fase de elaboración.

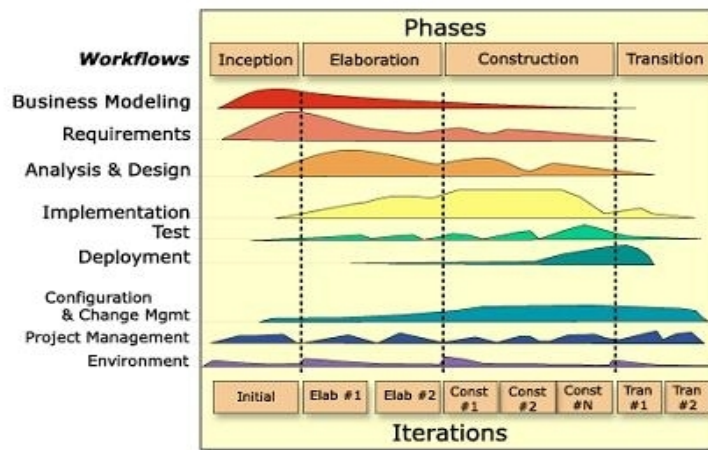


Figura 8: Fases de RUP

Durante la *fase de inicio*, se desarrolla una descripción del producto final a partir de una buena idea y se presenta el análisis de negocio para el producto. Esencialmente, esta fase responde a las siguientes preguntas:

- ¿Cuáles son las principales funciones del sistema para sus usuarios más importantes?
- ¿Cómo podría ser la arquitectura del sistema?
- ¿Cuál es el plan de proyecto y cuánto costará desarrollar el producto?

La respuesta a la primera pregunta se encuentra en un modelo de casos de uso simplificado que contenga los casos de uso más críticos. Cuando lo tengamos, la arquitectura es provisional, y consiste típicamente en un simple esbozo que muestra

los subsistemas más importantes. En esta fase, se identifican y priorizan los riesgos más importantes, se planifica en detalle la fase de elaboración, y se estima el proyecto de manera aproximada.

Durante la *fase de elaboración*, se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema.

La relación entre la arquitectura del sistema y el propio sistema es primordial. Una manera simple de expresarlo es decir que la arquitectura es análoga al esqueleto cubierto por la piel pero con muy poco músculo (el software) entre los huesos y la piel, sólo lo necesario para permitir que el esqueleto haga movimientos básicos. El sistema es el cuerpo entero con esqueleto, piel y músculos.

Por tanto, la arquitectura se expresa en forma de vistas de todos los modelos del sistema, los cuales juntos representan al sistema entero. Esto implica que hay vistas arquitectónicas del modelo de casos de uso, del modelo de análisis, del modelo de diseño, del modelo de implementación y modelo de despliegue. La vista del modelo de implementación incluye componentes para probar que la arquitectura es ejecutable. Durante esta fase del desarrollo, se realizan los casos de uso más críticos que se identificaron en la fase de comienzo. El resultado de esta fase es una línea base de la arquitectura.

Al final de la fase de elaboración, el director de proyecto esta en disposición de planificar las actividades y estimar los recursos necesarios para terminar el proyecto. Aquí la cuestión fundamental es: ¿son suficientemente estables los casos de uso, la arquitectura y el plan, y están los riesgos suficientemente controlados como para que seamos capaces de comprometernos al desarrollo entero mediante un contrato?.

Durante la *fase de construcción* se crea el producto se añaden los músculos (software terminado) al esqueleto (la arquitectura). Es en esta fase, la línea base de la arquitectura crece hasta convertirse en el sistema completo. La descripción evoluciona hasta convertirse en un producto preparado para ser entregado a la comunidad de usuarios. El grueso de los recursos requeridos se emplea durante esta

fase del desarrollo. Sin embargo, la arquitectura del sistema es estable, aunque los desarrolladores pueden descubrir formas mejores de estructurar el sistema, ya que los arquitectos recibirán sugerencias de cambios arquitectónicos de menor importancia. Al final de esta fase, el producto contiene todos los casos de uso que la dirección y el cliente han acordado para el desarrollo de esta versión. Sin embargo, puede que no esté completamente libre de defectos. Muchos de estos defectos se descubrirán y solucionarán durante la fase de transición.

La pregunta decisiva es: ¿cubre el producto las necesidades de algunos usuarios de manera suficiente como para hacer una primera entrega?

La *fase de transición* cubre el periodo durante el cual el producto se convierte versión beta. En la versión beta un número reducido de usuarios con experiencia prueba el producto e informa de defectos y deficiencias. Los desarrolladores corrigen los problemas e incorporan algunas mejoras sugeridas en una versión general dirigida a la totalidad de la comunidad de usuarios. La fase de transición conlleva actividades como la fabricación, formación del cliente, el proporcionar una línea de ayuda y asistencia, y la corrección de los defectos que se encuentren tras la entrega. El equipo de mantenimiento suele dividir esos defectos en dos categorías: los que tiene suficiente impacto en la operación para justificar una versión incrementada (versión *delta*) y los que pueden corregirse en la siguiente versión normal.

2.7. UML (LENGUAJE UNIFICADO DE MODELADO) ⁶

El lenguaje unificado de modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para diseñar, construir, hojear, configurar mantener y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e

⁶ “El Lenguaje Unificado de Modelado. Manual de Referencia” Ivar Jacobson, Grady Booch, James Rumbaugh. Madrid, 2000.

incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación, principios generales. Tiene partes estáticas, dinámicas de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero esta pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos.

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. La estructura estática define los tipos de objetos importantes para un sistema y para su implementación, así como las relaciones entre los objetos.

El comportamiento dinámico define la historia de los objetos en el tiempo y la comunicación entre objetos para cumplir sus objetivos. El modelar un sistema desde varios puntos de vista, separados pero relacionados, permite entenderlo para diferentes propósitos.

UML también contiene construcciones organizativas para agrupar los modelos en paquetes, lo que permite a los equipos de software dividir grandes sistemas en piezas de trabajo, para entender y controlar las dependencias entre paquetes, y para gestionar las versiones de las unidades del modelo, en un entorno de desarrollo complejo. Contiene construcciones para representar decisiones de implementación y para elementos de tiempo de ejecución en componentes.

UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguajes de programación, así como construir modelos por ingeniería inversa a partir de programas existentes. UML no es un lenguaje altamente formal pensado para probar teoremas.

Hay varios lenguajes de ese tipo, pero no son fáciles de entender ni de usar para la mayoría de los propósitos. UML es un lenguaje de modelado de propósito general. Para dominio especializado, tales como la composición de IGU, diseño de circuitos VLSI, o inteligencia artificial basada en reglas, podría ser más apropiada una herramienta especializada con un lenguaje especial. UML es un lenguaje de modelado discreto.

No se creo para modelar sistemas continuos como los basados en ingeniería y física. UML quiere ser un lenguaje de modelado universal, de propósito general, para sistemas discretos, tales compuestos por software, firmware o lógica digital.

2.8. VISTAS DE UML

No hay ninguna línea entre los diferentes conceptos y las construcciones en UML, pero por conveniencia, nosotros los dividimos en varias vistas. Una vista es simplemente un subconjunto de UML que modela construcciones que representan un aspecto de un sistema. La división en diversas vistas es algo arbitraria. Una o dos clases de diagramas proporcionan una notación visual para los conceptos de cada vista. En el nivel superior, las vistas se pueden dividir en tres áreas: clasificación estructural, comportamiento dinámico y gestión del modelo.

La clasificación estructural describe los elementos del sistema y sus relaciones con otros elementos. Los clasificadores incluyen clases, casos de uso componentes, nodos y elementos proporcionan la base sobre la cual se construye el comportamiento dinámico. La clasificación de las vistas incluye la vista estática, la vista de casos de uso, y la vista de implementación.

El comportamiento dinámico describe el comportamiento de un sistema en el tiempo. El comportamiento se puede describir como serie de cambios a las fotos del sistema dibujadas a partir de la visión estática. Las vistas de comportamiento dinámico incluyen vista de la máquina de estados, la vista de actividad y la vista de interacción.

La gestión del modelo describe la organización de los propios modelos en unidades jerárquicas. El paquete es la unidad genérica de organización para los modelos. Los

paquetes especiales incluyen a los modelos y a los subsistemas. La vista de gestión del modelo cruza las otras vistas y las organiza para el trabajo de desarrollo y el control de configuración.

ÁREA	VISTA	DIAGRAMAS	CONCEPTOS PRINCIPALES
Estructural	Vista estática	Diagrama de clases	Clases, asociación, generalización, interfaz.
	Vista de casos de uso	Diagramas de caso de uso	Casos de uso, actor, asociaron, extensión, inclusión, generalización de casos de uso.
	Vista de implementación	Diagrama de componentes	Componente, interfaz, dependencia, localización.
	Vistas de despliegue	Diagrama de despliegue	Nodo, componente, dependencia, localización
Dinámica	Vista de maquina de estados	Diagrama de estados	Estado, evento transición, acción.
	Vista de actividad	Diagrama de actividad	Estado, actividad, transición de terminación, división unión
	Vista de interacción	Diagrama de secuencia	Interacción, objeto, mensaje, activación.
		Diagrama de colaboración	Colaboración, interacción, rol de colaboración, mensaje.
Gestión del modelo	Vista de gestión del modelo	Diagrama de clases	Paquete, subsistema, modelo.
Extensión de UML	Todas	Todos	Restricción, estereotipo, valores etiquetados

3.1. CAPTURA DE REQUISITOS

Llamamos captura de requisitos al acto de descubrimiento, es el proceso de averiguar, normalmente en circunstancias difíciles, lo que se debe construir. De hecho, es tan difícil que todavía no es poco común para los equipos de proyecto el comenzar a escribir código (lo cual es bastante fácil) antes de que hayan firmando simplemente lo que se supone que debe hacer el código (lo cual es difícil de determinar).

En este apartado se enumerarán los requisitos identificados con el objetivo de encontrar los requisitos correctos para construir el sistema correcto, para lo cual se describirá el contexto del sistema a través del modelado del contexto y del negocio, modelos que nos permiten comprender al sistema en cuestión.

3.2. REQUISITOS CANDIDATOS

Consiste en una lista de ideas y características deseables del sistema a desarrollar, esta lista se utiliza en la etapa de planificación y ayudan a descubrir los requerimientos del sistema. Los requisitos candidatos identificados son:

- Ver información del hospital, visión, misión, historia, autoridades y personal.
- Consultar servicios y prestaciones
- Consultar sobre los programas, como ser qué es el programa, a quiénes va dirigido, requisitos para acceder al programa, servicios que cubre.
 - o Seguro Público de Salud (SPS)
 - o SUSAT
 - o SSPAM
 - o Bebé Churo y Sanito (Alimento Complementario)
 - o Abuelo Querido Churo y Sanito (Complemento Nutricional)

o Mamá Chura (Suplemento Nutricional)

- Ver información de las secciones o unidades del hospital (maternidad, laboratorio, medicina general, pediatría, cirugía) detallando el servicio, ubicación, médicos y horarios de atención, requisitos para la atención al paciente y costos.
- Consulta externa
- Actividades y noticias recientes
- Investigaciones y artículos
- Guías de pabellones y habitaciones.
- Preguntas frecuentes
- Foros categorizados

3.3. CONTEXTO DEL SISTEMA

Existen dos aproximaciones que nos ayudan a expresar el contexto de un sistema en forma utilizable para desarrolladores de software: modelado del dominio y modelado del negocio.

3.3.1. MODELO DEL NEGOCIO

El modelado del negocio describe los procesos, existentes u observados con el objetivo de comprenderlos.

El Hospital San Juan de Dios de Entre Ríos en la actualidad cuenta con sistemas internos que permiten gestionar los procesos que ocurren diariamente, sin embargo no cuenta con medios de comunicación que faciliten la información entre el hospital y los pacientes.

Consultar Información

Cuando un ciudadano de a pie, desea conocer los servicios y prestaciones que ofrece el hospital, no cuenta con los medios necesarios para informarse adecuadamente, dando como resultado la desinformación de la población, lo cual deriva en diversos problemas como ser:

- El ciudadano no conoce los beneficios que posee por ley y los requisitos para acceder a dichas prestaciones.
- El ciudadano debe averiguar por su cuenta los requisitos, lugares de atención y procedimientos para acceder a un programa.
- El ciudadano se encuentra desinformado de las prestaciones que cubre cada programa, lo cual provoca en muchas ocasiones, que el paciente sea víctima de personas inescrupulosas que se aprovechan de los pacientes para obtener beneficios personales.

Por otro lado no existen medios para enviar y recibir información entre las entidades que requieren reportes periódicos emitidos por la entidad.

Enviar reportes

El Hospital regional San Juan de Dios de Entre Ríos responde al SEDES, a quién de forma periódica debe enviar reportes de los programas, inventario de farmacia, cantidad y personas inscritas en los programas, entre otros, estos reportes son obtenidos de forma periódica y a requerimiento de la institución, los cuales para ser enviados al SEDES, son quemados en un CD y enviado por alguno de los medios de transporte a la ciudad de Tarija, provocando retrasos en el tiempo de envío y molestias a los funcionarios tanto de Entre Ríos como de Tarija, puesto que deben ir a dejar y recoger los informes de las oficinas de los centros de transporte.

En la actualidad se cuentan con el Sistema SIAF (Sistema Integrado de Administración Financiera) el cual está compuesto de los siguientes módulos:

Contabilidad y Presupuesto, Venta de servicios Hospitalarios, Inventario de Medicamentos-Farmacia, Control de Almacenes – Inventarios, Licitaciones – Cotizaciones, Control de personal y Control de Activos Fijos.

Organigrama

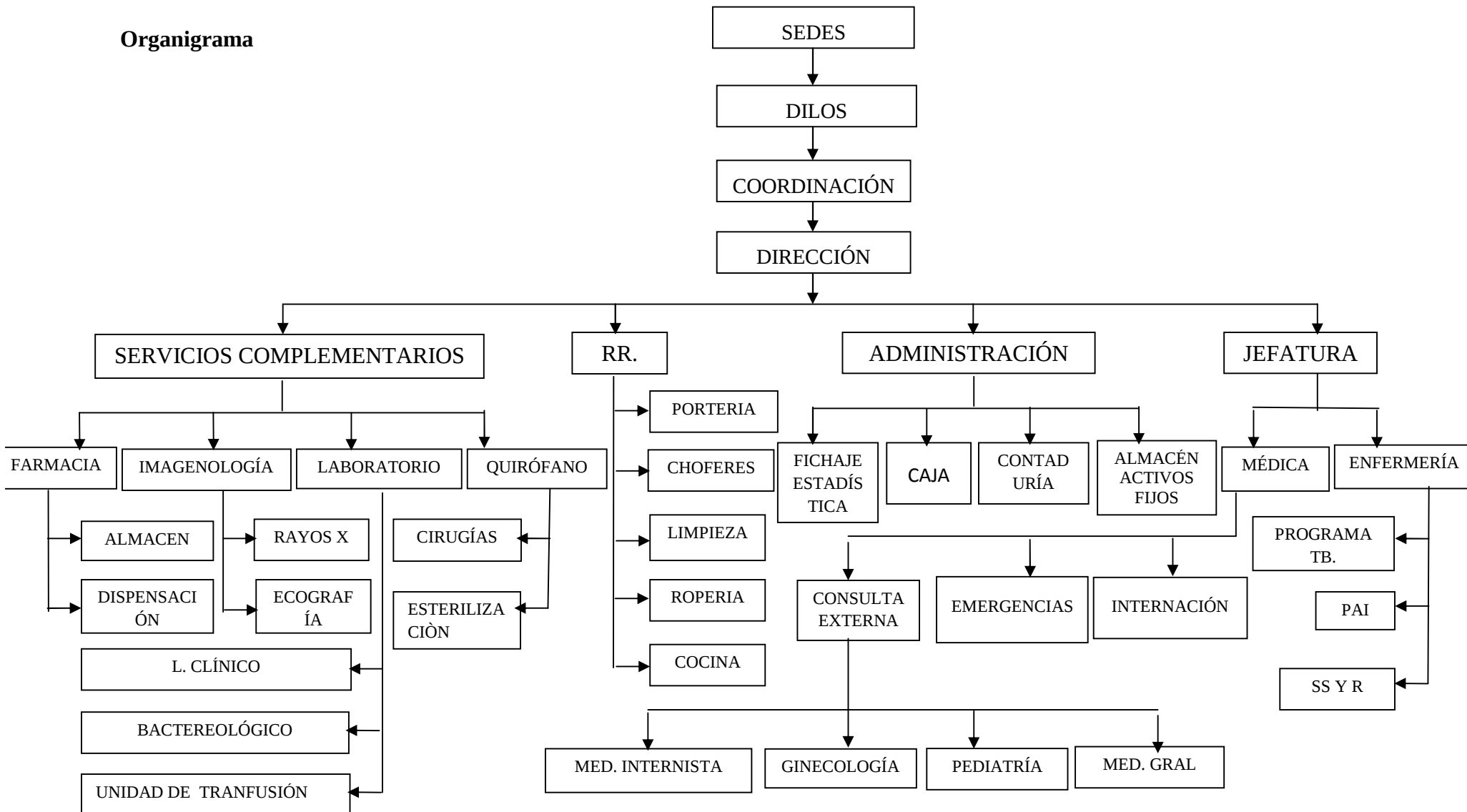


Figura 9: Organigrama del Hospital Regional San Juan de Dios de Entre Ríos

3.3.2. MODELO DE DOMINIO

Un modelado del dominio describe los conceptos importantes del contexto como objetos del dominio y enlaza estos objetos unos con otros.

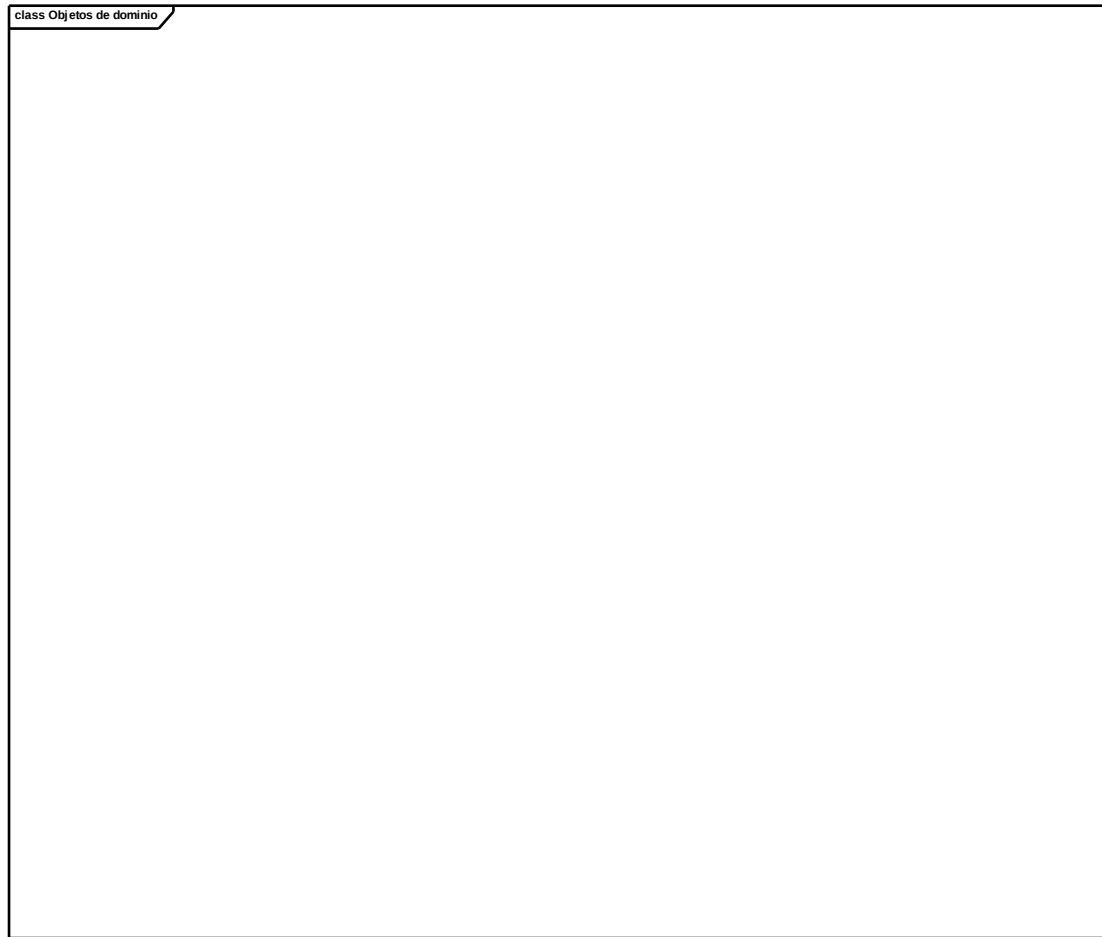


Figura 10: Modelo de Domino

3.4. REQUISITOS FUNCIONALES

Responde a la interrogante ¿qué hace el sistema? Estos requisitos son descritos más adelante a través de los diagramas de casos de uso.

