
Universidad de las Ciencias Informáticas

FACULTAD 10



TÍTULO:

**ANÁLISIS Y DISEÑO
DEL SISTEMA INTRANET 2**
.....

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

AUTORES:

Lianelys Braña Verdeal

Yeinier Chacón Romero

TUTORES:

Lic. Dunia Suárez Ferreiro

Ing. Yulier Molina Ramírez

**CIUDAD DE LA HABANA, JULIO 2007
«AÑO 49 DE LA REVOLUCIÓN»**

DECLARACIÓN DE AUTORÍA

.

Declaramos ser los autores de la presente Tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del 2007.

Nombre Autor

Nombre Tutor

Firma del Autor

Firma del Tutor

..... DATOS DE CONTACTO

Yulier Molina Ramírez (ymolina@uci.cu): Graduado de Ingeniero en Ciencias Informáticas, primera graduación de la UCI. Posee categoría docente de Instructor Recien Graduado. Actualmente imparte clases de P1 y está vinculado a un proyecto productivo.

Dunia Suárez Ferreiro (dsuarez@uci.cu): Graduada de Ingeniero en Ciencias de computación en la CUJAE. Posee categoría docente de Instructor Recien Graduado. Actualmente imparte clases de P4 y es líder del proyecto productivo Intranet 2.

«Hay una fuerza motriz
más poderosa que el vapor,
la electricidad
y la energía atómica:
la voluntad»

Albert Einstein

AGRADECIMIENTOS

.....

A nuestro Comandante en Jefe	Gracias a mis padres,
Fidel Castro Rúz,	por estar siempre presentes
por haber hecho realidad el sueño	y ser los mejores del mundo.
de ser un profesional.	A mi Tata,
A Sara Revé por su maravilloso trabajo.	por apoyarme en todo momento.
A Dunia.	A mi abuela
A mi tutor Yulier Molina	por su crianza y amor.
por su preocupación.	A mis abuelos,
A Susel Vázquez	aunque ya no estén.
por su ayuda.	A mi bebe,
A mis compañeros de aula,	por su interminable amor y paciencia.
Leynier y Karel.	A todos los que confiaron en mí
A la decana Sayda por luchar conmigo	y dieron su granito de arena.
en los primeros años y por su ayuda.	Yanedi, Susel, Sara, David.....
A la UCI	A mis amigas,
por haberme dado estos maravillosos	Zu y Duny.
cinco años.	Gracias siempre al gran soñador,
	Fidel Castro.

Yeinier Chacón Romero

Lianelys Braña Verdeal

DEDICATORIA

.....

A mis padres Ana María y Carlos que tanto
esfuerzo y sacrificio han hecho para que yo
realice este sueño.

A mi hermana Yoly que tantos consejos
que me ha dado.

A mis abuelos Diego y Noelia por tanto amor.
A los que ya no están a mi lado, sepan que
seguiré formándome como profesional.

Silvio que le habría gustado verme converti-
do en ingeniero, mi abuelo Beto que sé que
me está mirando y siempre está conmigo.
Mi tía Gladis por trasmitirme su alegría y
sus ganas de vivir.

A mi tía Nelsa y mis primos Luis y Yunior
por los buenos fines de semana que me he
pasado en su casa y por tanto cariño.

A mi primo Ismael por siempre preocuparse.
Al resto de mi familia por creer en mí.

A mi novia Yulie por su paciencia y cariño.

A mis vecinos por su confianza.

A mis colegas y amigos Sergio, Luis Enri-
que, Yurisander, Felipe, Yordi, Yoalber, el
chiqui, Leo, Zambrano, Reinier, Edisbel,
Alex, Leynier, Yoel, Karel, Osmell, Carlos,
Rafael, el pichi, Alexis, Amado, Norbeys,
Maikel por su amistad; sepan que siempre
los llevo en el corazón.

A mi mamá y papá por darme la vida.

A mi hermana por estar siempre a mí lado. A
mi abuela por su confianza.

A mi bebe por aguantarme tantos años,
y por darme su amor.

A mi familia completa por ser el sustento de
mi felicidad.

A mi coquito pequeño por darme tantas
alegrías. A mis abuelos que aunque ya se
han ido, siempre los tengo presentes en
cada momento.

A mis amigos del pre, que tanto los extra-
ños, mi macha querida, machenca, mis
niños....., todos.

A mis profesores de tantos años que forja-
ron mi pensamiento y persona.

A la UCI por darme la oportunidad.

A mis amigas Dunia y Zuyen por ayudarme
y cargar conmigo todos estos años de larga
espera.

A todas las personas que me han ayudado
y confiado en mí.

Y principalmente al Quijote, al de la sonrisa
amplia e ideas de futuro, a nuestro
Comandante Fidel.

Chacón Lia

RESUMEN

.....

Las Intranets actualmente agilizan todos los procesos dentro de una empresa o institución, esta hace posible una comunicación más eficiente con todos los miembros y que toda información sea conocida por los usuarios que la visiten. A partir de estas afirmaciones se decide crear en la UCI una Intranet colaborativa orientada a la gestión, organización e información de contenido. La Intranet actual presenta disímiles problemas en su funcionamiento y esta es la principal causa de la toma de decisión de crear una nueva Intranet que cumpla con las expectativas actuales de la universidad y que responda a sus procesos y servicios.

Por tanto el objetivo concreto de este trabajo consiste en realizar el análisis y diseño de sistema de la Intranet 2, planteando todos los artefactos desarrollados en los diferentes flujos de trabajos que se encuentran aquí reflejados.

Este documento recoge los resultados de todo el trabajo realizado. Se estudian las características esenciales de la Intranet y se realiza un análisis de las tendencias y tecnologías existentes seleccionándose las más apropiadas para desarrollar la aplicación. Además, se muestran los resultados del análisis y diseño de la propuesta del sistema, y al final se concluyen y se recomiendan algunos aspectos para el mejoramiento futuro del mismo.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA.....	4
1.1 Visión general de la Intranet.....	4
1.1.1 Conceptos.	4
1.1.2 Objetivos.....	4
1.1.3 Como funciona una Intranet.....	5
1.1.4 Ventajas y desventajas de la Intranet.....	7
1.2 ¿Por qué migrar al Software Libre?.....	9
1.3 Requerimientos.....	10
1.3.1 Características de los requerimientos.....	10
1.3.2 Requerimientos funcionales y no funcionales.....	11
1.3.2.1 Categorías de los requerimientos no funcionales.....	12
1.3.3 Captura de requerimientos.....	14
1.3.4 Técnicas de validación de requerimientos.....	15
1.4 Sistemas Gestores de contenido (CMS).....	15
1.4.1 Categorías de los CMS	15
1.4.2 Características del CMS Plone.....	18
1.4.2.1 CMS Plone.....	18
1.4.3 Fundamentación de la selección del CMS.....	19
1.5 Sistemas gestores de Base Datos (SGBD).....	20
1.5.1 Características de los SGBD.....	20
1.5.1.1 MySQL.....	20
1.5.1.2 Oracle (Relational Data Base Management System).....	21
1.5.1.3 Microsoft SQL Server 2000.....	21
1.5.1.4 PostgreSQL.....	21
1.5.2 Fundamentación de la selección del SGBD.....	21
1.6 Metodologías de desarrollo del software.....	22

1.6.1 Proceso de desarrollo de software.....	22
1.6.2 Metodologías.....	22
1.6.2.1 El Proceso Unificado de Desarrollo (RUP).....	22
1.6.2.2 Extreme Programing (XP).....	25
1.6.2.3 Microsoft Solution Framework (MSF).....	26
1.6.2.4 Métrica.....	27
1.6.3 Fundamentación de la selección de la metodología de desarrollo de software.....	27
1.6.4 Herramientas de modelo visual.....	28
1.7 Análisis y Diseño de sistemas.....	28
1.7.1 Conceptos.....	28
1.7.1.1 Análisis de Sistemas.....	28
1.7.1.2 Diseño de Sistemas.....	29
1.7.2 Tendencia del Diseño.....	30
1.7.3 Patrones de Diseño.....	32
1.7.3.1 Reseña Histórica.....	32
1.7.4 Patrones de diseño web.....	33
1.7.4.1 Tipos de patrones de diseño web.....	33
1.7.4.2 Patrón de diseño MVC (Model View Controller) o modelo vista controlador.....	35
1.7.4.2.1 Ventaja del MVC.....	36
1.7.5 Patrones de Diseño de Software.....	36
1.7.5.1 Patrones de diseño de software. (GRAPS).....	37
1.7.5.2 Patrones de diseño de software GOF.....	41
1.7.5.2.1 Tipos de Patrones de diseño de software.GOF.....	41
 CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA.....	 47
2.1 Objeto de automatización.....	47
2.2 Información que se manejará en el Portal de la Intranet.....	47
2.3 Propuesta de sistema.....	48
2.4 Modelo del Dominio.....	49

2.4.1 Diagrama Conceptual Modelo del dominio.....	50
2.5 Especificación de los requisitos de software.....	51
2.5.1 Requerimientos Funcionales.....	51
2.5.2 Requerimientos no funcionales.....	57
2.6 Definición de los casos de uso.....	59
2.6.1 Definición de los actores.....	59
2.6.2 Listado de casos de uso del sistema.....	60
2.6.3 Diagrama de Casos de Uso del Sistema.....	63
2.6.4 Casos de uso expandidos.....	64
 CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA.....	 70
3.1 Modelo de análisis.....	70
3.1.1 Definición del modelo de análisis.....	70
3.1.1.1 Diagramas de clases de análisis.....	70
3.2 Patrones de diseño.....	74
3.3 Modelo de Diseño.....	74
3.3.1 Definición del modelo de diseño.....	74
3.3.2 Diagramas de interacción.....	74
3.3.2.1 Diagrama de Secuencia.....	75
3.4 Diagrama de clases de diseño.....	75
3.4.1 Diagrama de clases Web.....	76
3.5 Análisis y diseño en la calidad del software.....	76
3.6 Principios de diseño que se usaran.....	77
3.6.1 Estándares de la interfaz de la aplicación.....	77
 CONCLUSIONES GENERALES.....	 79
 RECOMENDACIONES.....	 80

REFERENCIAS BIBLIOGRÁFICAS.....	81
BIBLIOGRAFÍA.....	84
ANEXOS.....	86
ANEXO 1.....	86
ANEXO 2. MODELO DE DOMINIO.....	88
ANEXO 3. DIAGRAMAS DE SECUENCIA.....	89
ANEXO 4. DIAGRAMAS DE CLASES WEB.....	95
ANEXO 5. INTERFAZ DE LOS CU.....	105
GLOSARIO.....	110



ÍNDICE DE FIGURAS

Fig. 1 Esquema general de un CMS.....	16
Fig. 2 Representación del Proceso de desarrollo de software.....	24
Fig. 3 Metodología Extreme Programing.....	25
Fig. 4 Metodología MSF.....	26
Fig. 5 Estructura de un modelo convencional.....	35
Fig. 6 Funcionamiento del Controller.....	36
Fig. 7 Estructura de clases.....	38
Fig. 8 Estructura de la clase Venta con un nuevo método.....	38
Fig. 9 Estructura de clases.....	39
Fig.10 Estructura general del patrón Singleton.....	41
Fig.11 Estructura general del patrón de diseño Prototype.....	42
Fig.12 Estructura general del patrón de diseño Adapter.....	43
Fig.13 Estructura general del patrón de diseño Bridge.....	43
Fig.14 Estructura del patrón de diseño Fascade.....	44
Fig.15 Estructura del patrón de diseño Chain of Responsibility.....	45
Fig.16 Estructura del patrón de diseño Iterator.....	45
Fig.17 Diagrama de CUS.....	63
Fig.18 Diagrama de clases de análisis de CUS adicionar contenido.....	71
Fig.19 Diagrama de clases de del CUS análisis actualizar contenido.....	72
Fig.20 Diagrama de clases de análisis del CUS aprobar contenido.....	73
Fig.21 Diagrama de clases de análisis del CUS autenticar usuario.....	74
Fig.22 Estructura del Diseño Web propuesto para la Intranet2.....	78
Fig.23 Estructura del modelo vista controlador (MVC).....	86
Fig.24 Estructura general del patrón de diseño Abstract Factory.....	86
Fig.25 Estructura del patrón de diseño Observer.....	87
Fig.26 Modelo de Dominio.....	88
Fig.27 Diagrama de Secuencia del caso de uso «Autenticar Usuario».....	89