El sistema propuesto tiene como funcionalidades principales, las siguientes:

Gestionar contenido, consiste en crear, modificar y eliminar contenido en el sistema web, el contenido puede ser texto, imágenes, videos y/o sonido, adicionalmente se

toma en cuenta los estados de este contenido, que puede ser publicado o en borrador, así mismo el contenido puede ser público o privado. Para que el contenido pueda ser administrado de forma correcta es importante que incluya:

- **Gestionar páginas**, las páginas son los contenidos que se van a publicar en el sistema, tiene un título de la página y su contenido, que es un texto y opcionalmente fotografías, imágenes, videos, sonido.
- **Gestión multimedia**, que permitirá insertar contenido en forma de imágenes, sonido y video, dando la facilidad al usuario de localizar el contenido e insertarlo en la posición y tamaño deseado.
- **Gestión de menús**, consiste crear menús en forma de árbol, de tal manera que se puedan crear jerarquías de menús, que permitirán relacionar el contenido con el menú correspondiente.
- **Gestión de avisos**, es una sección especial donde se mostraran avisos u acontecimientos importantes que deben ser resaltados en el sitio web.
- **Gestión de noticias**, en esta sección se cargan noticias referentes a actividades generadas o relacionadas con el hospital y/o algunas de sus unidades, servicios y programas.
- **Gestión de preguntas frecuentes**, aquí se podrá cargar preguntas que generalmente hacen los pacientes y ciudadano en general, cada pregunta tiene su respuesta correspondiente.
- **Gestión de archivos**, esta sección del sistema cobra bastante importancia, puesto que es aquí donde el usuario podrá categorizar y subir reportes generados en los demás sistemas de la institución, con la finalidad de que los usuarios registrados tengan la capacidad de poder ingresar, ver y descargar los reportes.
- **Gestión de solicitud de información**, la gestión consistirá en ver las solicitudes de los internautas y responder a las interrogantes planteadas por

él/ella, aquí podremos ver el listado de consultar, responder y borrar consultas.

- **Gestión de foro**, básicamente tiene que ver con la otorgación de permisos, creación de usuarios y categorías del foro.

Administrar sistema

Se debe incluir en el sistema los elementos de configuración que permitirán el correcto funcionamiento del sistema.

- **Gestión de usuarios**, la gestión de usuarios es un aspecto que se debe tomar en cuenta, puesto que aquí se crean los roles de usuarios con sus respectivos permisos, de tal manera que se muestre el contenido sólo a los usuarios que posean con un determinado rol.
- **Configurar parámetros del sistema**, esta sección debe permitir configurar aspectos generales, como ser números de paginación, formato de fechas, tamaño de archivos y similares.
- **Administrar tipos de menús**, en esta sección se detallará la ubicación y cantidad de los menús, como ser superior, lateral, inferior. Estas categorías son con fines de organización y personalización del sistema.

Consultar Información

Esta es la sección pública, dedicada al usuario final, que mostrará toda la información pública ingresada en el sistema. El usuario puede:

- **Realizar búsquedas**, buscar información en la página a través de criterios de búsquedas, a través del cual se mostraran todos los contenidos que coincidan con el criterio introducido.
- **Participar en foro**, el usuario tendrá la opción de ver el foro y en caso de inscribirse podrá participar en él.

- **Solicitar información**, se le ofrecerá la opción de llenar un formulario donde podrá escribir su consulta y dejar su correo electrónico al cual se le enviará la respuesta a la pregunta formulada.
- **Navegar por el sitio**, se mostrará los menús, enlaces, noticias, eventos, preguntas frecuentes, foros y demás información que sea de carácter público.

3.5. REQUISITOS NO FUNCIONALES

Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad.

- Los archivos que están permitidos para el gestor de archivos son: pdf, doc, docx, xls,xlsx, no mayor a 2 MB.
- Las imágenes deben ser: png, jpeg, jpg, gif no mayor a 2 MB.
- Los videos y sonidos no deben exceder los 5 MB.

3.6. REQUISITOS ADICIONALES

Los requisitos que son más genéricos y no pueden relacionarse con un caso de uso en concreto son denominados requisitos adicionales.

- El sistema debe mostrar el contenido de forma ordenada y comprensible para el usuario final.
- El sistema debe ser fácil de administrar, para lo cual se debe tomar en cuenta estándares de usabilidad.
- El sistema debe ser rápido en cuestión de carga de contenidos e imágenes.
- El sistema preferentemente debe ser desarrollado bajo software libre.

4.1. CASOS DE USO

El esfuerzo principal en la fase de requisitos es desarrollar un modelo del sistema que se va a construir, y la utilización de los casos de uso es una forma adecuada de crear ese modelo.

Esto es debido a que los requisitos funcionales se estructuran de forma natural mediante casos de uso y a que la mayoría de los otros requisitos no funcionales son específicos de un solo caso de uso y pueden tratarse en el contexto de ese caso de uso.

4.2. MODELO DE CASOS DE USO

4.2.1. ACTORES

- **Super-Administrador**

Rol que tiene todos los privilegios y acceso a módulos del sistema, sin embargo como una norma de seguridad el sistema guardará las acciones realizadas por el usuario.

- **Administrador de Contenido**

Se separa este rol en dos con fines de organización, puesto que cada uno de ellos administra contenido diferente, aunque los permisos y los módulos al que tiene permiso son los mismos, el tipo de información que cargan son diferente.

- o **Jefe de Unidad**

Ese rol administra contenido de las unidades o reparticiones del hospital, personal, información de las unidades, derechos y deberes.

- o **Responsable de Servicios**

Este rol administra contenido de los servicios, requisitos, beneficiarios, horarios de intención y costos.

- **Usuario Registrado**

El usuario registrado podrá ver toda la información del sistema, tanto privada como pública, además podrá acceder al foro sin necesidad de crear otra cuenta para esta acción.

El usuario registrado sólo puede ser dado de alta por el super-administrador, puesto que este usuario tiene permisos para ver y descargar información privada.

- **Usuario**

Este es el rol público, la división realizada es meramente con fines de comprensión y no refleja niveles de acceso diferentes.

El paciente puede navegar y ver información del hospital y sus reparticiones, a ver las noticias, anuncios y avisos.

También puede realizar consultas, ver y participar en los foros de opinión e ingresar a preguntas frecuentes para informarse.

- o **Beneficiario**

Este usuario consulta información de los programas existentes en el hospital, requisitos, prestaciones y los seguros que cubre.

- o **Paciente Externo**

Este usuario consulta información referente a los servicios del hospital, horarios de atención y costos.

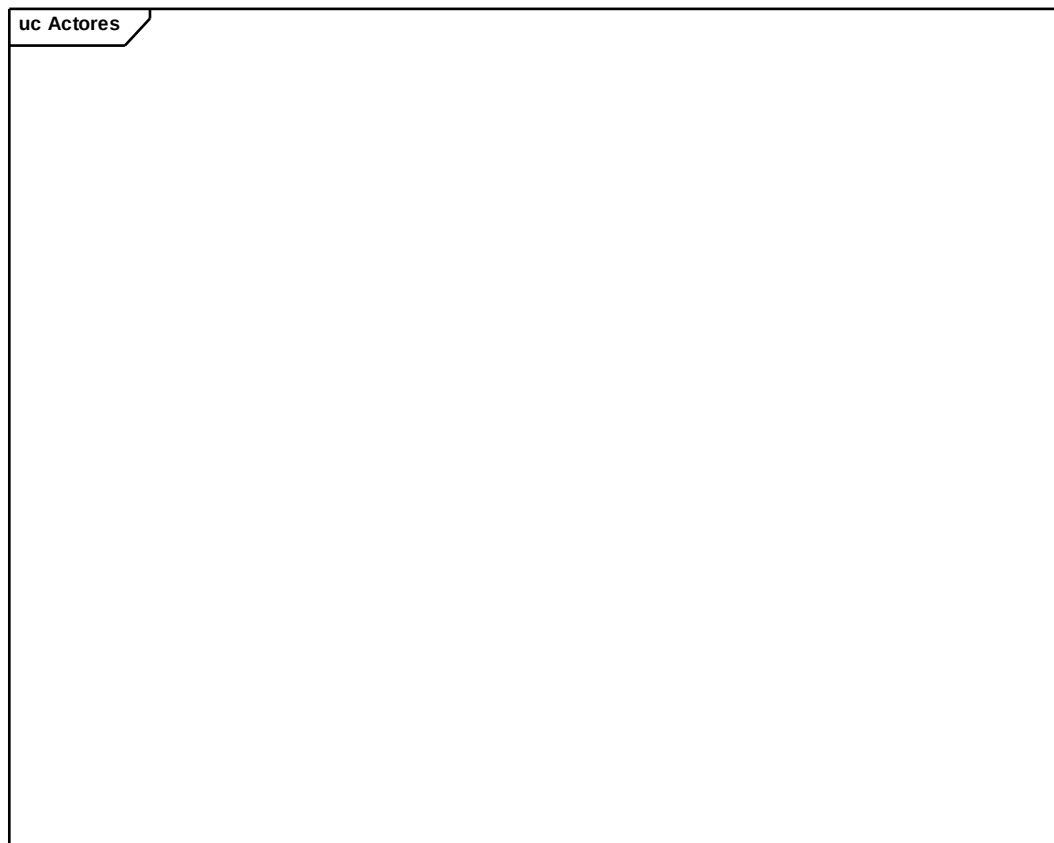


Figura 11: Actores del Sistema

4.2.2. CASOS DE USO

- Administrar Contenido del Sistema
 - o Gestionar páginas
 - o Gestionar menús
 - o Gestionar archivos
 - o Gestionar noticas
 - o Gestionar avisos
 - o Gestionar preguntas frecuentes
 - o Gestionar solicitud de información
- Administrar Sistema

- o Configurar parámetros del sistema
- o Gestionar usuarios
- o Gestionar tipos de menús.
- Consultar Información
 - o Realizar búsquedas
 - o Solicitar información
 - o Navegar por el sitio
 - o Participar en el foro

4.2.3. PRIORIZACIÓN DE CASOS DE USO

Los casos de uso deben clasificarse en primarios, secundarios y opcionales. Los casos primarios de uso representan los procesos comunes más importantes, los casos secundarios de uso representan procesos menores o raros. Los procesos opcionales de uso representan procesos que pueden no abordarse. Para ayudarnos a clasificar los casos de uso, utilizaremos parámetros que serán valorados del 1 al 5.

- a) Tiene fuerte repercusión en el diseño arquitectónico
- b) Con poco esfuerzo se pueden obtener información o ideas sobre el diseño
- c) Incluye funciones riesgosas, urgentes o complejas
- d) Requiere investigación a fondo o tecnología nueva y riesgosa
- e) Representa procesos primarios de la línea de negocios
- f) Apoya directamente el aumento de ingresos o la reducción de costos.

Caso de Uso	a	b	c	d	e	f	Suma
Gestionar páginas	5	4	5	4	5	5	28
Gestionar menús	2	5	3	4	4	4	22
Gestionar archivos	3	3	4	4	4	3	21

Gestionar noticas	2	2	3	3	3	2	15
Gestionar avisos	2	2	3	4	3	2	16
Gestionar preguntas frecuentes	2	2	3	3	3	2	15
Gestionar solicitud de información	2	2	2	2	3	2	13
Configurar parámetros del sistema	2	2	3	3	1	1	12
Gestionar usuarios	2	2	3	3	3	2	15
Gestionar tipos de menús	4	2	3	3	3	2	17
Realizar búsquedas	2	2	3	3	1	1	12
Solicitar información	2	2	3	3	2	3	15
Navegar por el sitio	4	4	4	5	5	5	27
Participar en el foro	2	2	3	3	3	3	16
Control de acceso	2	2	3	3	1	1	12

Clasificación	Caso de Uso	Justificación
Alto	Gestionar páginas Navegar por el sitio Gestionar menús Gestionar archivos Gestionar tipos de menús	Estos casos de uso son considerados altos porque representan el corazón del sistema, las funcionalidades principales se encuentran en estos casos de uso.
Mediano	Participar en el foro Gestionar avisos Gestionar noticas Gestionar preguntas frecuentes Gestionar usuarios Solicitar información Gestionar solicitud de información	Estos casos de uso, si bien forman parte del cuerpo del negocio, no repercuten de forma determinante en el diseño de la arquitectura del sistema.

Bajo	Configurar parámetros del sistema Realizar búsquedas Control de acceso	Estos casos de uso tienen una prioridad baja, es posible dejarlos para las últimas iteraciones del desarrollo del sistema.
------	--	--

4.2.4. DIAGRAMAS DE CASOS DE USO

El Diagrama de Casos de Uso permite visualizar el comportamiento del sistema o de una parte de él, mostrando la relación entre los actores y los Casos de Usos del sistema.

Estos especifican los requerimientos funcionales de una manera sencilla ya que definen el comportamiento del sistema y no su implementación.

4.2.4.1. DIAGRAMA DE CASO DE USO DE CONTEXTO

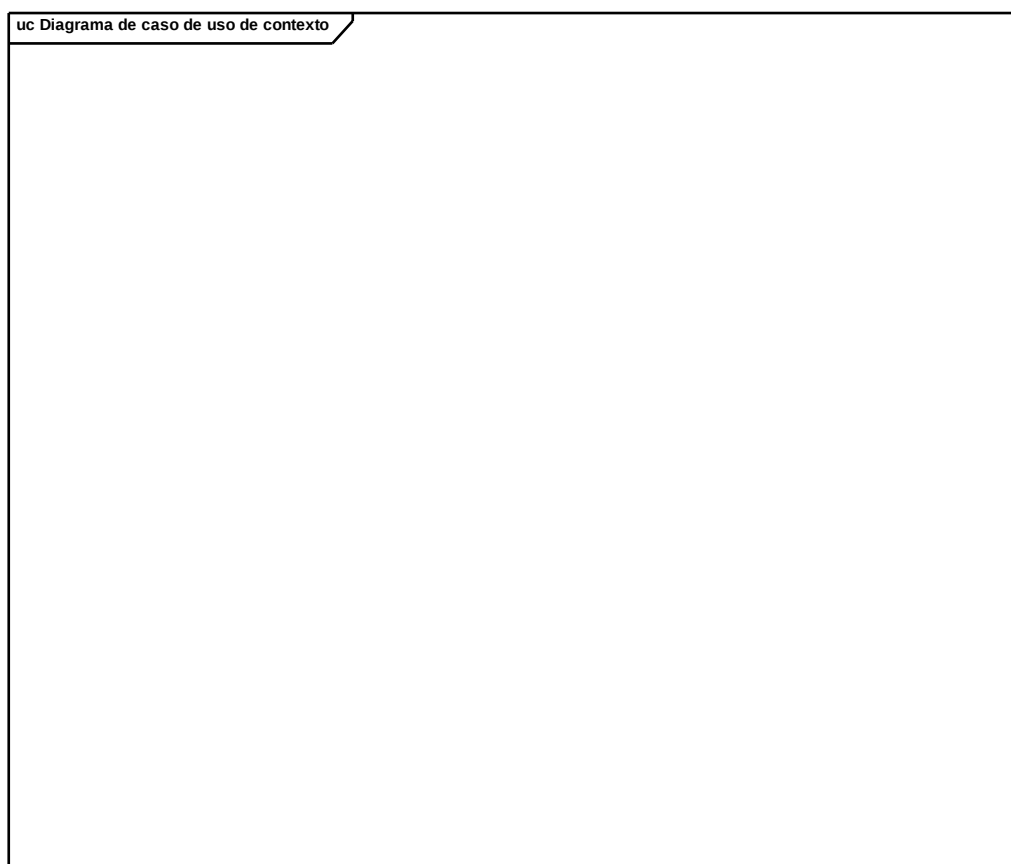


Figura 12: Diagrama de Contexto

4.2.4.2. DIAGRAMAS DE CASOS DE USO PRIMARIOS

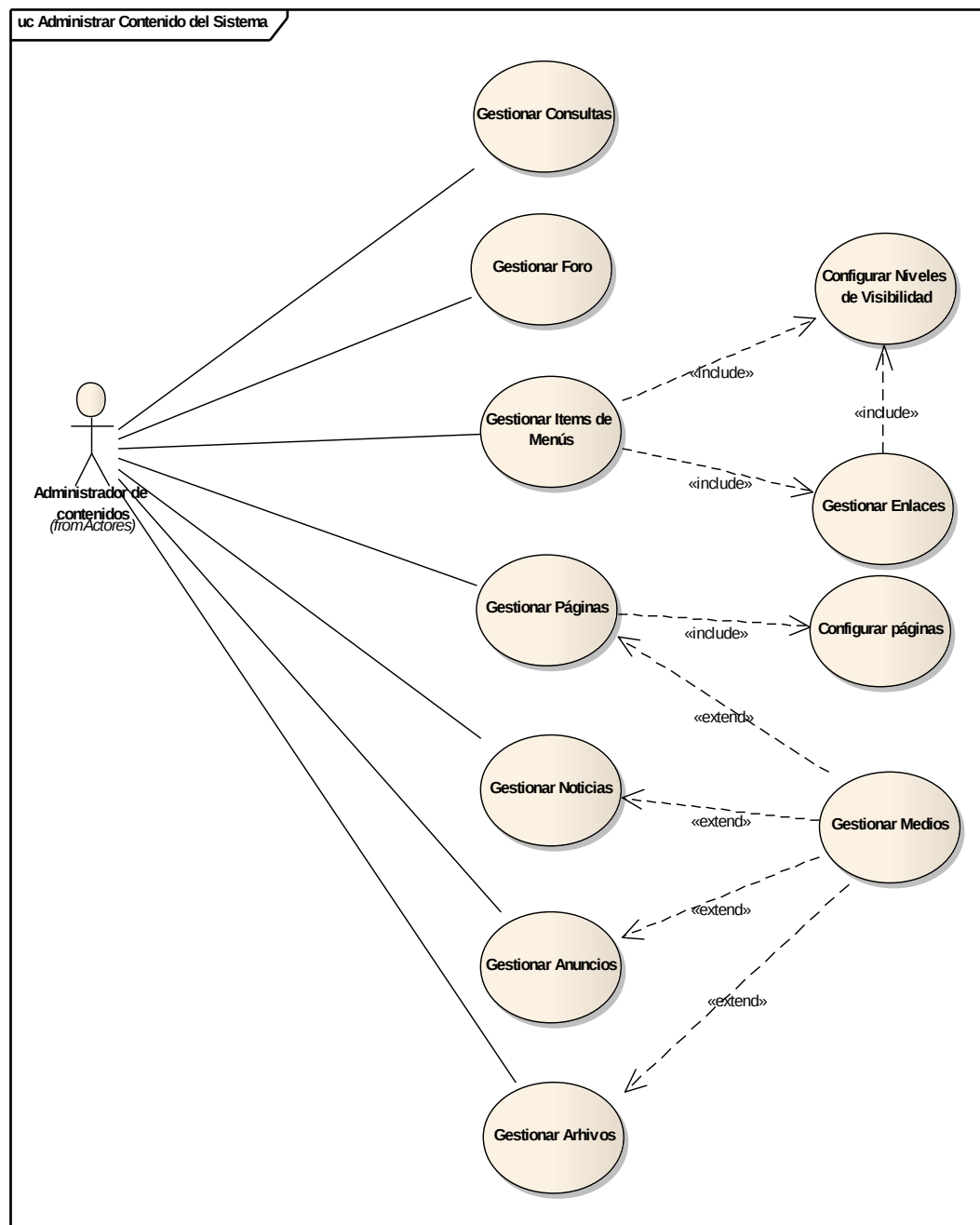


Figura13: Diagrama de Caso de Uso “Administrar Contenido del Sistema”

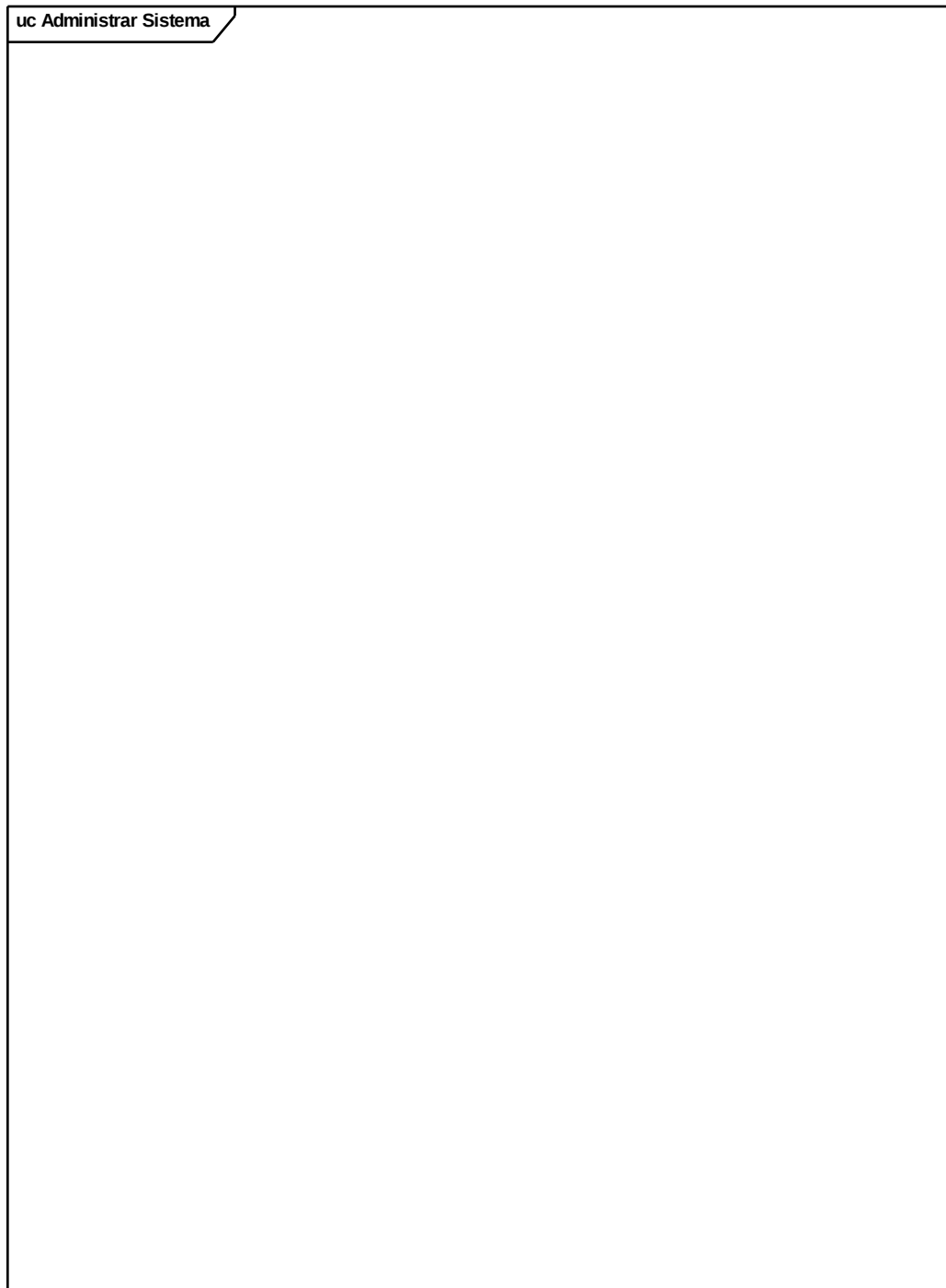


Figura14: Diagrama de Caso de Uso “Administrar Sistema”