Fig.28	Diagrama de secuencia del caso de uso «Adicionar Contenido». Sección «Adicionar Noticia».....	90
Fig.29	Diagrama de secuencia del caso de uso «Adicionar Contenido». Sección «Adicionar Aviso».....	91
Fig.30	Diagrama de secuencia del caso de uso «Adicionar Contenido». Sección «Adicionar Efeméride».....	91
Fig.31	Diagrama de secuencia del caso de uso «Actualizar Contenido». Sección «Actualizar Noticia».....	92
Fig.32	Diagrama de secuencia del caso de uso «Actualizar Contenido». Sección «Actualizar Aviso».....	92
Fig.33	Diagrama de secuencia del caso de uso «Actualizar Contenido». Sección «Actualizar Efeméride».....	93
Fig.34	Diagrama de secuencia del caso de uso «Aprobar Contenido». Sección «Aprobar Noticia».....	93
Fig.35	Diagrama de secuencia del caso de uso «Aprobar Contenido». Sección «Aprobar Aviso».....	94
Fig.36	Diagrama de secuencia del caso de uso «Aprobar Contenido». Sección «Aprobar Efeméride».....	94
Fig.37	Diagrama de clases de diseño Web CU «Adicionar Contenido». Sección «Adicionar Noticia».....	95
Fig.38	Diagrama de clases de diseño Web CU «Adicionar Contenido». Sección «Adicionar Aviso».....	96
Fig.39	Diagrama de clases de diseño Web CU «Adicionar Contenido». Sección «Adicionar Efeméride».....	97
Fig.40	Diagrama de clases de diseño Web CU «Actualizar Contenido». Sección «Adicionar Noticia».....	98
Fig.41	Diagrama de clases de diseño Web CU «Actualizar Contenido». Sección «Actualizar Aviso».....	99
Fig.42	Diagrama de clases de diseño Web CU «Actualizar Contenido». Sección «Actualizar Efeméride».....	100
Fig.43	Diagrama de clases de diseño Web CU «Aprobar Contenido». Sección «Aprobar Noticia».....	101
Fig.44	Diagrama de clases de diseño Web CU «Aprobar Contenido». Sección «Aprobar Aviso».....	102
Fig.45	Diagrama de clases de diseño Web CU «Aprobar Contenido». Sección «Aprobar Efeméride».....	103

Fig.46 Diagrama de clases de diseño Web CU «Autenticar Usuario».....	104
Fig.47 Interfaz completa.....	105
Fig.48 Adicionar noticia.....	105
Fig.49 Actualizar noticia.....	106
Fig.50 Adicionar aviso.....	106
Fig.51 Actualizar aviso.....	107
Fig.52 Adicionar efeméride.....	107
Fig.53 Solicitud de publicación.....	108
Fig.54 Aprobar solicitud.....	108
Fig.55 Aprobar contenido.....	109

ÍNDICE DE TABLAS

Tabla 1 Actores del Sistema.....	60
Tabla 2 CU Autenticar usuario (Resumen).....	60
Tabla 3 CU Adicional contenido (Resumen).....	60
Tabla 4 CU Actualizar contenido (Resumen).....	61
Tabla 5 CU Aprobar contenido (Resumen).....	61
Tabla 6 CU Enviar solicitud de publicación (Resumen).....	61
Tabla 7 CU Aprobar solicitud de publicación (Resumen).....	61
Tabla 8 CU Publicar sitio Web (Resumen).....	62
Tabla 9 CU Publicar página informativa (Resumen).....	62
Tabla 10 CU Gestionar usuario (Resumen).....	62
Tabla 11 CU Buscar información del contenido del portal (Resumen).....	62
Tabla 12 CU Mostrar estadística del portal (Resumen).....	63
Tabla 13 CU Autenticar usuario.....	64
Tabla 14 CU Adicionar Contenido.....	64
Tabla 15 CU Actualizar Contenido.....	65
Tabla 16 CU Aprobar Contenido.....	67

INTRODUCCIÓN.

En la actualidad, las tecnologías de la informática están presentes prácticamente en todas las funciones laborales. La disponibilidad de nuevas herramientas y la necesidad de una comunicación cada vez más ágil en las empresas, dieron nacimiento a lo que hoy llamamos INTRANET una red interna de comunicación e información que emplea la misma tecnología que se utiliza en Internet, siendo accesible únicamente para los usuarios miembros de la Organización. [1]

Una intranet es un sitio Web interno y privado de una empresa al que sus usuarios pueden acceder mediante un navegador y permite múltiples funcionalidades encaminadas a mejorar la eficiencia de la organización en la que se implementa. Sus servicios son muy similares a los de Internet, pero con dos ventajas fundamentales: mayor seguridad, al tratarse de un entorno cerrado y de acceso restringido, y mayor velocidad, ya que no está supeditada al funcionamiento de la red Internet, permitiéndole a la empresa:

- Publicar, compartir información y documentación siempre actualizada: noticias de interés, normativas internas, protocolos de actuación, manuales de referencia, circulares...
- Organizar sistemas de colaboración y flujos de trabajo: elaboración de proyectos y presupuestos, agendas de trabajo, gestión de documentos de producción...
- Establecer canales de comunicación interna: boletines de información, mensajería, foros, sistemas de comunicación escrita en tiempo real...
- Personalizar los contenidos a los que accede cada usuario en función de su perfil o puesto.

Entre otras funcionalidades en dependencia del lugar donde esta se encuentre implementada.[2]

La Universidad de las Ciencias Informáticas surgida al calor de la Batalla de Ideas en el 2002, toma en cuenta los beneficios que ofrece la implementación en el centro de una Intranet, pero con el aumento de la matrícula universitaria cada año, las prestaciones de ésta iban en ascenso y al mismo tiempo surgieron diferentes problemas en su funcionamiento como son:

- Servidor generalmente congestionado.
- Problemas de integridad de la Información y de la autogestión de la Intranet.
- Ausencia de personal con conocimientos del código fuente de la Intranet actual.
- Carencia de documentación de la misma.
- Decadencia natural de un software que no se mantiene.
- La plataforma informática actual no brinda las posibilidades para crear un correcto flujo de trabajo en equipo, definiendo responsabilidades, niveles y flujos de trabajos.

- Falta una estrategia general para el desarrollo de la arquitectura de la información, en la que se pauten una navegación estándar donde informaciones similares tengan el mismo nivel.

Sin tener en cuenta que la información no presenta un correcto flujo y que no todas las áreas de la universidad ven reflejados sus intereses en el portal, así como la carencia de la adecuada publicación de contenidos interesantes para todos los usuarios.

Por las dificultades anteriormente descritas surge la necesidad de desarrollar una nueva Intranet que funcione según las características actuales de la UCI, y que tuviese un diseño totalmente diferente, no solo visual, sino también en cuanto a la arquitectura de información. Al mismo tiempo de regirse por las normas de una Intranet corporativa, orientada a servicios y centrada generalmente a la gestión de la información endógena y exógena en pos de la creación de nuevos acervos y conocimientos que influyen en la producción de productos con alto valor agregado.

Con esto nos referimos a que no solo debe contar con tecnologías de punta sino que éstas deben ser conjugadas de manera eficiente para lograr un producto de alta calidad.

Motivado por esta necesidad surge el proyecto Intranet 2, en el cual se trabaja en función de solucionar estos problemas. Inicialmente se definieron dos etapas del proyecto:

1. Migración de la Intranet actual con la misma imagen que presenta hacia un CMS libre de fácil manipulación, que brinde facilidades para la administración y la publicación de noticias con mayor riqueza.
2. Desarrollo de una nueva Intranet (Intranet 2) con las características ya mencionadas.

Según la **Situación Problemática** planteada anteriormente, el **problema científico** a resolver queda formulado de la siguiente forma: ¿Cómo realizar el análisis y diseño de la Intranet de la Universidad de las Ciencias Informáticas para resolver las limitaciones de la intranet anterior?

El **objeto de estudio** lo constituye el análisis y diseño de aplicaciones web orientadas a los servicios de publicación de contenido.

De aquí se deriva que el **campo de acción** comprende el Análisis y Diseño de la Intranet universitaria a partir de la descripción de los servicios de publicación de servicios descritos en la Arquitectura de Información propuesta. Planteando como **objetivo general** realizar el Análisis y Diseño de la nueva Intranet para apoyar los procesos de gestión de la información y del trabajo colaborativo de los miembros de la comunidad universitaria.

De lo anterior se derivan las siguientes **preguntas científicas**:

- ¿Qué características y procesos enmarca una Intranet y qué tecnologías y metodologías de desarrollo del software utilizar para realizar el Análisis y Diseño?
- ¿Cómo identificar las características del sistema Intranet 2 a partir de los procesos y servicios que esta comprende?
- ¿Cómo realizar el Análisis y Diseño a partir de la descripción de los servicios propuestos en la Arquitectura de Información?

Con vistas al cumplimiento de las preguntas científicas se propone la realización de las siguientes tareas agrupadas en dos por cada pregunta:

- Realizar estudio de las deficiencias de la Intranet actual.
- Estudiar y fundamentar la elección de la metodología del software usada para realizar el Análisis y Diseño.
- Identificar servicios y procesos a incluir en la Intranet 2.
- Formalizar propuesta del sistema.
- Definir y realizar los diferentes artefactos de los flujos de Análisis y Diseño.
- Aplicar patrones de diseño.

El presente documento se estructura en tres capítulos:

- Capítulo I denominado «Fundamentación Teórica» donde se incluyen todos los aspectos teóricos que soportan este proyecto.
- Capítulo II denominado «Características del sistema» donde se plantea objeto de automatización, información que se maneja, propuesta del sistema, modelo del negocio o dominio, se determinan además los requerimientos funcionales y no funcionales del sistema, agrupándolos en casos de uso.
- Capítulo III denominado «Análisis y diseño del sistema» donde se determinan las clases que se utilizarán en la implementación del sistema y la relación entre ellas. La realización de los diagramas de interacción por cada caso de uso. Se definen los principios de diseño de interfaz y codificación.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

En este capítulo se incluirán todos los aspectos teóricos que soportan la investigación. Se ampliarán los conocimientos en cuanto a la intranet, su funcionamiento y ventajas que estas ofrecen. También se dará repuesta a la interrogante ¿por qué es necesaria la migración al software libre? Así como se expondrán las diferentes metodologías de desarrollo del software y se fundamentará el porqué de la elegida para desarrollar los artefactos de los diferentes flujos de trabajos. Igualmente se encontrarán algunos aspectos sobre el análisis y el diseño, así como otros temas que se tratarán en el desarrollo del trabajo.

1.1 Visión general de la Intranet.

Con el enorme crecimiento de Internet, un gran número de personas en las empresas lo usan para comunicarse con el mundo exterior, para compilar información y hacer negocios. Al internauta no le lleva mucho tiempo reconocer que los componentes que funcionan bien en Internet serían del mismo modo valioso en el interior de sus empresas y esa es la razón por la que las Intranets se están haciendo tan populares.

Crear una Intranet donde todas las informaciones y recursos se puedan usar sin dificultades tiene muchos beneficios, permitiendo a los usuarios trabajar juntos de un modo más sencillo y eficaz.

1.1.1 Conceptos.

Entre los diferentes conceptos que se plantean se tiene que una Intranet es:

- Una red privada que la tecnología Internet usó como arquitectura elemental.[3]
- Es la aplicación de la tecnología Internet a una red privada, un grupo de redes corporativas internas, cuyos usuarios son los miembros de la organización.[4]
- Conjuntos de redes internas con uso exclusivo para sus trabajadores.
- Red privada en la que los usuarios dentro de una empresa que trabajan con ella pueden acceder a Internet, pero aquellos en Internet no pueden entrar en la Intranet de dicha empresa.[5]

Existen otros conceptos, pero todos basan su esencia en que es un conjunto de redes pertenecientes a una empresa que se comunican entre ellas mediante TCP/IP donde solo tendrán accesos a ella los trabajadores de esta.