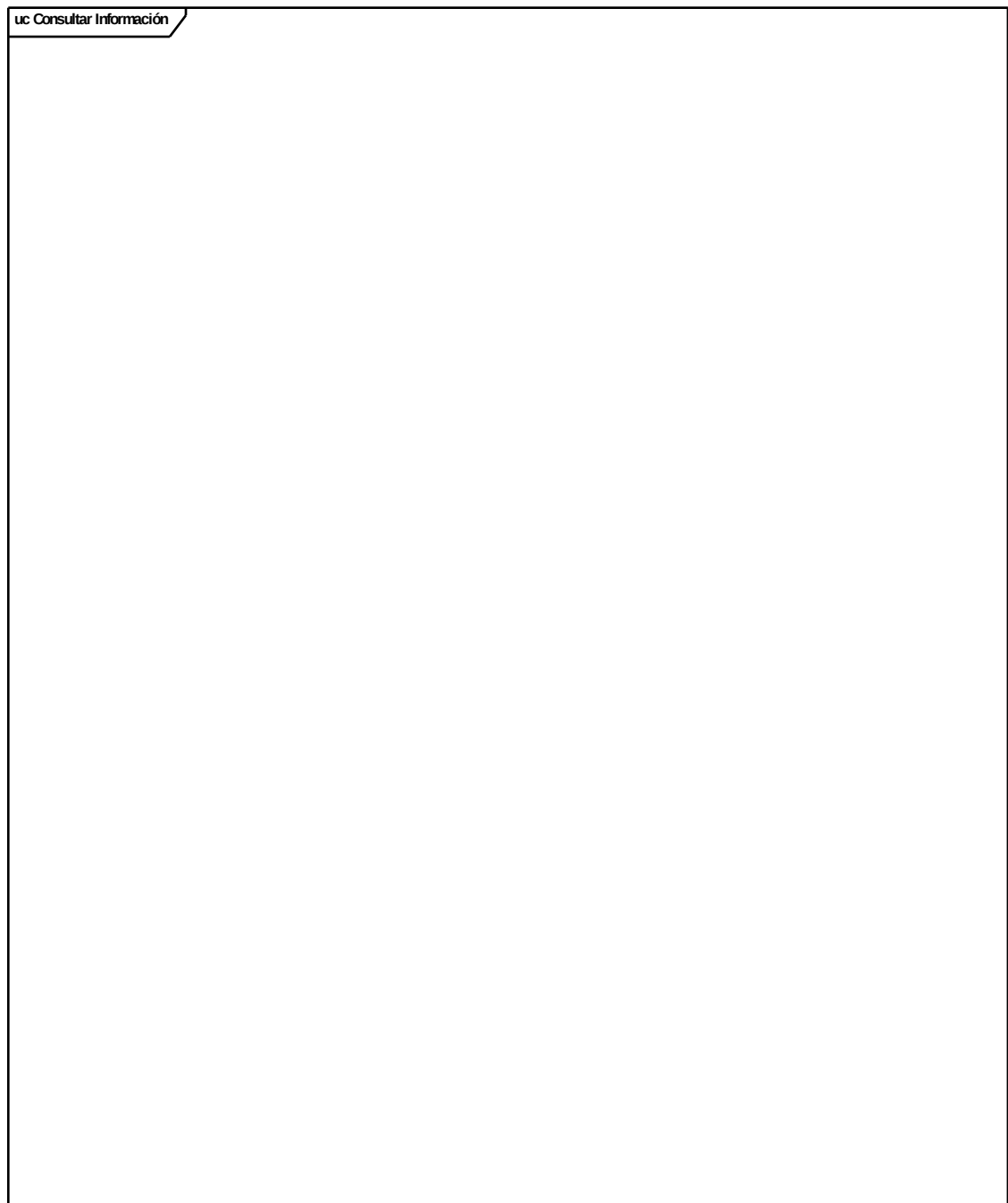


Figura15: Diagrama de Caso de Uso “Consultar Información”

4.2.4.3. DIAGRAMAS DE CASOS DE USO SECUNDARIOS

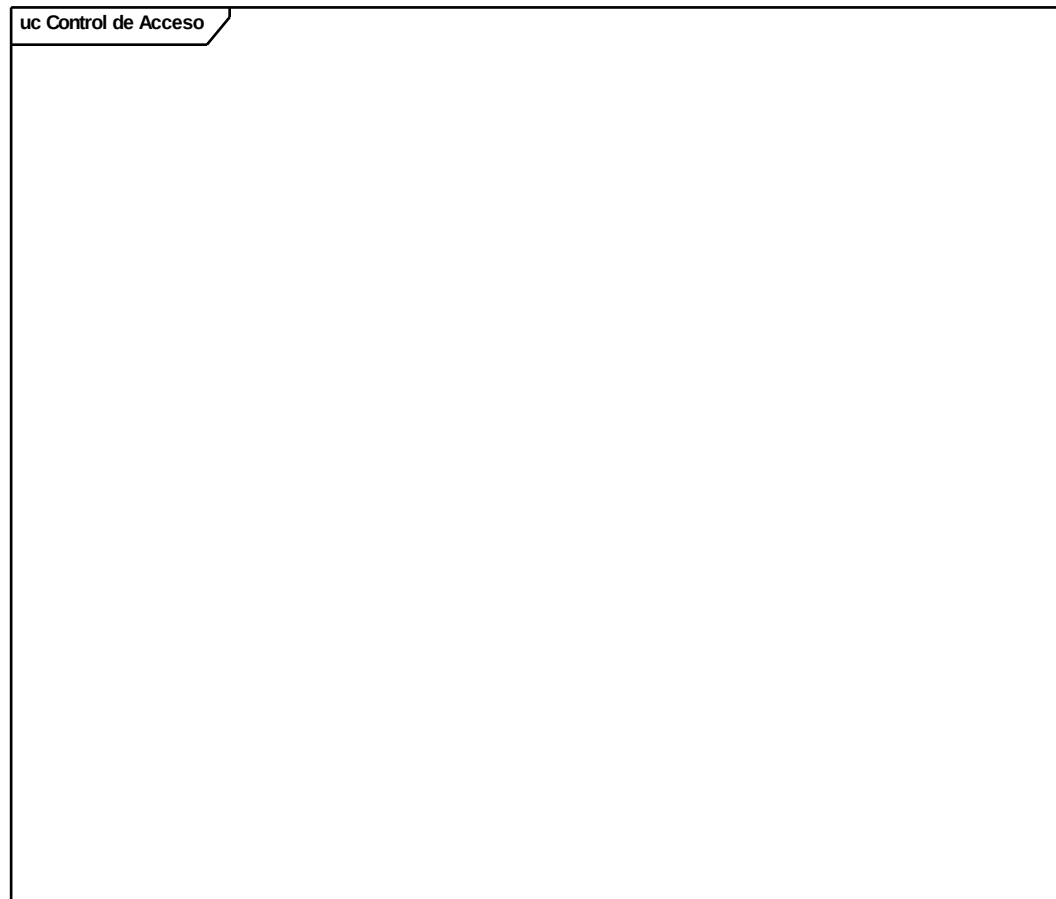


Figura16: Diagrama de Caso de Uso “Control de Acceso”

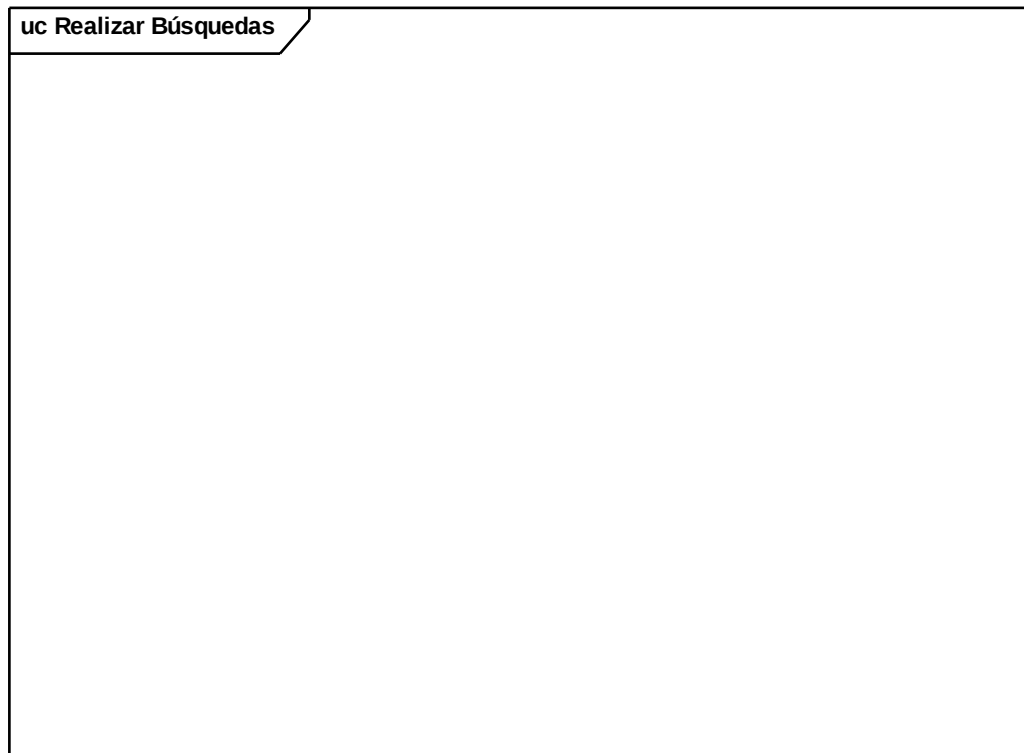


Figura17: Diagrama de Caso de Uso “Realizar Búsquedas”

4.2.4.4. DIAGRAMAS DE CASOS DE OPCIONALES

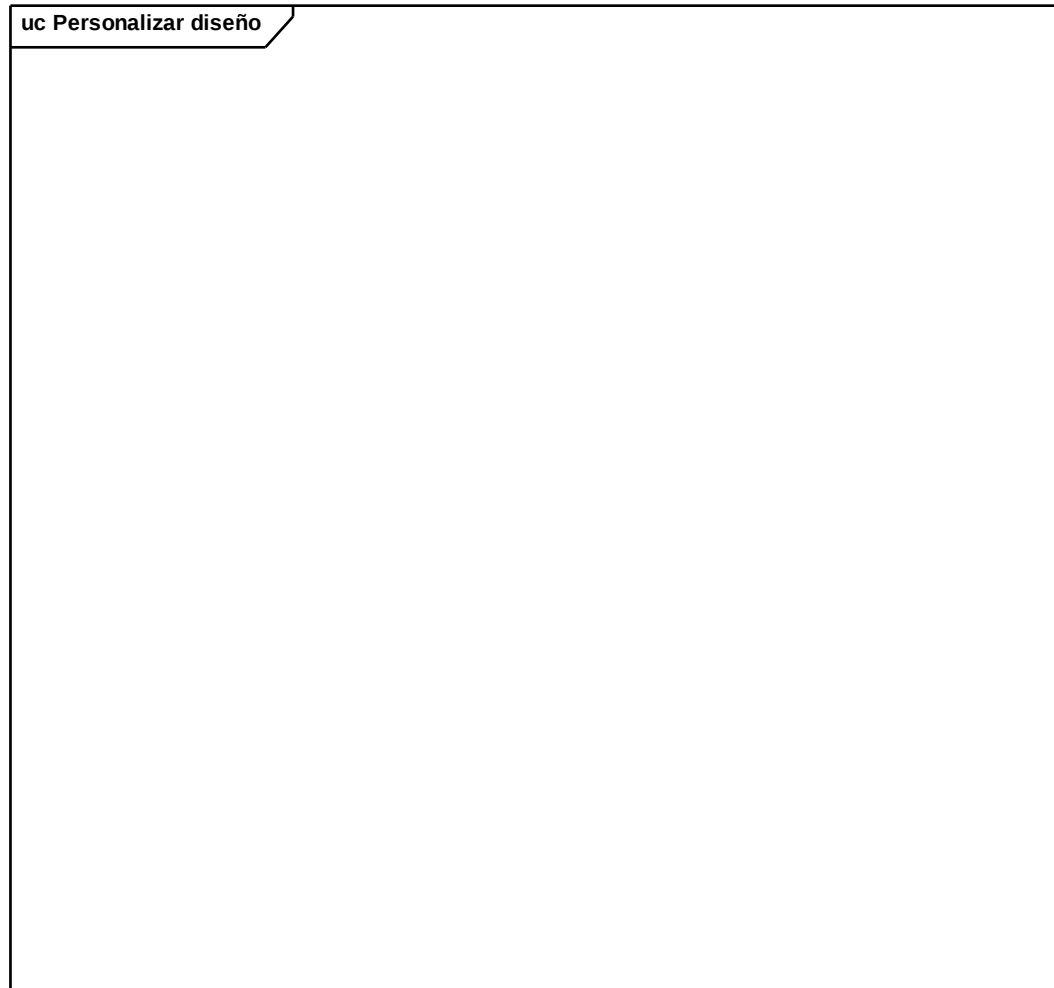


Figura 18: Diagrama de Caso de Uso "Personalizar Diseño"

4.2.5. DESCRIPCIÓN DE CASOS DE USO

Nombre:	Gestionar Páginas
Descripción: Las páginas son los contenidos que se van a publicar en el sistema, tiene un título de la página y su contenido, que es un texto y opcionalmente fotografías, imágenes, videos, sonido.	
Actores: • Super-Administrador • Administrador de contenidos	
Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.	
Flujo Normal	
Acción de los Actores	Respuesta del Sistema
1) El usuario entra al menú “contenido” y selecciona la opción “gestor de contenidos”.	2) El sistema muestra el listado de todas las páginas existentes con su respectivo detalle, cada una de ellas con las opciones: • Editar • Eliminar

3) El usuario selecciona alguna de las opciones. 5) Realizado el cambio por parte del usuario, selecciona la opción “guardar”.	• Activar/Desactivar (dependiendo del estado) • Privado/Público (dependiendo del permiso que tenga. Adicionalmente el sistema muestra un menú superior que permite: • Adicionar nueva página • Borrar en grupo • Activar/Desactivar en grupo • Privado/Público en grupo 4) El sistema identifica la opción realizada por el usuario y de acuerdo a eso muestra la página correspondiente. 6) El sistema retorna a la vista de listado de páginas.
Flujo Alternativo: 3) El usuario puede elegir otra opción del menú de administración, con lo cual se cancela el caso de uso. 5) El usurario puede “cerrar” la ventana abierta, con lo cual se cancela cualquier cambio realizado sobre la página y pasa al paso 6).	
Poscondiciones: Se realizaron los cambios correspondientes	

Nombre:	Gestión Multimedia
Descripción: Permitirá insertar contenido en forma de imágenes, sonido y video, dando la facilidad al usuario de localizar el contenido e insertarlo en la posición y tamaño deseado	
Actores: • Super-Administrador • Administrador de contenidos	
Precondiciones: • Estar logueado en el sistema y tener el permiso correspondiente. • Estar en el caso de uso “crear página” o “editar página”.	
Flujo Normal:	
Acción de los Actores	Respuesta del Sistema
1) El usuario ingresa a la opción “insertar multimedia”. 3) El usuario hace click en el botón “examinar” y selecciona el archivo multimedia que desea cargar y hace click en “subir”.	2) El sistema le muestra una ventana emergente de cargado de multimedia.

6) El usuario configura datos adicionales, como posición del archivo multimedia, largo, ancho y nombre, luego hace click en la opción “insertar”.	5) El sistema valida el tipo y tamaño del archivo para luego copiar el archivo en una carpeta del sistema. 7) El sistema inserta el contenido multimedia en la página.
Flujo Alternativo: 3) El usuario puede “cerrar” la ventana sin haber cargado el archivo y se retorna al paso 1. 5) No se reconoce la extensión del archivo, el sistema muestra un mensaje de error y regresa al paso 2. 6) El usuario puede “cerrar” la página sin guardar los datos, en ese caso el archivo multimedia ya fue subido al sistema, pero no relacionado con alguna página.	
Poscondiciones: • El archivo se encuentra en una carpeta del sistema y se hace referencia al mismo desde la base de datos.	

Nombre:	Gestión de Menús
Descripción: Consiste crear menús en forma de árbol, de tal manera que se puedan crear jerarquías de menús, que permitirán relacionar el contenido con el menú correspondiente.	
Actores: • Super-Administrador • Administrador de contenidos	
Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.	
Flujo Normal:	
Acción de los Actores	Respuesta del Sistema
1) El usuario ingresa al menú del administrador y selecciona el tipo de menú con el que desea trabajar.	2) El sistema mostrará el menú correspondiente con el listado de los ítems que posee. Además presentará las siguientes opciones: • Editar • Borrar • Activo/Desactivo Así mismo se presentaran otras opciones en el menú superior: • Nuevo ítem

3) El usuario seleccionará la acción que desea realizar. 5) El usuario realizará las tareas necesarias y hará click en la opción “guardar”.	• Borrar por grupo • Activar/Desactivar por grupo 4) El sistema mostrará la pantalla correspondiente. 6) El sistema guardará los datos y volverá al paso 2.
Flujo Alternativo: 3) El usuario puede elegir otra opción del menú de administración, con lo cual se cancela el caso de uso. 5) El usuario puede “cerrar” la ventana abierta, con lo cual se cancela cualquier cambio realizado sobre la página y pasa al paso 2.	
Poscondiciones: Los datos fueron guardados correctamente.	

Nombre:	Gestión de Avisos
Descripción: Es una sección especial donde se mostraran avisos u acontecimientos importantes que deben ser resaltados en el sitio web.	

Actores:

- Super-Administrador
- Administrador de contenidos

Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.

Flujo Normal:

Acción de los Actores	Respuesta del Sistema
1) El caso de uso inicia cuando el usuario ingresa al módulo de avisos. 3) El usuario selecciona la opción que sea realizar. 5) El usuario realizará las acciones deseadas y hará click en la opción “guardar”	2) El sistema le mostrará el listado de avisos existentes con sus respectivas opciones: • Insertar • Borrar • Editar • Activar/Desactivar 4) El sistema envía la acción correspondiente de acuerdo a la cual se mostrará la página seleccionada por el usuario. 6) El sistema validará los datos y en caso de no existir

	errores, guardará los cambios en la base de datos y volverá a la paso 2.
Flujo Alternativo: 3) El usuario puede salir del módulo sin realizar ninguna acción, simplemente desplazándose a otra opción del menú. 6) En caso de encontrarse errores, el sistema mostrará un mensaje y devolverá al usuario al paso 5.	
Poscondiciones: Las acciones fueron guardadas correctamente.	

Nombre:	Gestión de Noticias
Descripción En este caso de uso se cargan noticias referentes a actividades generadas o relacionadas con el hospital y/o algunas de sus unidades, servicios y programas.	
Actores: • Super-Administrador • Administrador de contenidos	
Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.	
Flujo Normal:	

Acción de los Actores	Respuesta del Sistema
1) El caso de uso inicia cuando el usuario ingresa al módulo de noticias. 3) El usuario selecciona la opción que sea realizar. 5) El usuario realizará las acciones deseadas y hará click en la opción “guardar”	2) El sistema le mostrará el listado de las noticias existentes con sus respectivas opciones: • Insertar • Borrar • Editar • Activar/Desactivar 4) El sistema envía la acción correspondiente de acuerdo a la cual se mostrará la página seleccionada por el usuario. 6) El sistema validará los datos y en caso de no existir errores, guardará los cambios en la base de datos y volverá a la paso 2.
Flujo Alternativo: 3) El usuario puede salir del módulo sin realizar ninguna acción, simplemente desplazándose a otra opción del menú. 6) En caso de encontrarse errores, el sistema mostrará un mensaje y devolverá al	

usuario al paso 5.

Poscondiciones:

Las acciones fueron guardadas correctamente.

Nombre:	Gestión de Preguntas Frecuentes
Descripción Aquí se podrá cargar preguntas que generalmente hacen los pacientes y ciudadano en general, cada pregunta tiene su respuesta correspondiente.	
Actores: • Super-Administrador • Administrador de contenidos	
Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.	
Flujo Normal:	
Acción de los Actores	Respuesta del Sistema
1) El caso de uso inicia cuando el usuario ingresa al módulo de preguntas frecuentes.	2) El sistema le mostrará el listado de las preguntas existentes con sus respectivas opciones:

3) El usuario selecciona la opción que sea realizar. 5) El usuario realizará las acciones deseadas y hará click en la opción “guardar”	• Insertar • Borrar • Editar • Activar/Desactivar 4) El sistema envía la acción correspondiente de acuerdo a la cual se mostrará la página seleccionada por el usuario. 6) El sistema validará los datos y en caso de no existir errores, guardará los cambios en la base de datos y volverá a la paso 2.
Flujo Alternativo: 3) El usuario puede salir del módulo sin realizar ninguna acción, simplemente desplazándose a otra opción del menú. 6) En caso de encontrarse errores, el sistema mostrará un mensaje y devolverá al usuario al paso 5.	
Poscondiciones: Las acciones fueron guardadas correctamente.	