1.1.2 Objetivos.

Las Intranets se utilizan para permitir a las empresas llevar a cabo transacciones, hacer pedidos, enviar facturas, y efectuar pagos. Para proteger la información corporativa delicada, y para asegurar que los piratas no perjudiquen a los sistemas informáticos y a los datos. También brinda gran cantidad de beneficios, entre ellos: optimizar y compartir los recursos de la empresa, incrementar su productividad, minimizar el uso del papel, aumentar la presta-

ción de servicios y la calidad de los mismos a los clientes, disponibilidad de información oportuna que apoye la búsqueda de soluciones y la toma de decisiones.

El objetivo central de la Intranet en una organización debe ser la creación de un sistema que apoye el desempeño de los trabajadores, de manera que les ayude a crear e integrar valor en sus procesos. Para cumplir con este objetivo, es necesario tener en cuenta los requerimientos de información y conocimiento del tipo institucional, del trabajador, de los productos y servicios, de los documentos, del contenido; pero sobretodo de los procesos medulares del negocio. Esto último se refiere a toda información que está vinculada con las actividades y tareas que son propias de la organización.

Con estas características y con la posibilidad de personalización de las páginas para cada empleado, que entra con su nombre de usuario y contraseña del dominio y con la facilidad de encontrar información en función de su departamento, perfil profesional, y otros detalles personales, se realizará la nueva Intranet Universitaria.

1.1.3 Como funciona una Intranet.

El centro de una Intranet es la World Wide Web (WWW). En muchos casos gran parte de la razón por la que se creó una Intranet en primer lugar es que la Web facilita la publicación de la información y formularios usando el Lenguaje de Marcado de Hipertexto (HTML). La Web permite la creación de páginas iniciales multimedia, que están compuesta de texto, gráficos, y contenidos multimedia como sonido y vídeo. Los enlaces de hipertexto te permiten saltar desde un lugar en la Web a otro, lo que significa que puedes saltar a lugares dentro de una Intranet o fuera en Internet desde una página inicial.

Las Intranets están basadas en la arquitectura cliente / servidor. El software cliente (un navegador para Web), se ejecuta en una computadora local y el software servidor en una Intranet anfitriona. El software cliente y el software servidor no necesitan ejecutarse en el mismo sistema operativo. [6]

Los procesos realizados en una empresa son fuertes candidatos a formar parte de las aplicaciones de la Intranet, de esta forma, los diferentes agentes participantes del proceso, cuentan con mecanismos claros para la realización de cada actividad involucrada, y se tiene también un registro de eventos. Esto permite agilizar y controlar la ejecución de tareas, mientras se hace posible la identificación de cambios y requerimientos en los flujos de trabajo operativos.

Los procesos que se pueden ser considerables para la Intranet son:

- Los procesos de negocio que involucran la producción, distribución y actualización de información tradicionalmente en papel. Ejemplos de ello lo constituyen: los directorios de empleados, clientes y estudiantes; boletines, periódicos y servicios, noticias, manuales de uso y especificaciones, listas de precios, material de apoyo; reportes y listados de consultas frecuentes o eventuales; manuales de procedimientos, funciones y políticas corporativas; formas, formatos y plantillas; y solicitudes de procesos.
- Procesos que requieren consolidar información de diversas fuentes para la toma de decisiones.

- Algunos de los procesos de planteados dentro de la organización requieren de un alto grado de comunicación y colaboración entre sus empleados, tal como diseño de nuevos productos, rediseño de procesos y proyectos de ingeniería. Este tipo de procesos son de gran beneficio si están incorporados en la Intranet, en especial si los individuos involucrados se encuentran geográficamente dispersos.
- Otros procesos como búsquedas de información y aplicaciones cliente/servidor de uso frecuente y generalizado, pueden estar en la Intranet.

Algunos ejemplos de aplicaciones a desarrollar para la Intranet son:

- Actas y acuerdos.
- Reglamentos Internos.
- Calendarios laborales como fechas de pago.
- Consultas al presupuesto.
- Manuales de instalación de software institucional.
- Manuales de procedimientos de las diferentes dependencias.
- Información sobre Seminarios, Conferencias, Congresos.
- Soporte Técnico.
- Foros de discusión.
- Cuestionarios.
- Formularios Electrónicos.
- Boletines y Noticias.
- Solicitudes Internas de Servicios.
- Información sobre cursos y capacitación a funcionarios.
- Cursos Virtuales.
- Evaluaciones de Personal.
- Secciones de Búsqueda.
- Consultas a bases de datos.
- Directorios telefónicos.

Los procesos a incluir en la Intranet, Trulin los tipifica en cinco tipos de servicios:

1. Servicios de colaboración: Incluye la publicación de contenido, correo electrónico, noticias y espacios de discusión.
2. Servicios transaccionales: Son las aplicaciones de base de datos, formas y reportes. Por ejemplo inventarios de productos, administración de ventas.

3. Servicios de proceso: Es el flujo de trabajo, gerencia de proyecto, agendas compartidas.
4. Servicios de directorio: Búsquedas, administración de accesos y recursos, certificación.
5. Servicios de utilidades: Administración y desarrollo de trabajos en red, respaldo, seguridad y desempeño.

La demanda por este tipo de aplicaciones es alta, existen gran cantidad de procesos factibles ha desarrollar para una Intranet, en general todo lo relacionado con los servicios brindados a las diferentes dependencias administrativas son potenciales aplicaciones.

1.1.4 Ventajas y desventajas de la Intranet.[7]

La Intranet se considera como una herramienta de trabajo colaborativo, que aporta múltiples beneficios, dentro de ellos, es posible identificar tanto ventajas estratégicas como tácticas.