Nombre:	Gestión de Archivos
Descripción: Este caso de uso del sistema cobra bastante importancia, puesto que es aquí donde el usuario podrá categorizar y subir reportes generados en los demás sistemas de la institución, con la finalidad de que los usuarios registrados tengan la capacidad de poder ingresar, ver y descargar los reportes.	
Actores: • Super-Administrador • Administrador de contenidos	
Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.	
Flujo Normal:	
Acción de los Actores	Respuesta del Sistema
1) El caso de uso inicia cuando el usuario ingresa al módulo de subir archivos.	2) El sistema le mostrará el listado de los archivos existentes con sus respectivas opciones: • Insertar • Borrar • Editar • Activar/Desactivar

3) El usuario selecciona la opción que sea realizar. 5) El usuario realizará las acciones deseadas y hará click en la opción “guardar”	4) El sistema envía la acción correspondiente de acuerdo a la cual se mostrará la página seleccionada por el usuario. 6) El sistema validará los datos y en caso de no existir errores, guardará los cambios en la base de datos y volverá al paso 2.
Flujo Alternativo: 3) El usuario puede salir del módulo sin realizar ninguna acción, simplemente desplazándose a otra opción del menú. 6) En caso de encontrarse errores, el sistema mostrará un mensaje y devolverá al usuario al paso 5.	
Poscondiciones: Las acciones fueron guardadas correctamente.	

Nombre:	Gestión de Solicitud de Información
Descripción La gestión consistirá en ver las solicitudes de los internautas y responder a las interrogantes planteadas por él/ella, aquí podremos ver el listado de consultar,	

responder y borrar consultas.

Actores:

- Super-Administrador
- Administrador de contenidos

Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.

Flujo Normal:

Acción de los Actores	Respuesta del Sistema
1) El caso de uso inicia cuando el usuario ingresa al módulo de solicitud de información.	2) El sistema le mostrará el listado de las solicitudes de información enviadas por los usuarios con sus respectivas opciones: • Responder • Búsqueda • Filtro respondida/sin responder
3) El usuario selecciona la opción que sea realizar.	4) El sistema envía la acción correspondiente de acuerdo a la cual se mostrará la página seleccionada por el usuario.
5) El usuario realizará las acciones deseadas y hará	

click en la opción “guardar”	6) El sistema validará los datos y en caso de no existir errores, guardará los cambios en la base de datos y volverá a la paso 2.
Flujo Alternativo: 3) El usuario puede salir del módulo sin realizar ninguna acción, simplemente desplazándose a otra opción del menú. 6) En caso de encontrarse errores, el sistema mostrará un mensaje y devolverá al usuario al paso 5.	
Poscondiciones: Las acciones fueron guardadas correctamente.	

Nombre:	Gestión de foro
Descripción Básicamente tiene que ver con la otorgación de permisos, creación de usuarios y categorías del foro.	
Actores: • Super-Administrador • Administrador de contenidos	
Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.	
Flujo Normal:	

Acción de los Actores	Respuesta del Sistema
1) El caso de uso inicia cuando el usuario ingresa al módulo de foro. 3) El usuario selecciona el menú al que desea entrar. 5) El usuario hace click sobre la opción deseada.	2) El sistema mostrará un menú con las opciones de configuración del foro: • Foro - o Tema - o Mensajes • Usuarios • Configuración • Roles - o Privilegios 4)El sistema le mostrará las opciones existentes en cada menús, como ser: • Insertar • Borrar • Editar • Activar/Desactivar 6) El sistema envía la acción correspondiente de acuerdo a la cual se mostrará la página seleccionada

7) El usuario realizará las acciones deseadas y hará click en la opción “guardar”	por el usuario. 8) El sistema validará los datos y en caso de no existir errores, guardará los cambios en la base de datos y volverá a la paso 4.
Flujo Alternativo: 3) El usuario puede salir del módulo sin realizar ninguna acción, simplemente desplazándose a otra opción del menú. 8) En caso de encontrarse errores, el sistema mostrará un mensaje y devolverá al usuario al paso 4.	
Poscondiciones: Las acciones fueron guardadas correctamente.	

Nombre:	Gestión de Usuarios
Descripción: La gestión de usuarios es un aspecto que se debe tomar en cuenta, puesto que aquí se crean los roles de usuarios con sus respectivos permisos, de tal manera que se muestre el contenido sólo a los usuarios que posean con un determinado rol.	
Actores: Super-Administrador	

Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.	
Flujo Normal:	
Acción de los Actores	Respuesta del Sistema
1) El caso de uso inicia cuando el usuario ingresa al módulo de gestión de usuarios.	2) El sistema le mostrará un listado de los usuarios con información adicional y opciones: • Nuevo usuario • Editar usuario • Activar/Desactivar usuario • Rol
3) El usuario selecciona la opción a la que desea ingresar.	4) El sistema captura la acción y muestra la página correspondiente.
5) El usuario realiza las acciones correspondientes y hace click en la opción “guardar”	6) El sistema valida los datos y si todo esta en orden procesa los datos y regresa a la pantalla 2.

Flujo Alternativo:

3) El usuario puede salir del módulo sin realizar ninguna acción, simplemente desplazándose a otra opción del menú.

6) En caso de encontrarse errores, el sistema mostrará un mensaje y devolverá al usuario al paso 2.

Poscondiciones:

El usuario ha realizado con éxitos las acciones requeridas.

Nombre:	Configurar parámetros del sistema
Descripción Este caso de uso debe permitir configurar aspectos generales, como ser números de paginación, formato de fechas, tamaño de archivos y similares.	
Actores: Super-Administrador	
Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.	
Flujo Normal:	
Acción de los Actores	Respuesta del Sistema
1) El caso de uso inicia cuando el usuario hace click sobre la opción “configuración del sitio”.	

3) El usuario podrá cambiar los datos de acuerdo a su preferencia y debe hacer click en el botón “guardar cambios”	2) El sistema mostrará una pantalla con datos de configuración del sistema: • Tamaño máximo y mínimo de imágenes • Tamaño de archivos • Rol por defecto al activarse un nuevo usuario • N° para mostrar paginación 4) El sistema valida los datos introducidos y guarda los datos en la base de datos y archivos de configuración correspondientes.
Flujo Alternativo: 3) El usuario podrá salir del formulario en cualquier momento con sólo elegir otra opción del menú de administración. 4) En caso de existir errores, el sistema mostrará los errores y regresará al paso 2.	
Poscondiciones: Las acciones del usuario han sido guardadas correctamente.	

Nombre:	Administrar Tipos de Menús
Descripción En este caso de uso se detallará la ubicación y cantidad de los menús, como ser superior, lateral, inferior. Estas categorías son con fines de organización y personalización del sistema.	
Actores: Super-Administrador	
Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.	
Flujo Normal:	
Acción de los Actores	Respuesta del Sistema
1) El caso de uso inicia cuando el usuario hace click en la opción “administrar menú” 3) El usuario hace click sobre la acción que desea realizar.	2) El sistema muestra un listado de los menús existentes con su respectiva ubicación (superior, inferior, izquierda o derecha) • Nuevo • Eliminar • Activar/Desactivar 4) El sistema recupera la acción y muestra la página

5) El usuario realiza las acciones deseadas y hace click en la opción “guardar”.	correspondiente. 6) El sistema verifica que los datos sean correctos y guarda los datos, regresa al paso 2.
Flujo Alternativo: 3) El usuario podrá salir del formulario en cualquier momento con sólo elegir otra opción del menú de administración. 6) En caso de existir errores, el sistema mostrará los errores y regresará al paso 2.	
Poscondiciones: Las acciones fueron guardadas correctamente.	

Nombre:	Realizar Búsquedas
Descripción Este caso de uso permite buscar información en la página a través de criterios de búsquedas, a través del cual se mostraran todos los contenidos que coincidan con el criterio introducido.	
Actores: Usuario	

Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.	
Flujo Normal:	
Acción de los Actores	Respuesta del Sistema
1) El caso de uso se encuentra presente en la parte pública del sistema, desde el cual el usuario podrá ingresar texto para buscar en el sistema.	2) El sistema, en base al criterio introducido, buscará las coincidencias y listará el contenido encontrado.
3) El usuario hará click sobre el contenido que le interese.	4) El sistema mostrará el contenido seleccionado.
Flujo Alternativo: 2) En caso de no encontrar resultados, se mostrará un mensaje indicando que no se encontraron resultados y se podrá repetir la búsqueda. 3) El usuario podrá hacer nuevas búsquedas y el procedimiento se repite retornando al paso 2.	
Poscondiciones: Las páginas fueron encontradas exitosamente.	

Nombre:	Participar en foro
Descripción Este caso uso permite que el usuario tendrá la opción de ver el foro y en caso de inscribirse podrá participar en él.	
Actores: Usuario	
Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.	
Flujo Normal:	
Acción de los Actores	Respuesta del Sistema
1) El caso de uso inicia cuando el usuario hace click en el menú “Foro”. 3) El usuario podrá: 3.1) Navegar y ver las opiniones y mensajes del foro. 3.2) Ingresar al foro con nombre de usuario y contraseña. 3.3) Registrarse al foro 4) El usuario hace click en la opción responder (sólo si ingresó a la opción 3.3).	2) El sistema le mostrará el listado de foros, con sus respectivos temas y la cantidad de mensajes.

6) El usuario ingresará su mensaje y hará click en “enviar”.	5) El sistema le mostrará una pantalla con un editor para ingresar el mensaje. 7) El sistema valida y guarda los datos.
Flujo Alternativo: 3) El usuario podrá alternar entre las alternativas 3.1, 3.2 y 3.3.	
Poscondiciones:	

Nombre:	Solicitar Información
Descripción Se le ofrecerá la opción de llenar un formulario donde podrá escribir su consulta y dejar su correo electrónico al cual se le enviará la respuesta a la pregunta formulada.	
Actores: Usuario	
Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.	
Flujo Normal:	
Acción de los Actores	Respuesta del Sistema
1) El caso de uso inicia cuando el usuario hace click	

en “solicitar información”. 3) El usuario introduce los datos requeridos y hace click en “enviar”.	2) El sistema mostrará un formulario solicitando los siguientes datos. • Nombre • Apellidos • Correo Electrónico • Mensaje 4) El sistema valida los datos y los guarda en la base de datos. El sistema muestra un mensaje de confirmación de envío.
Poscondiciones: Los datos han sido enviados correctamente.	
Nombre:	Navegar por el Sitio
Descripción Se mostrará los menús, enlaces, noticias, eventos, preguntas frecuentes, foros y demás información que sea de carácter público.	
Actores: Usuario	
Precondiciones: Estar logueado en el sistema y tener el permiso correspondiente.	
Flujo Normal:	

Acción de los Actores	Respuesta del Sistema
1) El caso de uso inicia cuando el usuario ingresa al sistema web. 3) El usuario hará click sobre el enlace deseado. El usuario podrá realizar el mismo proceso las veces que desee.	2) El sistema cargará el menú, enlaces, noticias, eventos y las opciones que sean de acceso público. 4) El sistema mostrará el contenido del enlace seleccionado.
Flujo Alternativo: El usuario puede salir del sistema con sólo cerrar el navegador.	

5.1. ANÁLISIS

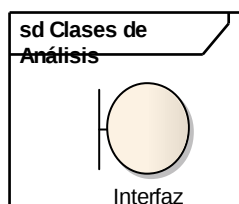
Durante el análisis se analizan los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos con el objetivo de conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que nos ayude a estructurar el sistema entero.

5.2. MODELO DE ANÁLISIS

El modelo análisis se representa mediante un sistema de análisis que denota el paquete de más alto nivel del modelo. La utilización de otros paquetes de análisis es por tanto una forma de organizar el modelo de análisis en partes más manejables que representan abstracciones de clases y posiblemente de subsistemas de diseño del sistema. Dentro del modelo de análisis, los casos de uso se describen mediante clases de análisis y sus objetos. Esto se representa mediante colaboraciones dentro del modelo a análisis que llamamos realizaciones de caso de uso.

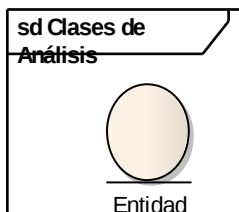
5.3. CLASE DE ANÁLISIS

5.3.1. CLASES DE INTERFAZ



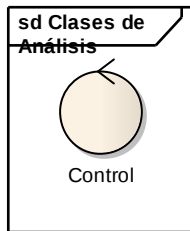
Se utilizan para modelar la interacción entre el sistema y sus actores (es decir, usuarios y sistemas externos). Esta interacción a menudo implica recibir (y presentar) información y peticiones de (y hacia) los usuarios y los sistemas externos.

5.3.2. CLASES DE ENTIDAD



Se utilizan para modelar información que posee una vida larga y que es a menudo persistente. Modelan la información y el comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto del mundo real, o un suceso del mundo real.

5.3.3. CLASES DE CONTROL



Representan coordinación, secuencia, transacciones y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto. También se utilizan para representar derivaciones y cálculos complejos.

5.4. REALIZACIÓN DE CASOS DE USO

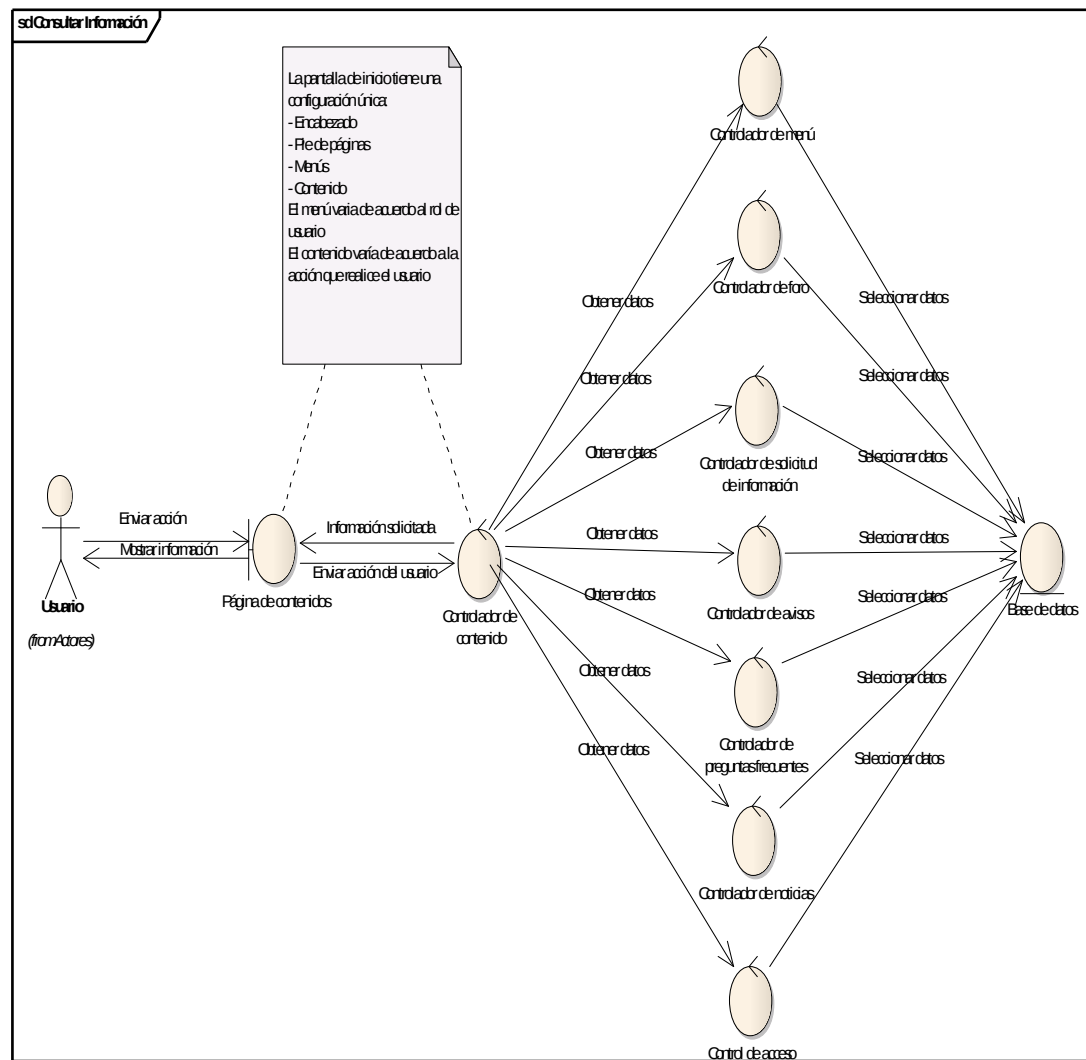
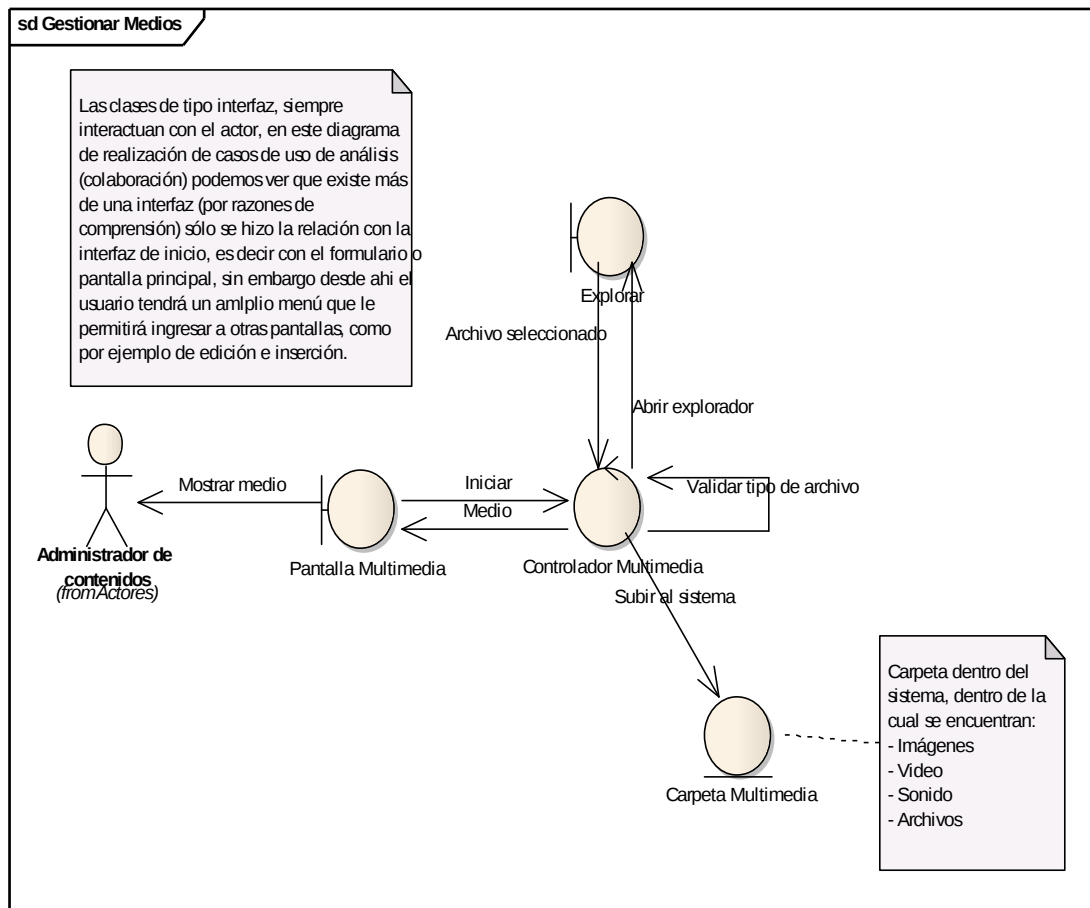
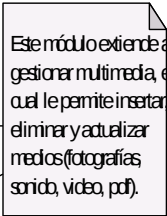


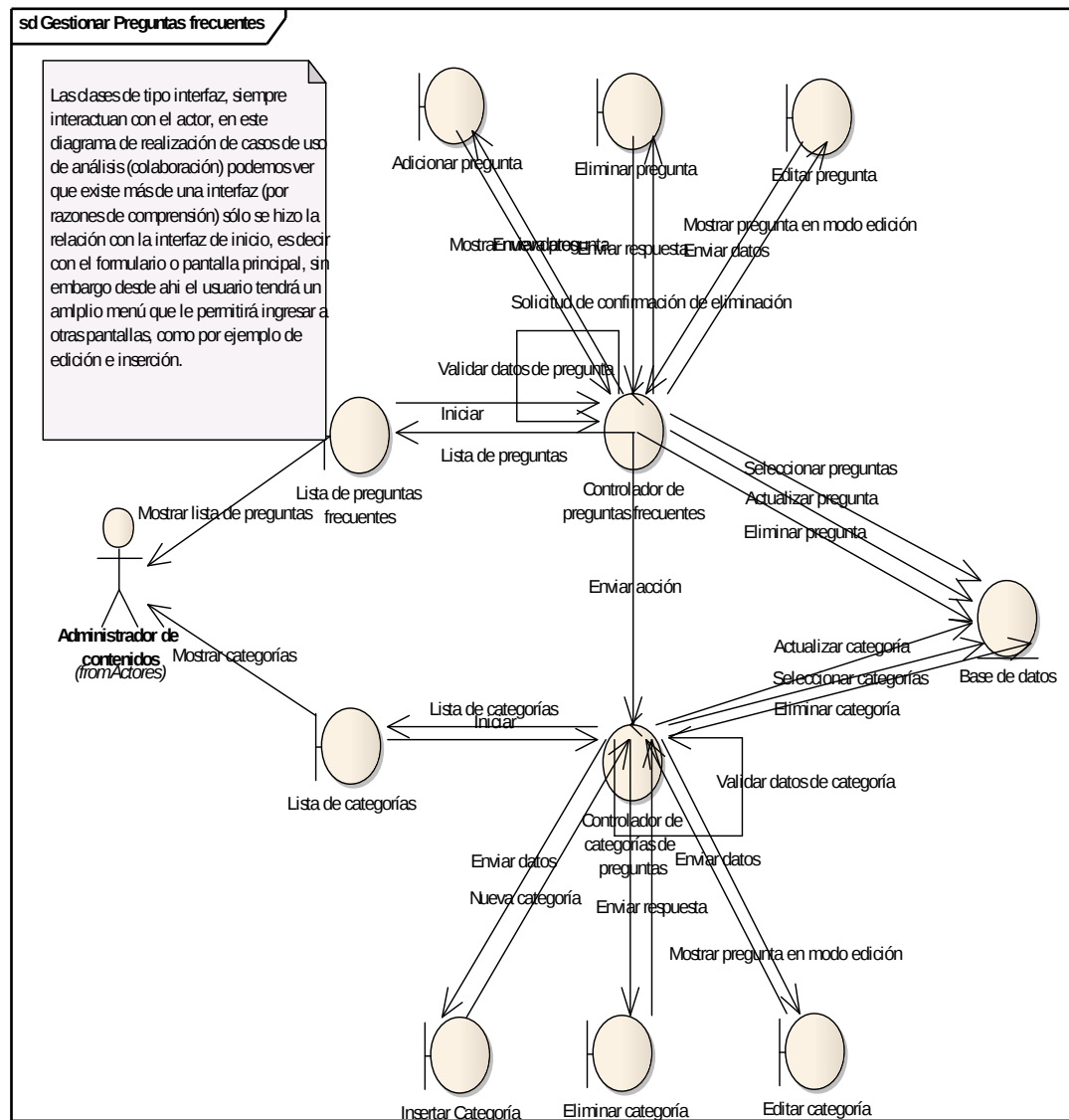
Figura 19: Diagrama de Colaboración que Muestra la Realización del Caso de Uso Consultar Información



**Figura 20: Diagrama de Colaboración que Muestra la Realización del
Caso de Uso Gestionar Medios**



Caso de Uso Gestionar Noticias



**Figura 22: Diagrama de Colaboración que Muestra la Realización del
Caso de Uso Gestionar Preguntas Frecuentes**

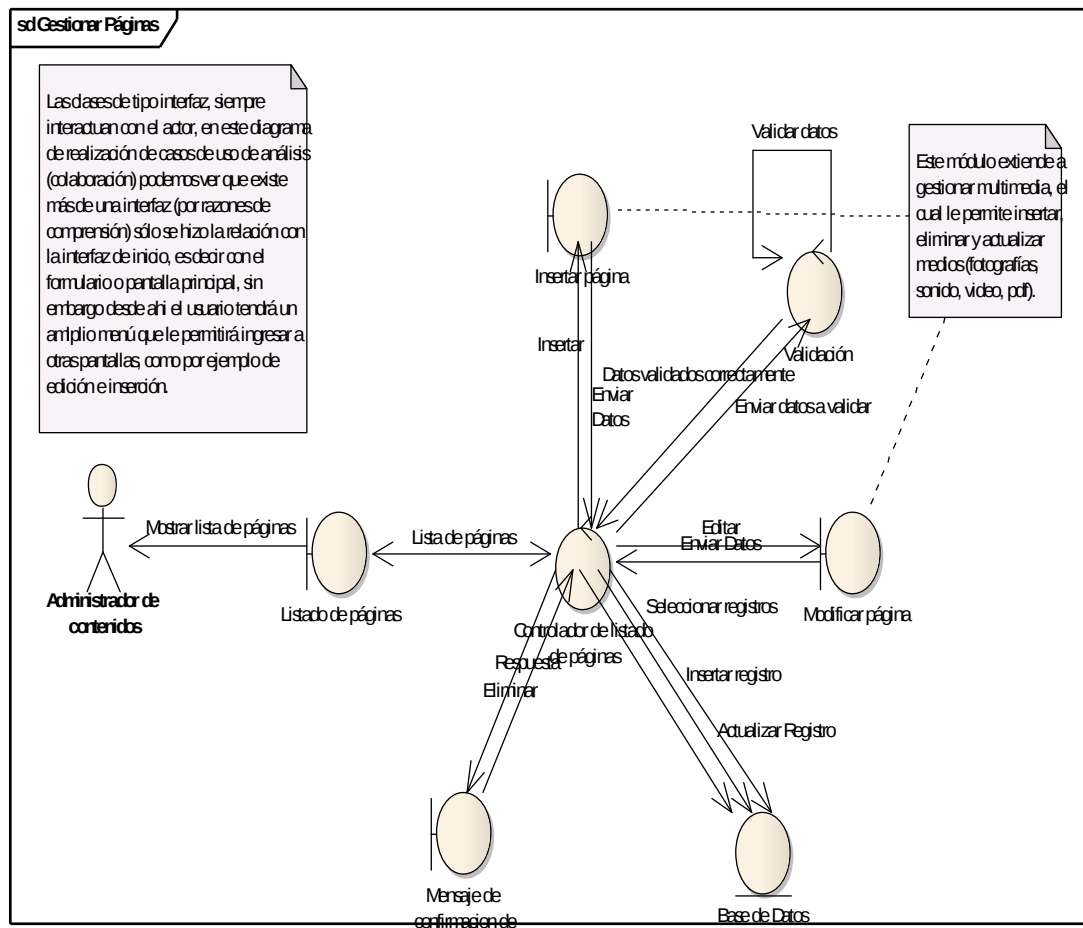
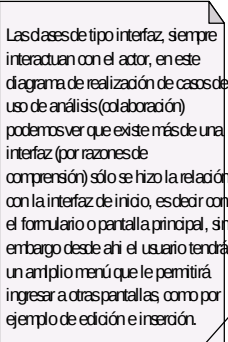


Figura 23: Diagrama de Colaboración que Muestra la Realización del

Caso de Uso Gestionar Páginas



Caso de Uso Gestionar Archivos

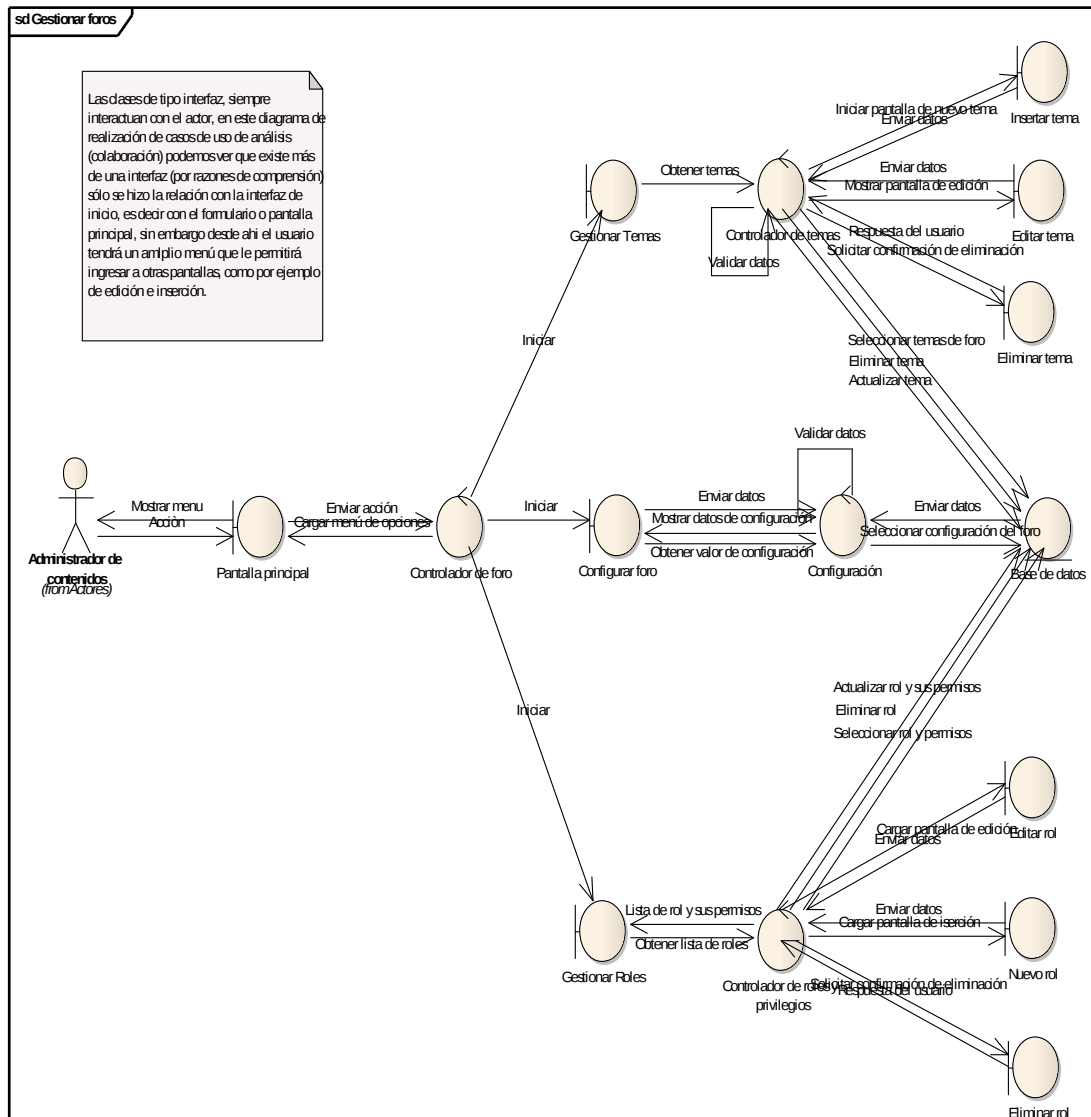


Figura 25: Diagrama de Colaboración que Muestra la Realización del Caso de Uso Gestionar Foro

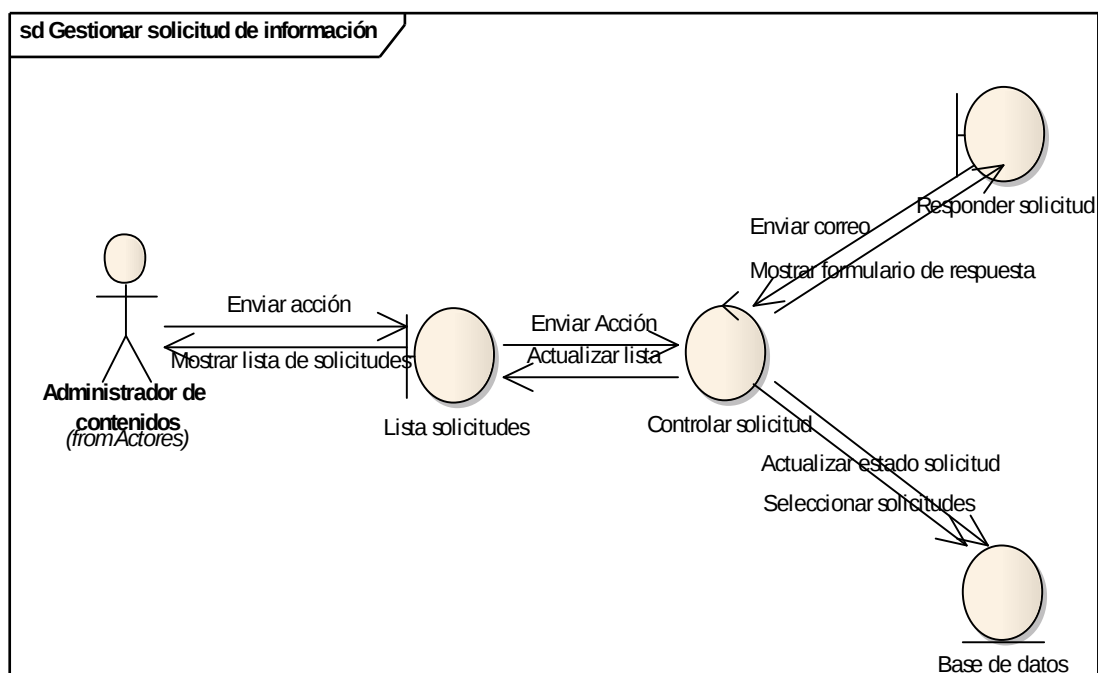


Figura 26: Diagrama de Colaboración que Muestra la Realización del Caso de Uso Gestionar Solicitud de Información

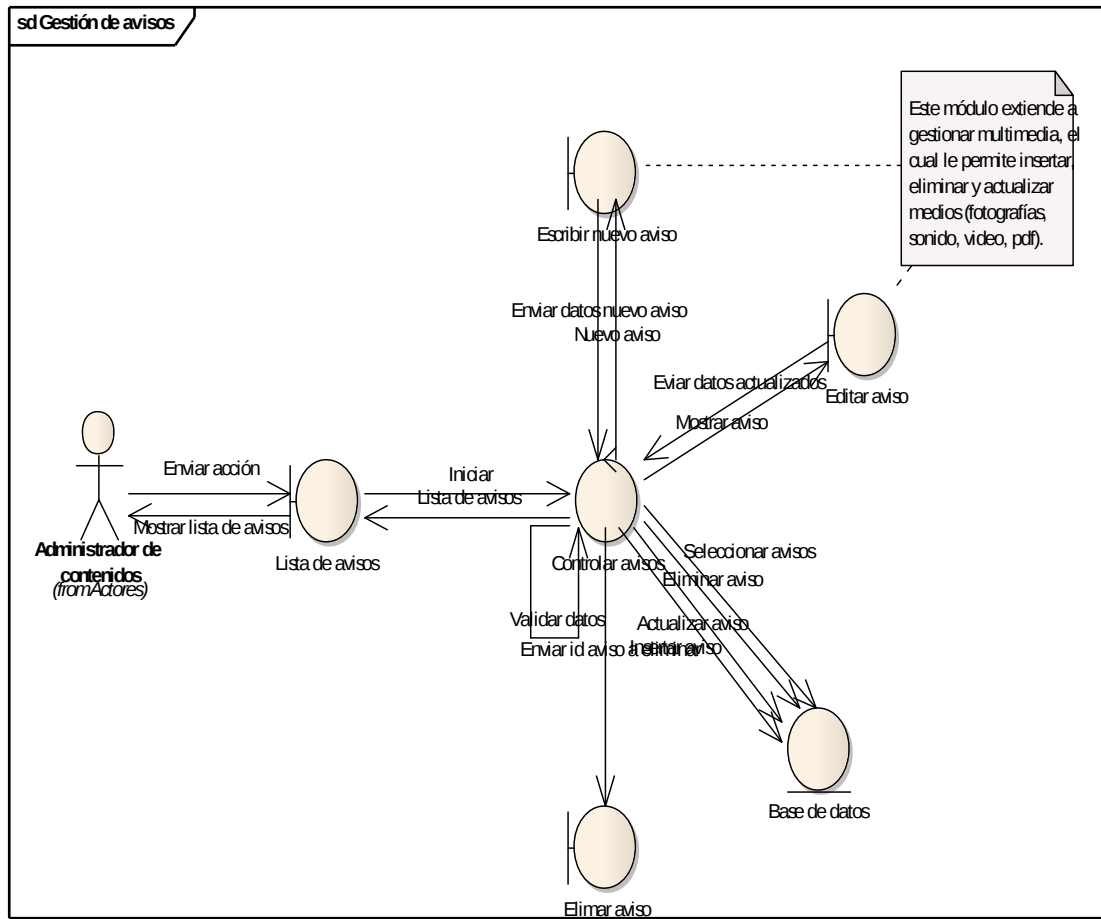


Figura 27: Diagrama de Colaboración que Muestra la Realización del Caso de Uso Gestionar Avisos

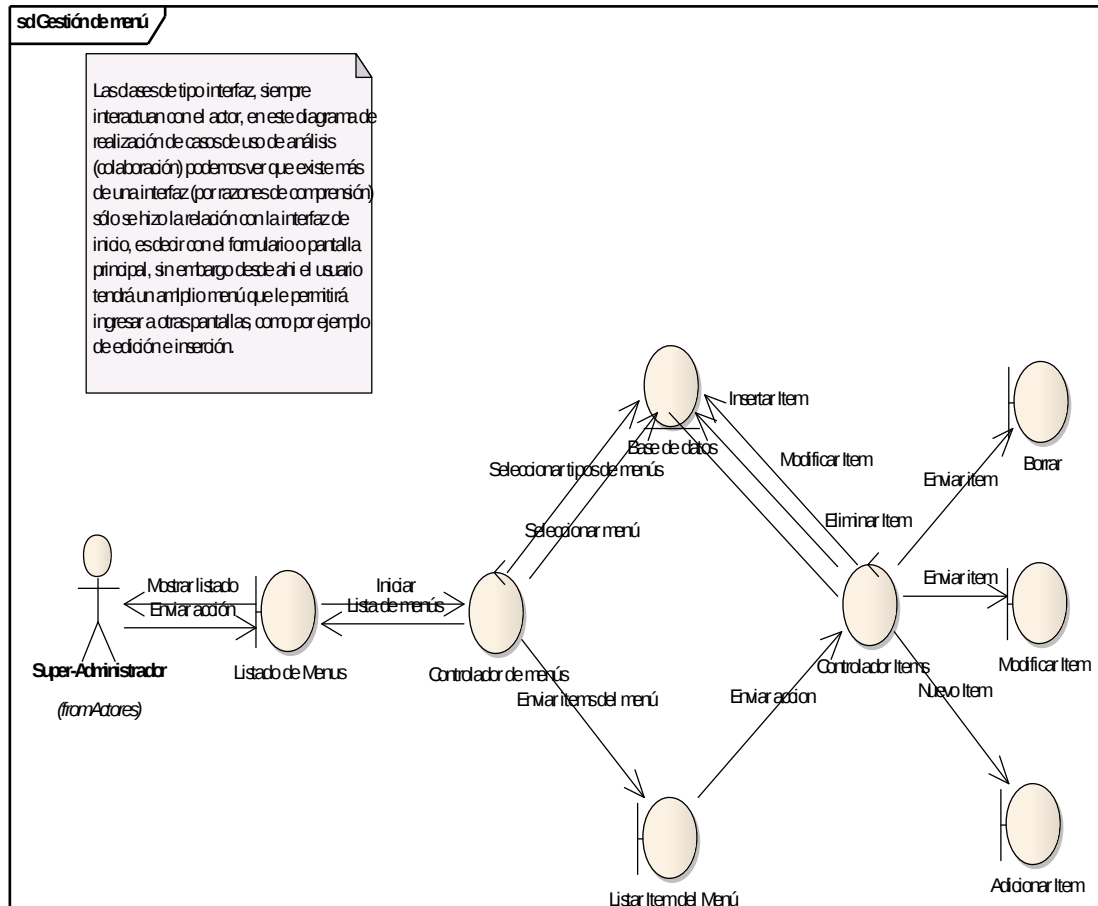


Figura 28: Diagrama de Colaboración que Muestra la Realización del Caso de Uso Gestionar Menús

5.5. DESCRIPCIÓN DE LA ARQUITECTURA

El presente sistema propone realizar un marco de trabajo que permita gestionar información dinámica de forma sencilla y amigable, ofreciendo la posibilidad a los administradores personalizar y organizar su información de acuerdo a los requerimientos de cada unidad, así mismo ofrecerá una plataforma de comunicación entre la institución y los internautas (usuarios del sistema).

Con este objetivo se han definido tres módulos que realizan tareas específicas destinadas a satisfacer los requerimientos de la institución:

- **Administrar Sistema**

Consiste en el módulo de administración y configuración del sistema, dónde los usuarios con rol de administrador general, podrán configurar el sitio, roles y permisos de usuarios, multimedia y panel de control del sistema.

- **Administrar Contenido del Sistema**

En este módulo se administra el contenido que se mostrará a los usuarios del sistema web, aquí los usuarios autorizados (jefes de unidad y responsables de servicios) podrán gestionar los foros del sistema, gestionar consultas, definir menús y enlaces para organizar y categorizar el contenido de las páginas a ser mostradas, subir imágenes, animaciones y texto del contenido de las páginas, se administraran avisos y anuncios de cada sección, administración de noticias, además podrán definir la duración de la publicación y si el contenido será público o privado.

- **Consultar Información**

Este módulo es el módulo público dirigido a los usuarios del sistema web, donde se mostrará toda la información cargada en el sistema de acuerdo a la organización, categorización y permisos definidos en el módulo de administración de contenidos, el usuario podrá navegar por cada una de las secciones, ver noticias, avisos y anuncios, ver preguntas frecuentes, participar en los foros y solicitar información.