Entre sus ventajas encontramos:

- Nos permite trabajar en grupo, en proyectos, compartir información, llevar a cabo conferencias visuales y establecer procedimientos seguros para el trabajo de producción.
- Es una red privada, donde los usuarios dentro de la empresa pueden acceder a Internet, pero aquellos en Internet no pueden entrar en la Intranet de dicha empresa.
- El software que se utilizan en las Intranets es estándar: software de Internet como él Netscape, Navigator, los Navegadores Explorer para Web de Microsoft y el Mozilla.
- Para proteger la información corporativa delicada, presenta barreras de seguridad llamadas firewall (esta tecnología usa una combinación de enrutadores, que permite a los usuarios utilizar los recursos de Internet, para evitar que los intrusos se introduzcan en ella.
- Se puede subdividir en varios niveles al momento de sobrepasar su tamaño y al ser difícil de manejar.
- Ayuda a la toma de decisiones al disponer de la información en tiempo real.
- A través de una sección de preguntas y respuestas eliminará las consultas reiteradas, por los mismos temas, a las mismas personas o departamentos.
- Aumenta el rendimiento en la gestión de grandes volúmenes de información, automatiza los procesos administrativos o de negocio (workflow automation).
- Quedan atrás las tareas de imprimir, distribuir y archivar los documentos.
- Posibilita una perfecta transmisión de la cultura empresarial a todos los empleados y colaboradores.
- Ahorro: Al proveer información instantánea y segura en formato electrónico, se elimina el tiempo y costo asociado a la publicación, duplicación y distribución asociados a la documentación en papel.

- Brinda a los funcionarios información de calidad, veraz y actualizada, beneficiando de esta forma, la elaboración de procesos y toma de decisiones, pues la información está en línea.
- Intranet ofrece un mayor nivel de eficiencia, al eliminar todo el proceso manual de las solicitudes, y permitir un flujo de la información a través del correo electrónico.
- Mejora la comunicación al utilizar el sistema de mensajería electrónico tanto en comunicaciones internas y externas, en tiempo real y diferido, y completamente integradas.
- Optimización en el uso de la red institucional y recursos tecnológicos instalados.

A pesar de las numerosas ventajas que trae consigo la implementación de la Intranet, se pueden presentar inconvenientes y problemas durante y después de su implementación, entre los más comunes encontramos:

- Administración de contenidos.
- Falta de recursos económicos.
- Sobrecarga de contenido.
- Seguridad.
- Control insuficiente.
- Navegación confusa.
- Políticas internas.

Al analizar cada problema, podemos indicar la Administración del Contenido como el principal. De ahí la importancia de la responsabilidad y compromiso de la actualización de la información.

Con relación a la falta de recursos económicos, aunque la Intranet utiliza la misma plataforma que Internet y los recursos ya instalados en la organización, se requiere de una adecuada planificación financiera, especialmente para el desarrollo de las aplicaciones, compra de software, capacitación de personal, reforzar la seguridad en la red y mantenimiento.

Otro problema es la sobrecarga de información, el cual se presenta cuando hay exceso de ella en la Intranet. Se presentan dificultades en la búsqueda de información útil, dada la existencia de información basura difícil de clasificar.

En cuanto a la seguridad, se requiere de una estrategia bien definida y no improvisada, para contar con mecanismos que eviten la intromisión de usuarios ajenos a la organización o usuarios internos no autorizados. El control insuficiente y falta de políticas internas, provocan problemas en los niveles de seguridad, cuando cualquier usuario puede acceder a todas las aplicaciones existentes. A la vez, la falta de políticas claras, acarrea en un desarrollo más lento, el personal involucrado lo ve como una opción y no como una política a nivel organizacional, por lo tanto las pruebas y puesta en ejecución de las aplicaciones es un proceso lento, el cual produce gastos innecesarios en tiempo y recursos. Por lo tanto es fundamental contar con políticas internas en el uso de la Intranet para garantizar el adecuado control y uso de la información y recursos disponibles.

El inconveniente de una navegación confusa, se produce cuando no se cuenta con un buen diseño de las páginas en términos de usabilidad y estructuración de la Intranet, dando como resultado que la información no sea encontrada en forma rápida y oportuna. La definición de un buen diseño es un punto clave, si consideramos que el impacto de malos diseños se traslada directamente a los empleados de la organización, porque cualquier problema de usabilidad, podría inmediatamente reducir el nivel de su productividad.

1.2 ¿Por qué migrar al Software Libre?

Para la implantación de una Intranet hay dos opciones básicas, optar por tecnologías Microsoft o Linux. Las segundas tienen la gran ventaja de que reducen los costos; entre otros beneficios que se obtienen del trabajo en un entorno de software libre.

El término «software libre» es la traducción del inglés «free software»; en inglés el término «free» conlleva confusión, ya que significa tanto «libre» como «gratis». Por ello fue necesario aclarar que la palabra «libre» en inglés, no se refiere a términos de coste económico, sino al término libertad (del inglés freedom). Con este significado se ha utilizado el término desde los años 80, apareciendo la primera definición documentada en el año 1989 en el boletín GNU's Bulletin, vol. 1 no. 6. En él se enumeran cuatro «libertades» que definen el software libre:

- La libertad de usar el programa, con cualquier propósito.
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades.
- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino.
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

Estas libertades son derechos, no obligaciones. Toda persona es libre de no usar estas libertades, pero también podría elegir el uso de todas ellas.

Es importante conocer que el software libre no excluye el uso comercial ni la distribución comercial, ya que no es software gratuito. Por ejemplo un número creciente de empresas basa su negocio completamente o al menos parcialmente en el software libre. El software libre hace legal proveer ayuda y asistencia, pero no lo hace imprescindible. [8]

En este sentido el software libre, es a menudo, la mejor opción para la empresa. Ya no en términos económicos, debido a que por ejemplo la implementación, ayuda o asistencia para software libre generalmente es más barata, sino en términos de control y/o despersonalización.

Otra gran ventaja es que la mayoría de los programas de software libre tienen una política de mejora basada en contribuciones de personas no relacionadas directamente con la creación y desarrollo del programa. Cuando se crea un módulo o herramienta con código fuente público hace posible que otros usuarios lo utilicen o mejoren.