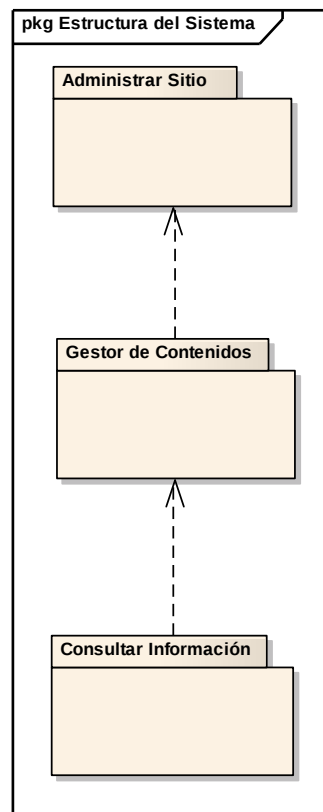


Figura 29: Diagrama de Paquetes que Muestra la Dependencia entre los Sub-Sistemas Identificados

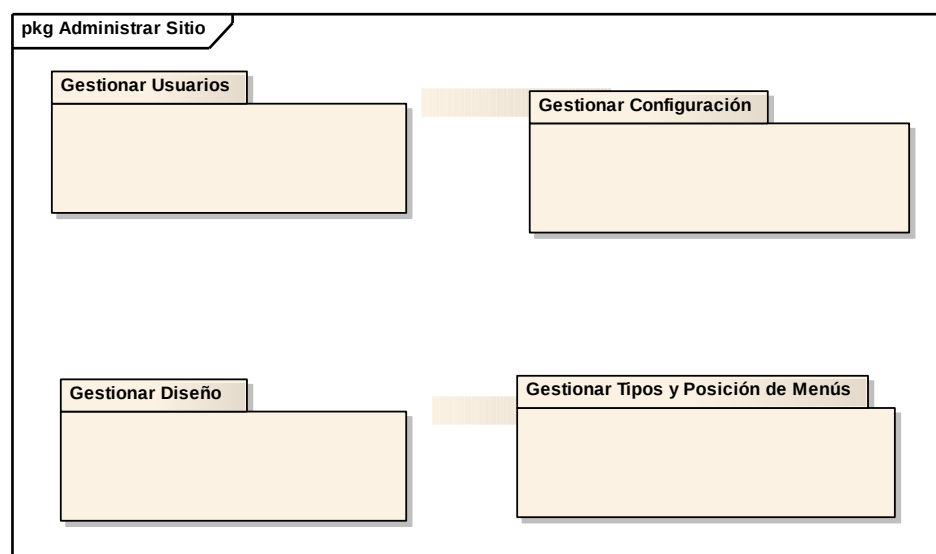


Figura 30: Sub-Sistema Administrar Sitio y sus Módulos que los Componen

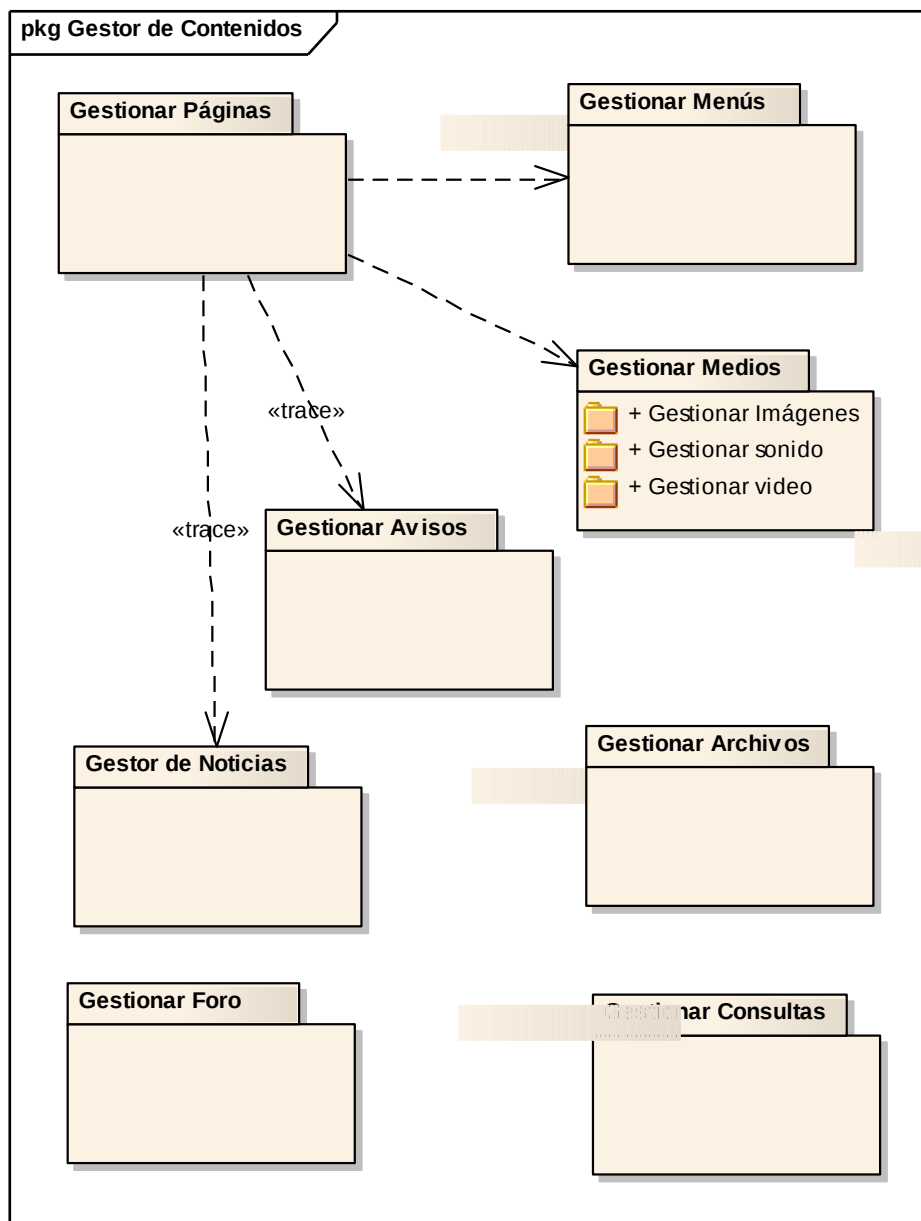


Figura 31: Sub-Sistema Gestor de Contenidos y sus Módulos que los Componen

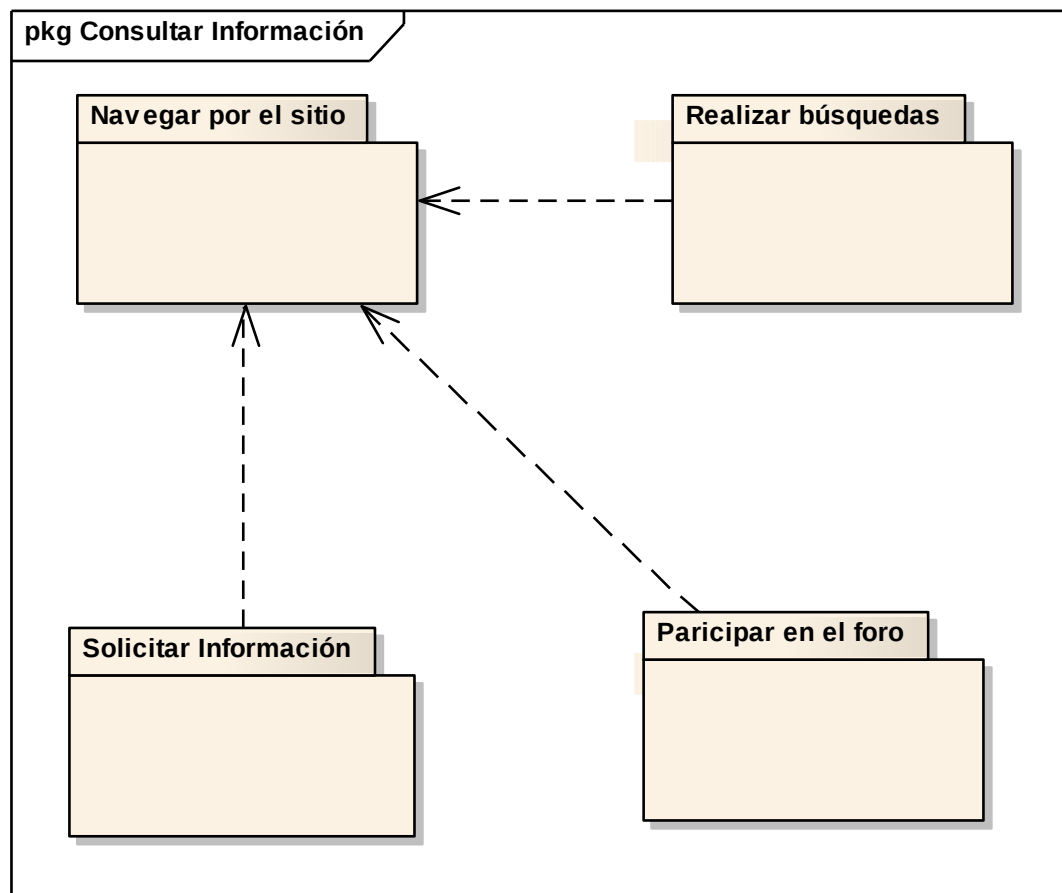


Figura 32: Sub-Sistema Consultar Información y sus Módulos que los Componen

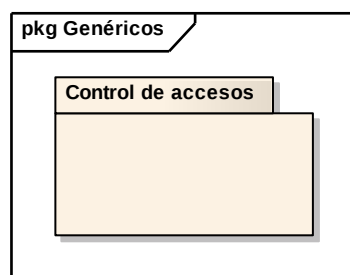


Figura 33: Sub-Sistema Control de Accesos

6.1. DISEÑO

En el diseño modelamos el sistema y encontramos su forma para que soporte los requisitos, incluyendo los requisitos no funcionales y otras restricciones. Consiste en un modelo físico, porque es un plano de la implementación.

6.2. DIAGRAMA DE CLASES

El Diagrama de clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones.

En la **(Figura 34)** se ilustra el diagrama de clases desde una perspectiva de especificación donde se muestran las clases, sus atributos y relaciones.

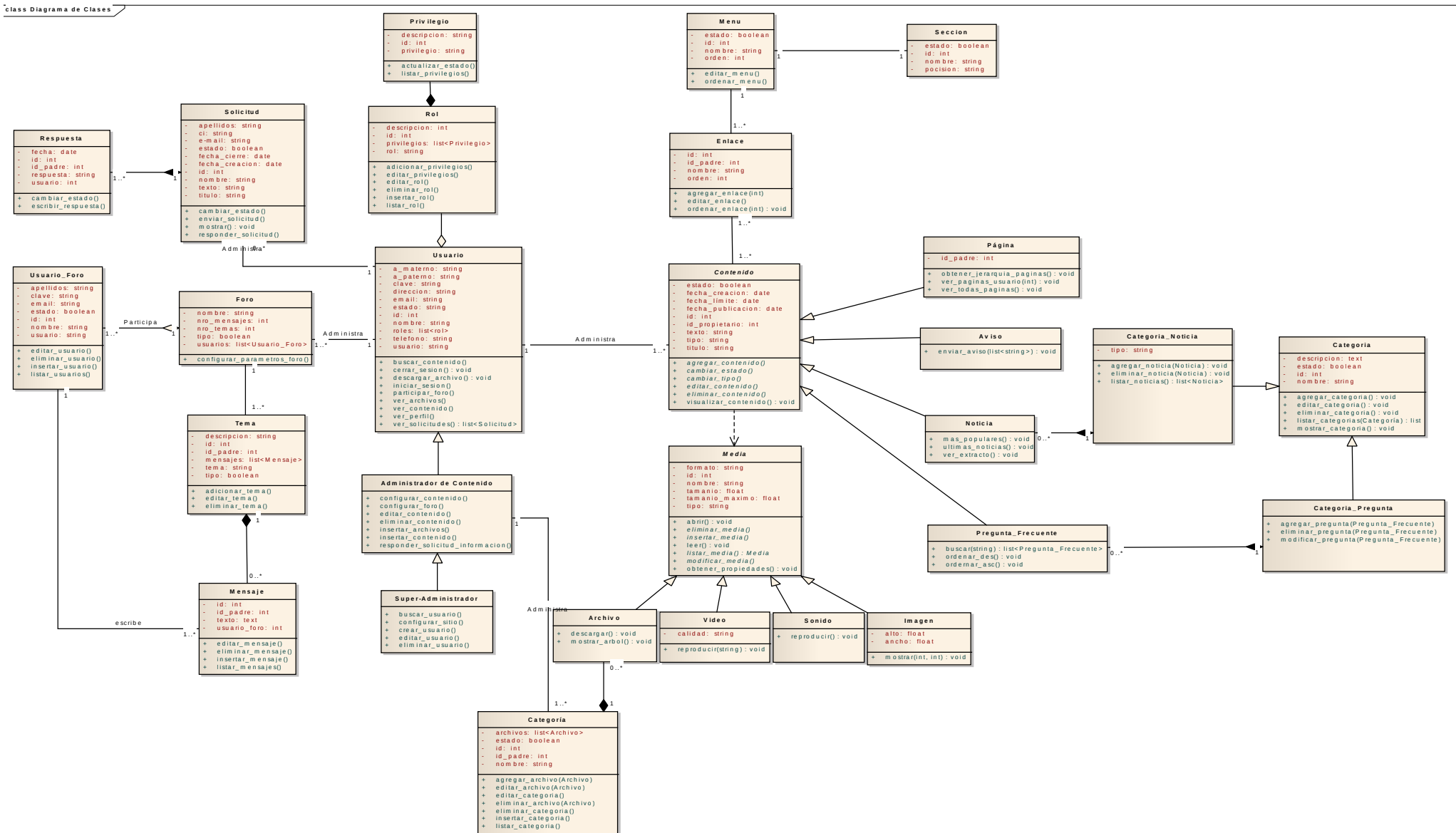


Figura 34: Diagrama de Clases

6.3. DIAGRAMA DE BASE DE DATOS

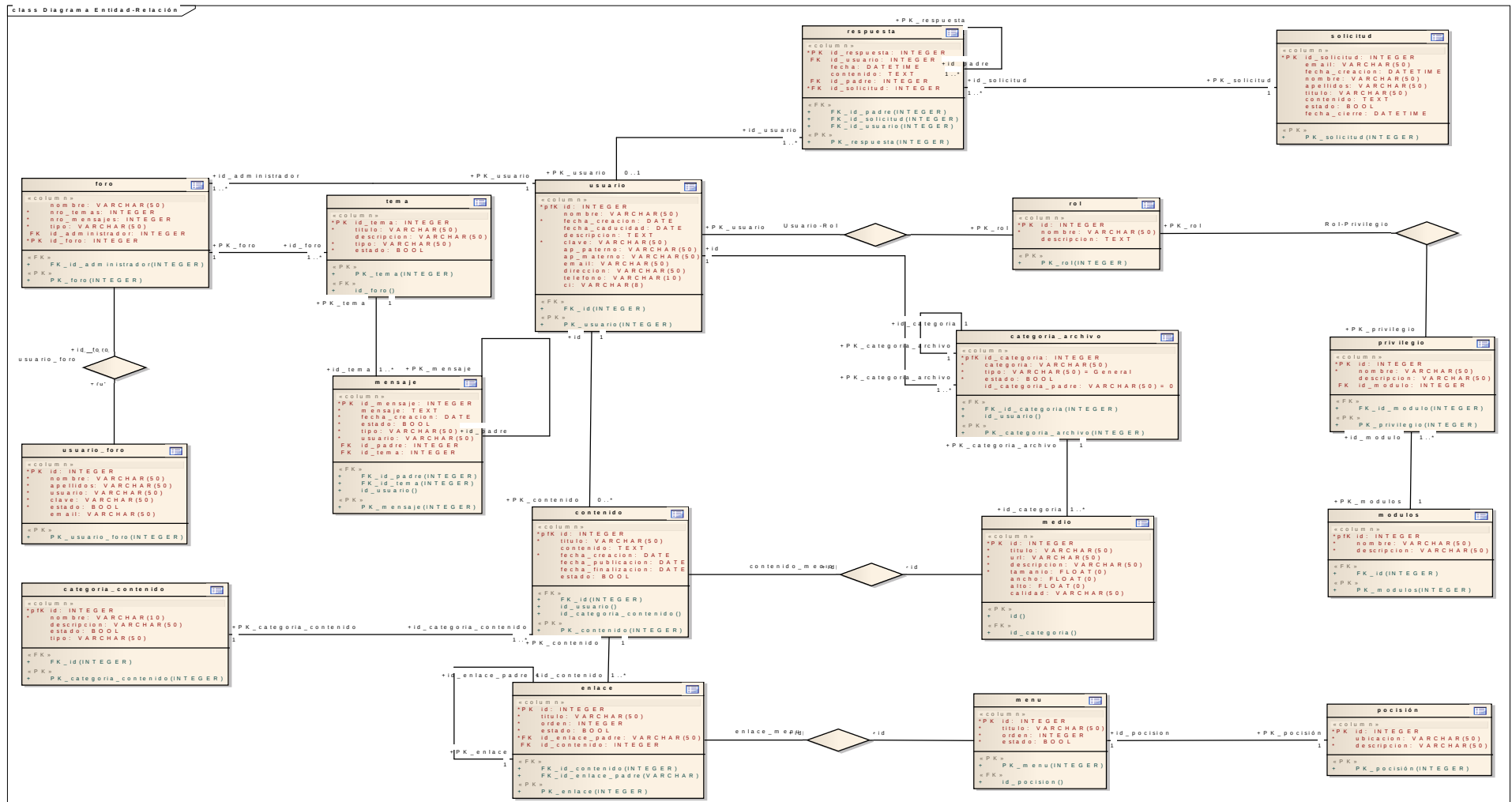


Figura 35: Diagrama de Base de Datos

6.3.1. CREACIÓN DE LA BASE DE DATOS EN SQL

USE hospital;

```
CREATE TABLE categoria_archivo(  
id_categoria INTEGER NOT NULL,  
categoria VARCHAR(50) NOT NULL,  
tipo VARCHAR(50) NOT NULL DEFAULT General,  
estado BOOL NOT NULL,  
id_categoria_padre VARCHAR(50) DEFAULT 0,  
PRIMARY KEY (id_categoria)  
KEY (id_categoria));
```

```
CREATE TABLE categoria_contenido(  
id INTEGER NOT NULL,  
nombre VARCHAR(10) NOT NULL,  
descripcion VARCHAR(50),  
estado BOOL,  
tipo VARCHAR(50),  
PRIMARY KEY (id),  
KEY (id));
```

```
CREATE TABLE contenido(  
id INTEGER NOT NULL,  
titulo VARCHAR(50) NOT NULL,  
contenido TEXT,  
fecha_creacion DATE NOT NULL,  
fecha_publicacion DATE,  
fecha_finalizacion DATE,  
estado BOOL,  
PRIMARY KEY (id),  
KEY (id));
```

```
CREATE TABLE enlace(  
id INTEGER NOT NULL,  
titulo VARCHAR(50) NOT NULL,  
orden INTEGER NOT NULL,  
estado BOOL NOT NULL,  
id_enlace_padre VARCHAR(50) NOT NULL,  
id_contenido INTEGER,  
PRIMARY KEY (id));  
KEY (id_contenido),  
KEY (id_enlace_padre));
```

```
CREATE TABLE foro(  
nombre VARCHAR(50),  
nro_temas INTEGER NOT NULL,  
nro_mensajes INTEGER NOT NULL,  
tipo VARCHAR(50) NOT NULL);  
id_administrador INTEGER,  
id_foro INTEGER NOT NULL,  
PRIMARY KEY (id_foro),  
KEY (id_administrador));
```

```
CREATE TABLE medio(  
id INTEGER NOT NULL,  
titulo VARCHAR(50) NOT NULL,  
url VARCHAR(50) NOT NULL,  
descripcion VARCHAR(50) NOT NULL,  
tamano FLOAT(0) NOT NULL,  
ancho FLOAT(0),  
alto FLOAT(0),  
calidad VARCHAR(50),  
PRIMARY KEY (id));
```

CREATE TABLE mensaje(

```
id_mensaje INTEGER NOT NULL,  
mensaje TEXT NOT NULL,  
fecha_creacion DATE NOT NULL,  
estado BOOL NOT NULL,  
tipo VARCHAR(50) NOT NULL,  
usuario VARCHAR(50) NOT NULL,  
id_padre INTEGER,  
id_tema INTEGER,  
PRIMARY KEY (id_mensaje),  
KEY (id_padre),  
KEY (id_tema));
```

CREATE TABLE modulos(

```
id INTEGER NOT NULL,  
nombre VARCHAR(50) NOT NULL,  
descripcion VARCHAR(50) NOT NULL,  
PRIMARY KEY (id)  
KEY (id));
```

CREATE TABLE menu(

```
id INTEGER NOT NULL,  
titulo VARCHAR(50) NOT NULL,  
orden INTEGER NOT NULL,  
estado BOOL NOT NULL,  
PRIMARY KEY (id));
```

CREATE TABLE posición(

```
id INTEGER NOT NULL,  
ubicacion VARCHAR(50) NOT NULL,  
descripcion VARCHAR(50) NOT NULL,
```

PRIMARY KEY (id));

```
CREATE TABLE privilegio(  
id INTEGER NOT NULL,  
nombre VARCHAR(50) NOT NULL,  
descripcion VARCHAR(50),  
id_modulo INTEGER,  
PRIMARY KEY (id));  
KEY (id_modulo));
```

```
CREATE TABLE respuesta(  
id_respuesta INTEGER NOT NULL,  
id_usuario INTEGER,  
fecha DATETIME,  
contenido TEXT,  
id_padre INTEGER,  
id_solicitud INTEGER NOT NULL,  
PRIMARY KEY (id_respuesta),  
KEY (id_padre),  
KEY (id_solicitud),  
KEY (id_usuario));
```

```
CREATE TABLE rol(  
id INTEGER NOT NULL,  
nombre VARCHAR(50) NOT NULL,  
descripcion TEXT,  
PRIMARY KEY (id));
```

```
CREATE TABLE solicitud(  
id_solicitud INTEGER NOT NULL,  
email VARCHAR(50),
```

```
fecha_creacion DATETIME,  
nombre VARCHAR(50),  
apellidos VARCHAR(50),  
titulo VARCHAR(50),  
contenido TEXT,  
estado BOOL,  
fecha_cierre DATETIME,  
PRIMARY KEY (id_solicitud));
```

```
CREATE TABLE tema(  
id_tema INTEGER NOT NULL,  
titulo VARCHAR(50) NOT NULL,  
descripcion VARCHAR(50),  
tipo VARCHAR(50) NOT NULL,  
estado BOOL NOT NULL,  
PRIMARY KEY (id_tema));
```

```
CREATE TABLE usuario (  
id INTEGER NOT NULL,  
nombre VARCHAR(50),  
fecha_creacion DATE NOT NULL,  
fecha_caducidad DATE,  
descripcion TEXT,  
clave VARCHAR(50) NOT NULL,  
ap_paterno VARCHAR(50),  
ap_materno VARCHAR(50),  
email VARCHAR(50),  
direccion VARCHAR(50),  
telefono VARCHAR(10),  
ci VARCHAR(8),  
PRIMARY KEY (id),
```