Actualmente el gran problema de la implementación de software libre en las empresas es el miedo a que estos programas dejen de mejorarse o desarrollarse. Pero esto es una falsa creencia, ya que la filosofía del software libre impide que nunca se de este punto, en cambio esta situación crítica puede darse con el software no libre.

Todas estas afirmaciones llegan a la conclusión de la necesidad del uso del software libre en nuestro país, no solo los beneficios económicos sino también la posibilidad de usar un software que es libre en su definición y no en el sentido financiero.

1.3 Requerimientos.

Uno de los aspectos más importantes en el proceso de desarrollo de software es determinar qué necesidades del cliente debe cubrir el sistema. Las capacidades (requerimientos) que los clientes exigen a las aplicaciones cubren necesidades referentes básicamente a funcionalidad (qué debe hacer el sistema), almacenamiento de información (qué información se va a manejar en el sistema) u otras características no funcionales como la facilidad de uso y el rendimiento. Según La IEEE Standard Glossary of Software Engineering Terminology define un requerimiento como:

1. Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
2. Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.
3. Una representación documentada de una condición o capacidad como en 1 o 2.

Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos.

1.3.1 Características de los requerimientos.[9]

Los requerimientos de software presentan algunas particularidades, entre estas se encuentran:

- **Deben ser especificados por escrito.** Como todo contrato o acuerdo entre dos partes.
- **Posibles de probar o verificar.** Si un requerimiento no se puede comprobar, entonces ¿cómo sabemos si cumplimos con él o no?
- **Descritos como una característica del sistema a entregar.** Esto es: **que** es lo que el sistema debe de hacer (y **no como** debe de hacerlo)
- **Lo más abstracto y conciso posible.** Para evitar malas interpretaciones.

La especificación de los requerimientos debe ser precisa, completa y clara. Para verificar lo anterior se presentan los siguientes aspectos cuya valoración objetiva podría ser de gran utilidad en la especificación de los requerimientos.

Precisa

- ¿Son los requerimientos consistentes? ¿No se contradicen los unos con los otros?
- ¿Algún requerimiento se encuentra en conflicto con algo que ya se ha declarado o restringido (entorno del negocio, entorno técnico, costo, planificación, y recursos)?

- ¿Soportan los requerimientos los objetivos del negocio, sistema de software y el proyecto?
- ¿Son necesarias todas las actividades y operaciones?
- ¿Algún requerimiento no es requerido o se encuentra fuera del alcance del proyecto?

Completo

- ¿Son las metas y objetivos del sistema de software clara y completamente definidos?
- ¿Se han manejado todos los eventos y condiciones?
- ¿Han sido especificadas todas las operaciones?
- ¿Son las operaciones suficientes para reunir los objetivos del sistema de software?
- ¿Se han especificados todas las definiciones y reglas requeridas?
- ¿Satisface las especificaciones el nivel de detalle requerido el equipo de diseño?

Claros

- ¿Los requerimientos se encuentran limpios de polarización de implementación (no restringidos a una alternativa de diseño específica)?
- ¿Se han declarado en forma precisa y concisa todos los requerimientos?

Las respuestas obtenidas a las interrogantes anteriores nos permiten tener una noción objetiva de lo correcto o no de la especificación de los requerimientos extraídos a lo largo de la vida del software.

1.3.2 Requerimientos funcionales y no funcionales.[10]

Los requisitos se pueden clasificar en: funcionales y no funcionales.

■ Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Estos no alteran la funcionalidad del producto, esto quiere decir que los requerimientos funcionales se mantienen invariables sin importarle con que propiedades o cualidades se relacionen.

■ Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

1.3.2.1 Categorías de los requerimientos no funcionales.

Existen múltiples categorías para clasificar a los requerimientos no funcionales, siendo las siguientes las más representativas de un conjunto de aspectos que se deben tener en cuenta, aunque no limitan a la definición de otros.

- **Requerimientos de Software:** Debe mencionarse el software del que se debe disponer, por ejemplo: Sistema Operativo Windows 95 o Superior; Máquina Virtual de Java versión 1.3 o Superior; etc.
- **Requerimientos de Hardware:** Al igual que en la sección anterior enunciar aquí los elementos de hardware de que se disponen, por ejemplo: se requiere disponer de un MODEM estándar o una tarjeta digitalizadora de video, etc.
- **Restricciones en el diseño y la implementación:** Este tipo de requerimiento especifica o restringe la codificación o construcción de un sistema, son restricciones que han sido ordenadas y deben ser cumplidas estrictamente. Ejemplos de ellas son:
 - Estándares requeridos.
 - Lenguajes de programación a ser usados para la implementación.
 - Uso obligatorio de ciertas herramientas de desarrollo.
 - Restricciones en la arquitectura o el diseño.
 - Bibliotecas de clases.
- **Requerimientos de apariencia o interfaz externa:** Este tipo de requerimiento describe la apariencia del producto. Es importante destacar que no se trata del diseño de la interfaz en detalle sino que especifican cómo se pretende que sea la interfaz externa del producto. Los requerimientos de apariencia se vuelven más importantes a medida que los productos de software se mueven hacia áreas más orientadas al consumidor. Requerimientos de apariencia también pueden ser necesidades de cumplir con normas estándares como por ejemplo las interfaces de sistemas tipo Windows/Apple/Motif, o con los estándares de la empresa para la cual se esté desarrollando el software.
- **Requerimientos de Seguridad:** Este es quizás el tipo de requerimiento más difícil, que provocará los mayores riesgos si no se maneja correctamente. La seguridad puede ser tratada en tres aspectos diferentes:
 - **Confidencialidad:** La información manejada por el sistema está protegida de acceso no autorizado y divulgación.

- **Integridad:** la información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos. Pueden incluir también mecanismos de chequeo de integridad y realización de auditorías.
- **Disponibilidad:** Significa que los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

La seguridad de un sistema no solo tiene en cuenta la seguridad del sistema propiamente dicho sino, además, el ambiente en el que se usará el sistema. Por lo que se tiene que contemplar la seguridad física del lugar donde se usa la aplicación, los controles administrativos que se establecen de acceso al sistema y las regulaciones legales que afecta o determina el uso del sistema y que serán tenidas en cuenta si se incumple.