KEY (id)) ;

```
CREATE TABLE usuario_foro(  
id INTEGER NOT NULL,  
nombre VARCHAR(50) NOT NULL,  
apellidos VARCHAR(50) NOT NULL,  
usuario VARCHAR(50) NOT NULL,  
clave VARCHAR(50) NOT NULL,  
estado BOOL NOT NULL,  
email VARCHAR(50),  
PRIMARY KEY (id));
```

```
CREATE TABLE usuario_rol(  
id_usuario INTEGER NOT NULL,  
id_rol BIGINT NOT NULL);
```

6.4. DIAGRAMA DE PAQUETES

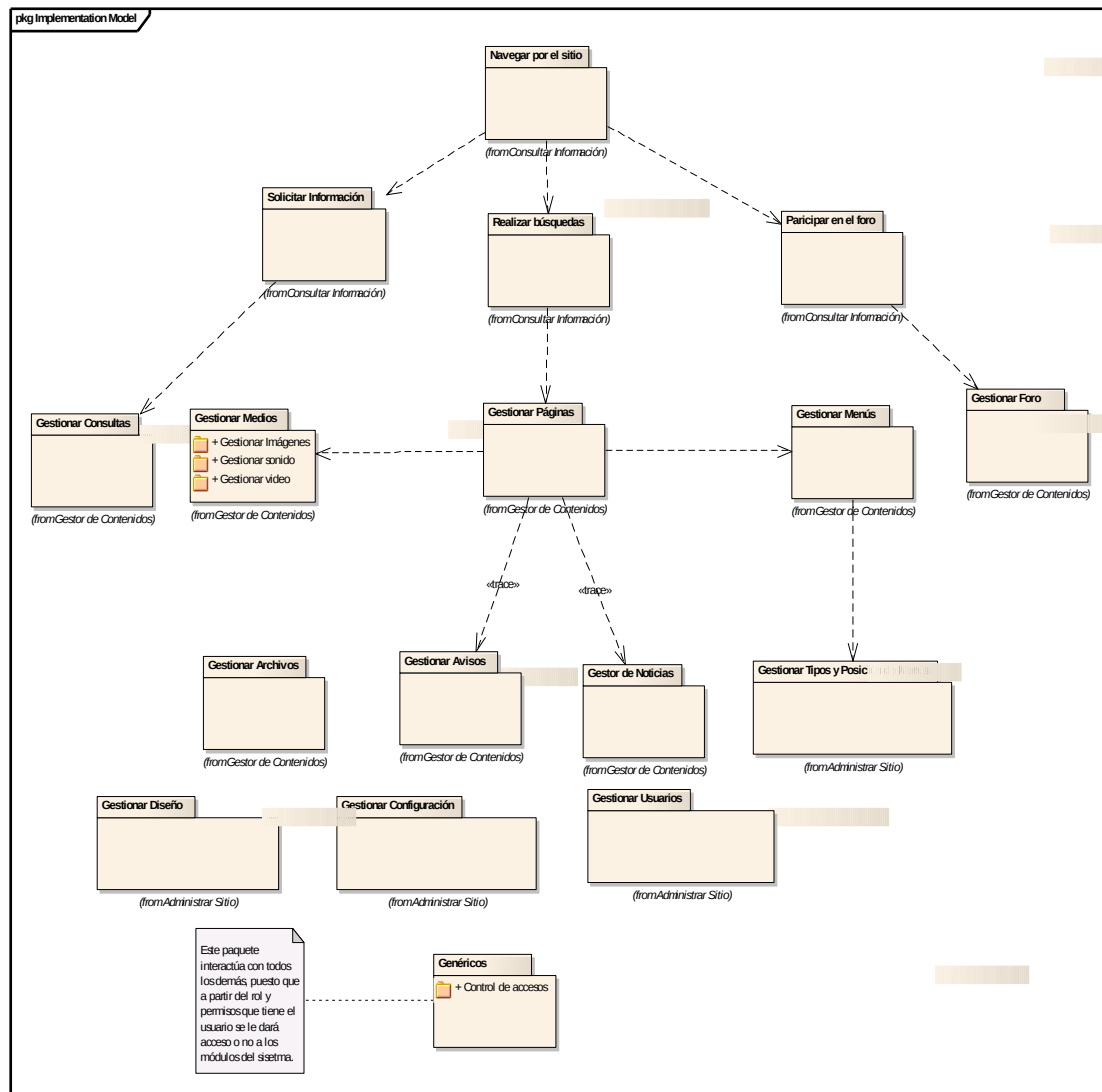


Figura 36: Diagrama de Paquetes que Muestra las Relaciones entre los Módulos del Sistema

6.5. DESCRIPCIÓN DE LA ARQUITECTURA

Se seleccionó el patrón de layers (capas) o tiers (niveles) ya que éste permite dar soporte a la arquitectura MVC utilizada para la implementación del software. Este patrón permite la descomposición de una aplicación en un conjunto de capas independientes y ordenadas jerárquicamente. Cada nivel o capa utiliza los servicios de la capa inmediatamente inferior y ofrece sus

servicios a la capa inmediatamente superior. Para la implementación de este software se ha utilizado específicamente el patrón arquitectónico de 3 capas, ya que es el que se ajusta con el patrón MVC, por proponer la siguiente separación:

- **Capa de Presentación**

Reside en el cliente y es la encargada de presentar la interfaz con la que interactuará el usuario.

- **Capa Lógica**

Reside en un nivel intermedio entre el cliente y el servidor de base de datos, y es la encargada de planificar y priorizar las solicitudes de información. En este caso, por tratarse de una aplicación Web, utilizaremos un Servidor de Aplicaciones Web.

- **Capa de Datos**

Esta capa reside en el nivel de recursos, el cual está compuesto por tantos recursos como el sistema pretende integrar. Normalmente se aloja en el Servidor de Base de Datos.



Figura 37: Modelo de la Arquitectura de Tres Capas

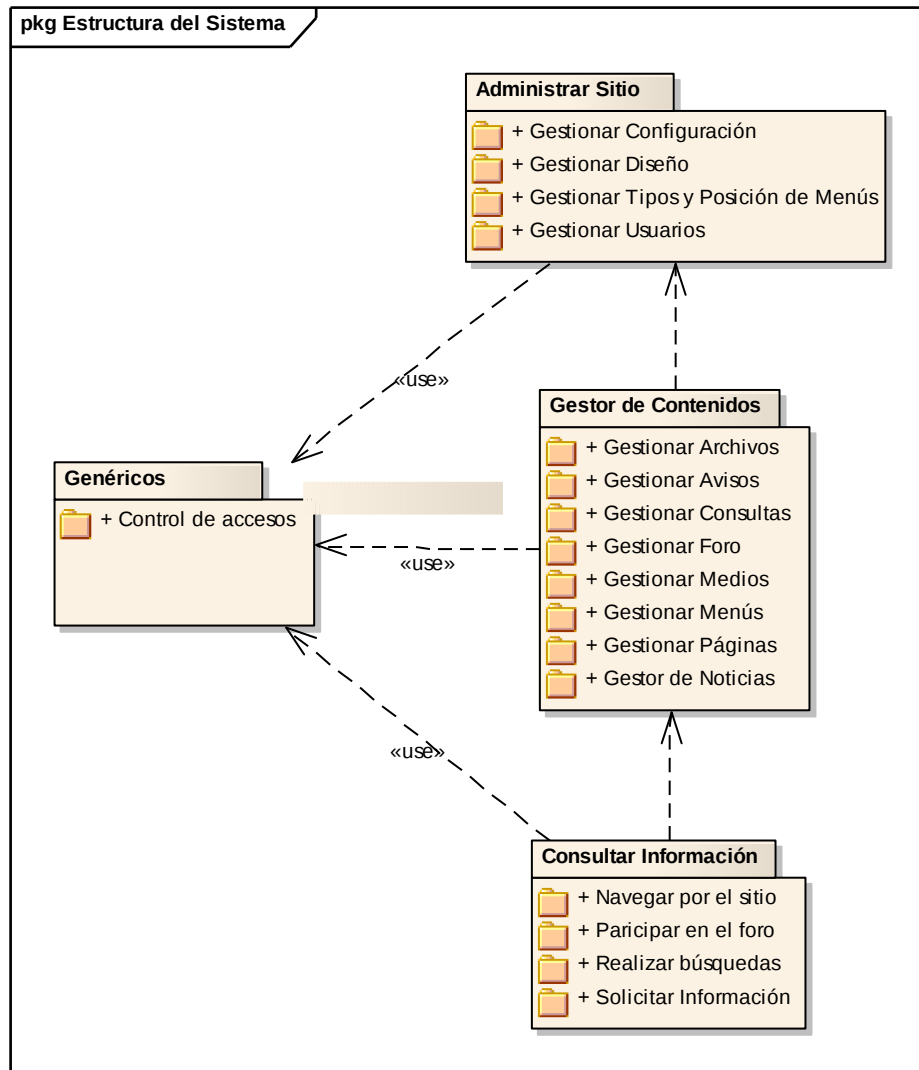


Figura 38: Diagrama de Paquetes que Muestra los Módulos del Sistema y sus Relaciones

6.6. PROTOTIPO DE INTERFAZ DE USUARIO

En esta sección se mostrarán las pantallas principales del sistema, las interfaces (pantallas, formularios, mensajes y reportes) fueron obtenidas a partir de los diagramas de casos de uso, descripción de casos de uso y de las clases de análisis de interfaz.

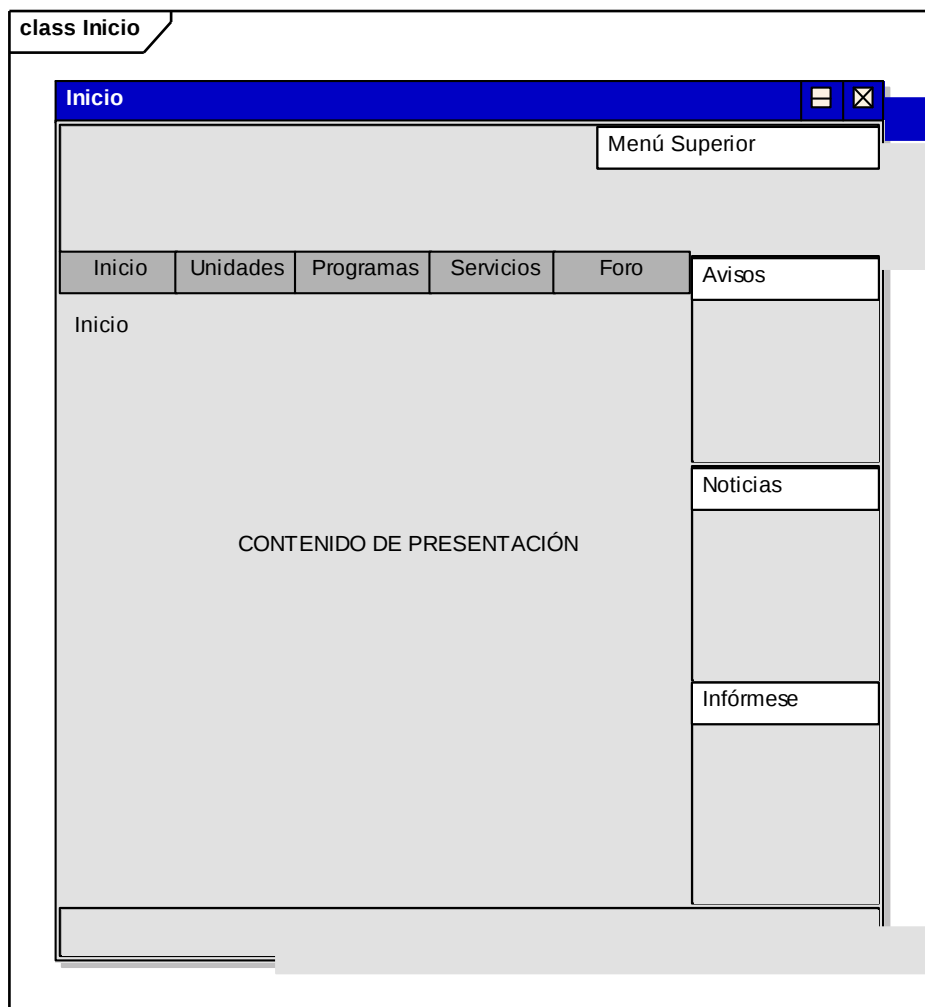


Figura 39: Pantalla de Inicio

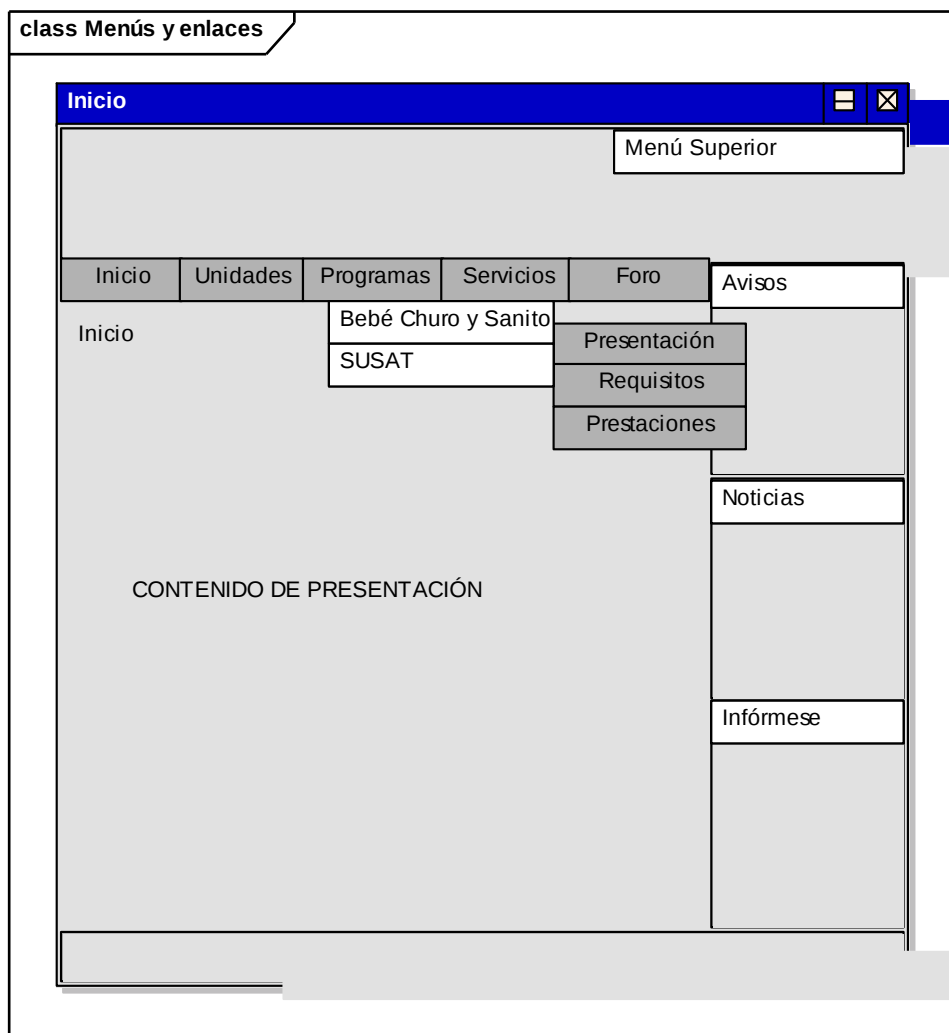


Figura 40: Pantalla que Muestra los Menús y Enlaces

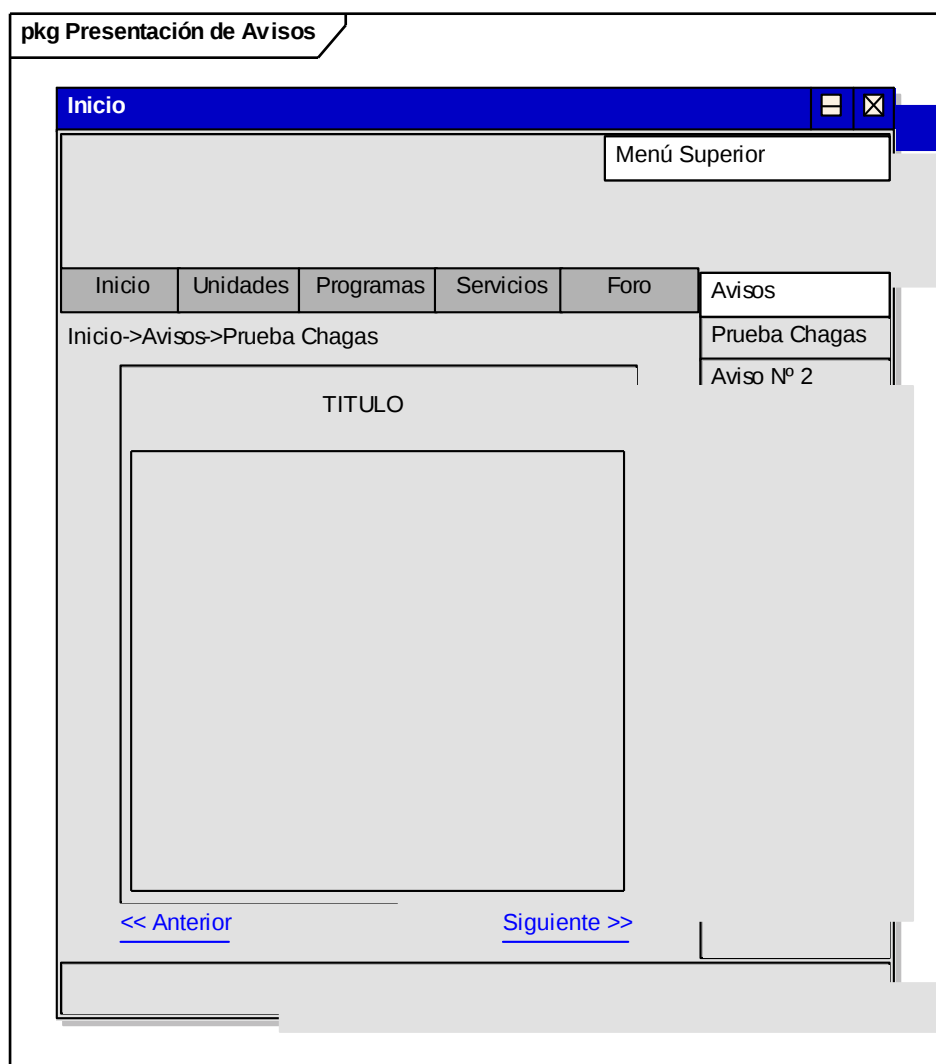


Figura 41: Pantalla de Presentación de Avisos

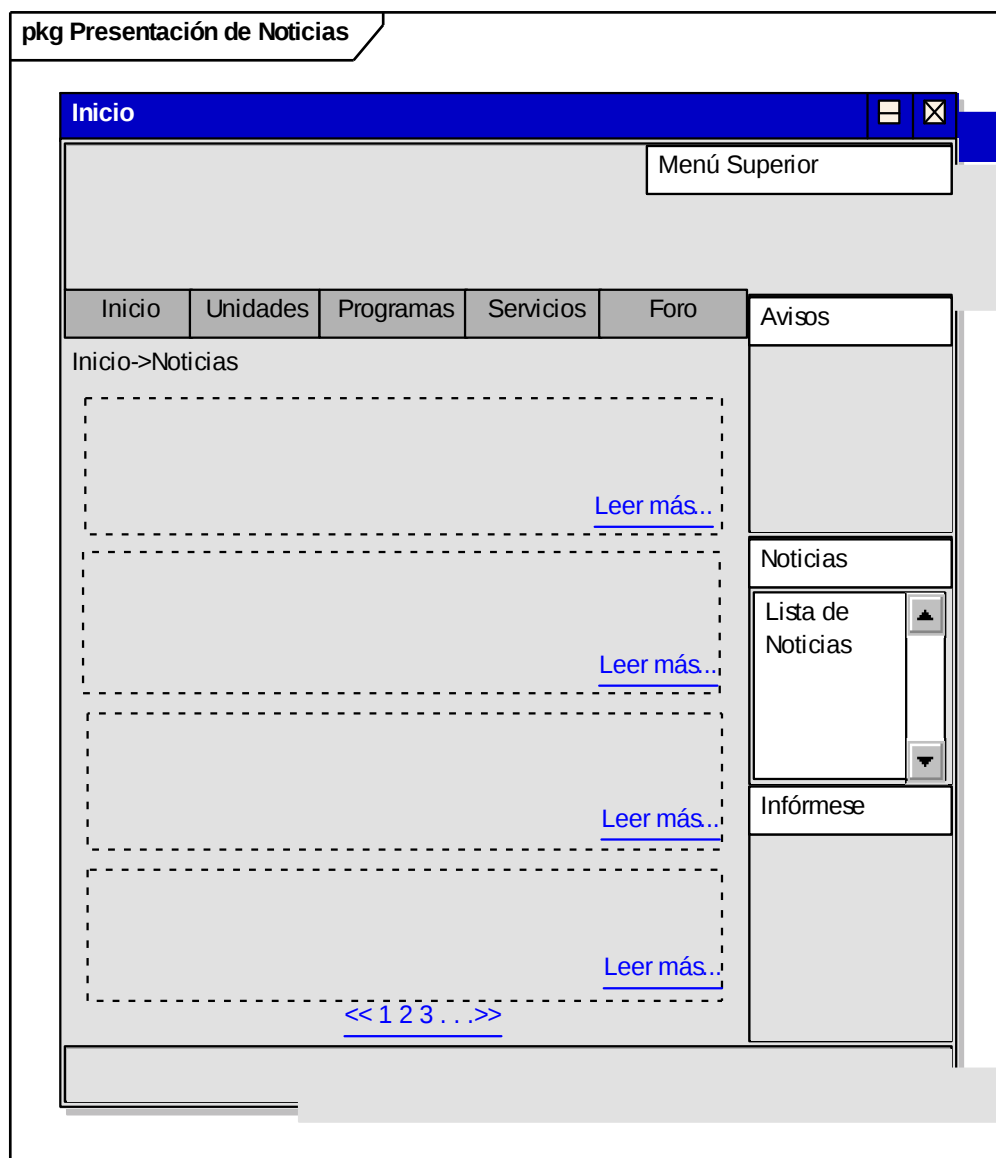


Figura 42: Pantalla de Presentación de Noticias

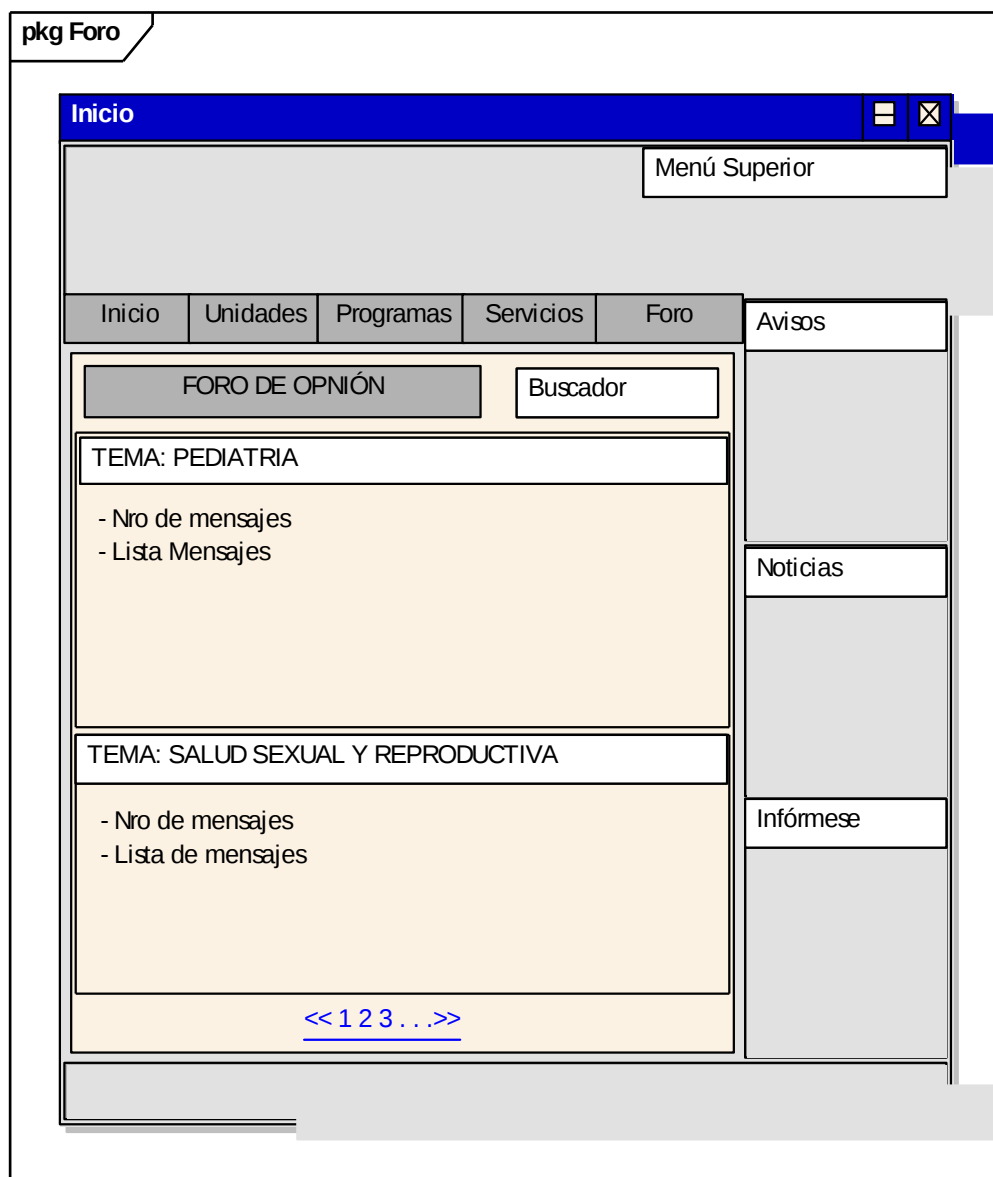


Figura 43: Pantalla de Participar en Foro (parte pública)

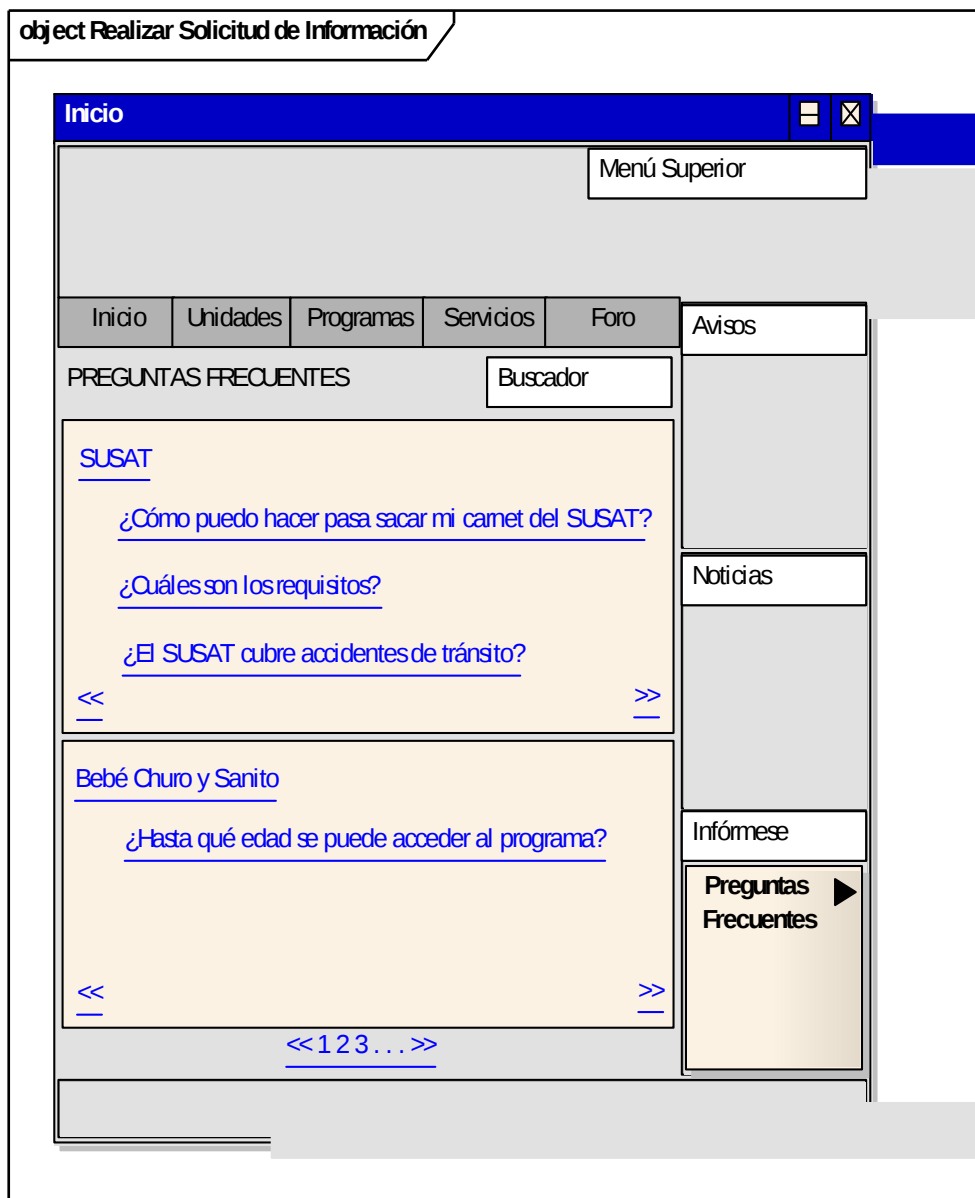


Figura 44: Pantalla de Preguntas Frecuentes

object Solicitar Información

Inicio

Menú Superior

Inicio

Unidades

Programas

Servicios

Foro

Avisos

REALIZAR SOLICITUD DE INFORMACIÓN

Nombre

Apellidos

Correo Electrónico

Escribir Consulta:

Enviar

Noticias

Infórmese

Figura 45: Pantalla de Solicitud de Información

object Ingresar al Sistema

Ingresar al sistema

ADMINISTRACIÓN DEL SISTEMA

Nombre de Usuario

Contraseña

Aceptar

Figura 46: Pantalla de Ingreso al Sistema

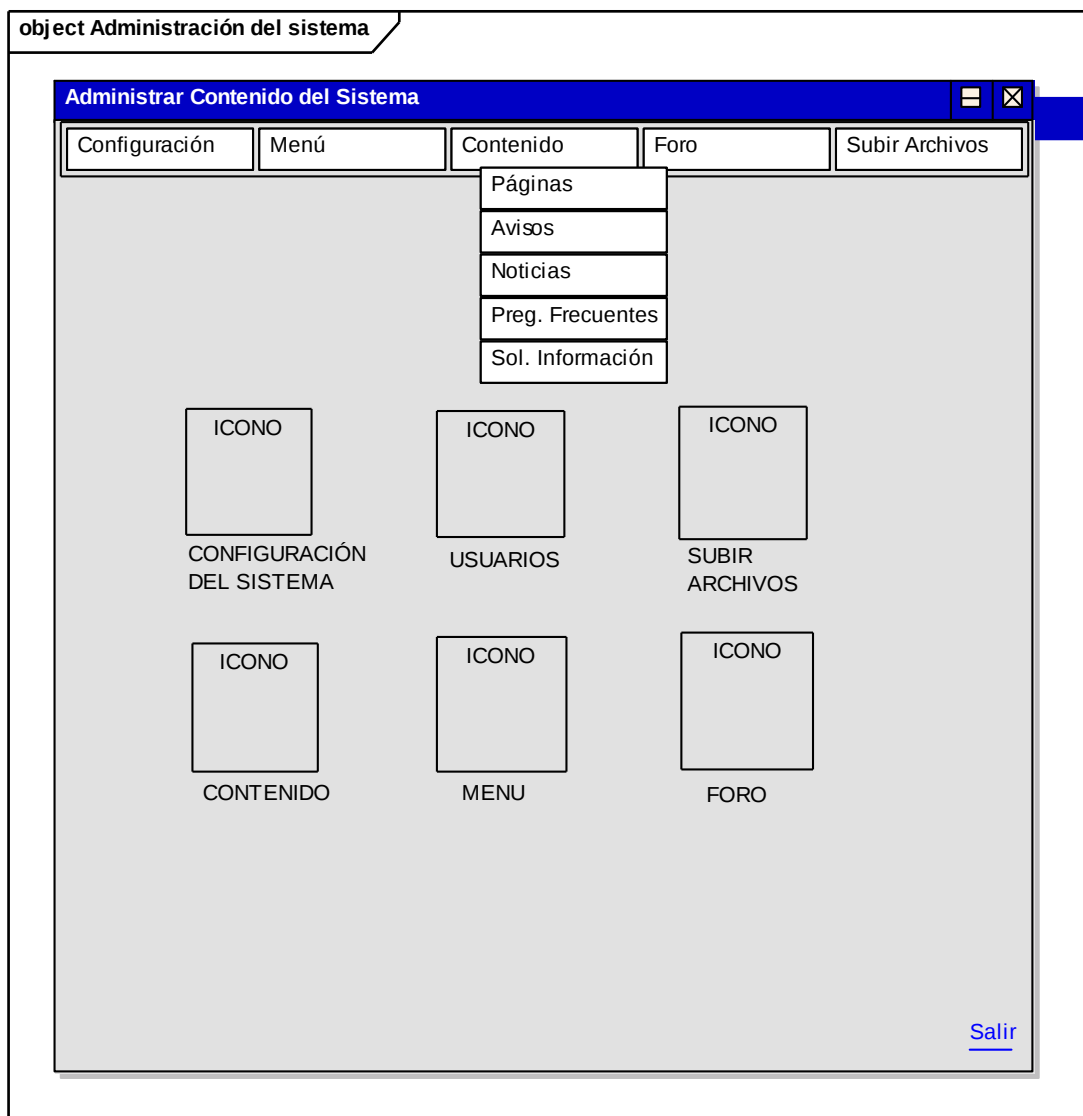


Figura 47: Pantalla del Administrador

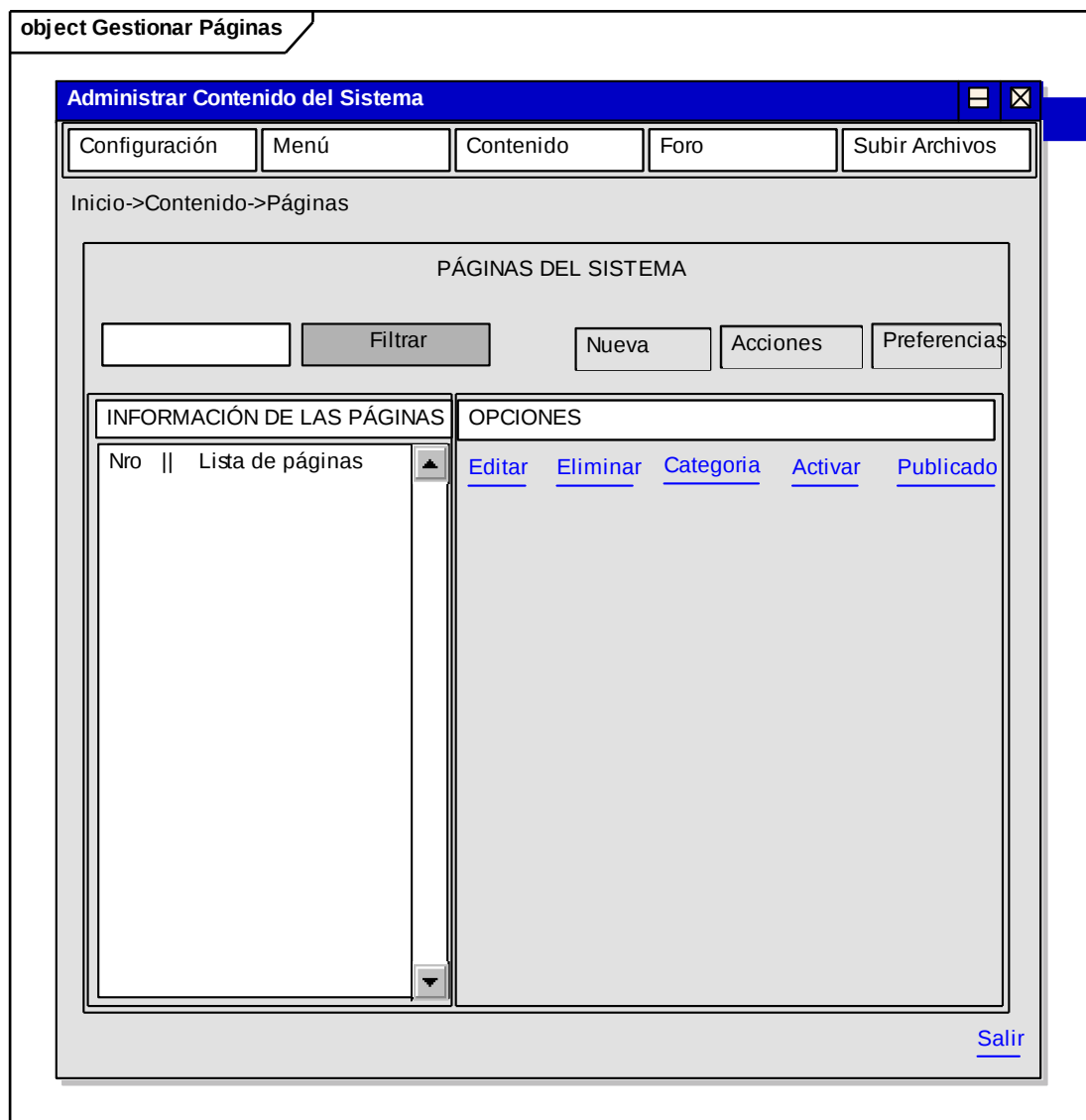


Figura 48: Pantalla Gestionar Páginas, aquí se Muestra el Bosquejo de la Gestión de Módulos.

object Editar Página

Administrar Contenido del Sistema

ConfiguraciónMenúContenidoForoSubir Archivos

Inicio->Contenido->Páginas->Editar Página

Título

Categoría

Estado

Tipo

Contenido:

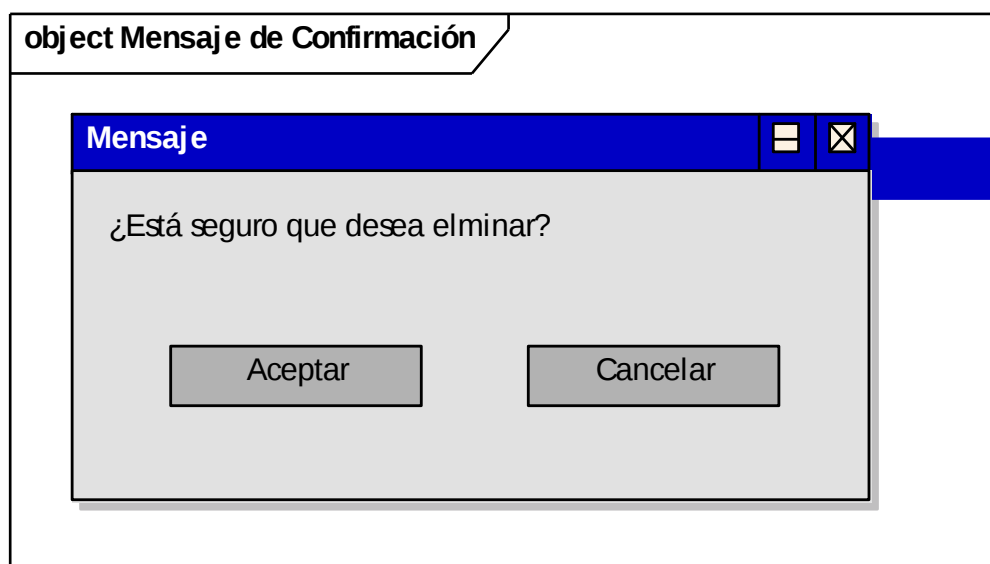
EDITOR DE TEXTO Y MULTIMEDIA

Guardar

Cancelar

Salir

Figura 49: Pantalla Editar Páginas, aquí se Muestra el Bosquejo de la Edición de Información.



**Figura 50: Pantalla de Confirmación de Eliminación, aquí se Muestra
el Bosquejo de los Mensajes Mostrados al Usuario**

7.1. CONCLUSIONES

Al concluir el presente trabajo dirigido se ha llegado a las siguientes conclusiones:

- El estudio realizado en la etapa inicial (ingeniería de requerimientos) posibilitó ordenar y clasificar toda la información obtenida para proceder a realizar la lista de requerimientos respectivos y encontrar los actores del sistema y el modelo de dominio del mismo.
- El análisis y diseño realizado demuestran que el presente trabajo es un sistema de administración de contenidos, por lo que se estudiaron CMS de licencia libre como Joomla y Wordpress para diseñar un sistema genérico pero que a la vez se ajuste a los requerimientos del hospital.
- Tomando la metodología del proceso unificado se logró diseñar el sistema pasando por todas las etapas de desarrollo (diseño) de acuerdo a las especificaciones requeridas.
- El lenguaje de Modelado UML empleado en la documentación del sistema se complementa adecuadamente con la metodología RUP, permitiendo una mejor comprensión del sistema y plasmando por medio de los diferentes diagramas las principales características del sistema.

7.2. RECOMENDACIONES

El presente trabajo dirigido está enfocado al análisis de requerimiento, análisis y diseño, por lo que se recomienda:

- Realizar los demás flujos de trabajo (implementación, pruebas e implantación) del desarrollo del sistema para contar con un sistema web completo que satisfaga las necesidades del Hospital Regional San Juan de Dios.
- Incentivar para que los demás centros de salud provean de servicios de información y control vía Web, para mejorar la atención de los ciudadanos.

- Integrar los sistemas de control existentes en el Hospital, al sistema web, de tal manera que se pueda proveer de una plataforma completa de información, control y comunicación entre los pacientes, el hospital e instituciones que así lo requieran.

8.1. GLOSARIO DE TÉRMINOS DEL NEGOCIO

Beneficiario

Personas que sean beneficiarios de alguno de los programas de salud existentes en nuestro departamento.

Paciente

En la medicina y en general en las ciencias de la salud, el paciente es alguien que sufre dolor o malestar (muchas enfermedades causan molestias diversas, y un gran número de pacientes también sufren dolor). En términos sociológicos y administrativos, paciente es el sujeto que recibe los servicios de un médico u otro profesional de la salud, sometiéndose a un examen, a un tratamiento o a una intervención.

Prestación

Acciones de salud, tecnología o dispositivos médicos, tales como consultas médicas, exámenes y procedimientos; medicamentos; artículos farmacéuticos y de laboratorio; material quirúrgico, instrumental y demás elementos o insumos que se requieran para el diagnóstico de un problema de salud y su tratamiento o seguimiento.

Programa

Un programa de salud es un conjunto de acciones implementadas por un gobierno con el objetivo de mejorar las condiciones de salud de la población. En nuestro medio contamos con los programas:

Seguro Público de Salud (SPS)

Seguro Universal de Salud Autónomo de Tarija (SUSAT)

Seguro de Salud para el Adulto Mayor (SSPAM)

Bebé Churo y Sanito (Alimento Complementario)

Abuelo Querido Churo y Sanito (Complemento Nutricional)

Mamá Chura (Suplemento Nutricional)

SEDES

Servicio Departamental de Salud

Unidad

Cada una de las secciones en las que está dividido el hospital, por ejemplo: unidad de ginecología, unidad de terapia intensiva, unidad de pediatría y similares.