La seguridad es un requerimiento no funcional que genera posiblemente requerimientos funcionales en dependencia de la propuesta de seguridad que se tenga para el sistema.

■ **Requerimientos de Usabilidad:** Estos requerimientos describen los niveles apropiados de usabilidad, dados los usuarios finales del producto, para ello debe revisarse la especificaciones de los perfiles de usuarios y echarle un vistazo a las clasificaciones de sus niveles de experiencia.

Debe conocerse:

- ¿Qué tipo de personas son?
- ¿Qué tipo de producto necesitan para realizar su trabajo?
- ¿Qué tipo de requerimiento de usabilidad haría el producto adecuado para ellos?

Los requerimientos de usabilidad se derivan de una combinación de lo que el cliente está tratando de lograr con el producto y lo que los usuarios finales esperan del mismo, estos elementos deben tenerse claro antes de escribirlos.

■ **Requerimientos de Soporte:** Abarcan todas las acciones a tomar una vez que se ha terminado el desarrollo del software con motivos de asistir a los clientes de este así como lograr su mejoramiento progresivo y evolución en el tiempo.

También se tienen:

- **Requerimientos Legales**
- **Requerimientos de confiabilidad**
- **Requerimientos de interfaz Interna**

Todos estos trazan las características que permitirán que el producto de software sea vendido y aceptado por el usuario.

1.3.3 Captura de requerimientos

El mayor problema actualmente en la industria del software es el inadecuado entendimiento de las necesidades de los usuarios, la incapacidad de absorber cambios en los requisitos y las seguidas insatisfacciones de los clientes por inaceptable o bajo desempeño del software. Últimamente se ha convertido en una tendencia la poca comunicación entre los clientes y los desarrolladores de software, motivo a esta situación es que se plantean un conjunto de técnicas que permiten mejoras en la captura de las funcionalidades del software.

Primeramente hay que tener una idea inicial para determinar QUÉ hacer, no CÓMO hacerlo, presentar un enfoque preliminar para identificar actores, procesos y recursos que apoyan al sistema actual, así como documentar todas las relaciones del sistema.

Se deben priorizar los requerimientos dividiéndolos en esenciales (deben ser cumplidos), deseables pero no esenciales y por último opcionales pero eliminables.

Entre las técnicas en la captura de requerimientos encontramos el procedimiento para el desarrollo del proceso de ingeniería de requisitos (PROCIR), este está implementado y documentado por el Departamento de Sistemas y Computación del Instituto Tecnológico de Morelia en México. Plantea entre sus premisas:

- Motivar a las organizaciones a aplicar un enfoque basado en procesos
- Su diseño permite aplicarlo a cualquiera de los tipos de software existentes.
- Estimular a las organizaciones de software a desarrollar las actividades de la ingeniería de requisitos de forma organizada, sistemática y repetitiva, con el objetivo de crear una base de conocimiento que les permita aplicar las mejores prácticas para obtener mayor grado de satisfacción del cliente.
- Provocar la participación de clientes y usuarios en todas las actividades del proceso, involucrándolos en la toma de decisiones de los cambios solicitados

El procedimiento se basa en los principios, métodos y técnicas de la ingeniería de requisito, está enfocado a obtener una especificación de requisitos del software con calidad a través del desarrollo de la gestión de los requisitos que la conforman a lo largo de todo el ciclo de vida del proyecto software, y en él se integran clientes, usuarios, desarrolladores y otros implicados, facilitando una mejor y más rápida comprensión del problema. [11]

Otra de las técnicas más usadas en la captura de requerimientos la conforman el conjunto de entrevistas y encuentro con los especialistas en información. También encontramos las tradicionales encuestas a los usuarios y el proceso de levantamiento de información en las diferentes áreas y departamentos del centro en donde se implementará el sistema. Estas fueron las usadas en el levantamiento de los requerimientos del presente trabajo. Los autores realizaron una serie de encuentros con las diferentes direcciones de la Universidad y trabajaron en conjunto con los especialistas de información encargados de realizar toda la arquitectura y definición de procesos y servicios que el Portal soportará.

1.3.4 Técnicas de validación de requerimientos

La validación de requerimientos es muy importante, pues un levantamiento de requerimientos con errores que no se detecten a tiempo, además de no conducir a resultados inesperados provoca costos excesivos y gran pérdida de tiempo.

Un prototipo es una versión inicial de un sistema de software que se utiliza para demostrar los conceptos, probar las opciones de diseño y entender mejor el problema, su solución y para conocer si las funcionalidades obtenidas responden verdaderamente a lo esperado por el usuario.

Un prototipo puede revelar errores u omisiones en los requerimientos propuestos, favorece la comunicación entre clientes y desarrolladores, da una primera visión del producto.

La realización de prototipos es una de las propuestas mas usadas en la validación de los requerimientos actualmente por sus beneficios que reportan a los usuarios y desarrolladores. [12]

1.4 Sistemas Gestores de contenido (CMS).

El Content Management System o sistema gestor de contenidos permite la gestión y administración de portales y ha sido una revolución para Internet.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio (Fig. 1). Permite manejar de manera independiente el contenido por una parte y el diseño por otra. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior que permite que estos contenidos sean visibles a todo público

1.4.1 Categorías de los CMS

Existen diferentes tipos de CMS y se pueden agrupar en las siguientes categorías:

Sitios web: Sitio web con contenido y funcionalidad diversa que sirve como fuente de información o como soporte a una comunidad.

Algunos CMS para Sitios web:

- ASP Nuke (ASP): Uno de los pocos CMS basados en ASP de código libre.
- door108 (ex DoorSoft) (PHP): CMS totalmente en español. Incluye novedades como Multitarea.
- DotNetNuke (.NET): CMS desarrollado en .NET, gratis y con fuentes. Más que un CMS en sí es un Framework de .NET pensado para desarrollar CMS entre otras cosas.
- Drupal (PHP): Poderoso CMS muy conocido por la calidad de su código y por la seguridad que brinda, es estable y de actualización continua, configuración sencilla, instalación ágil, importante cantidad de módulos y themes, excepcional documentación, comunidad activa y muy amigable, gran concepto de nodo.

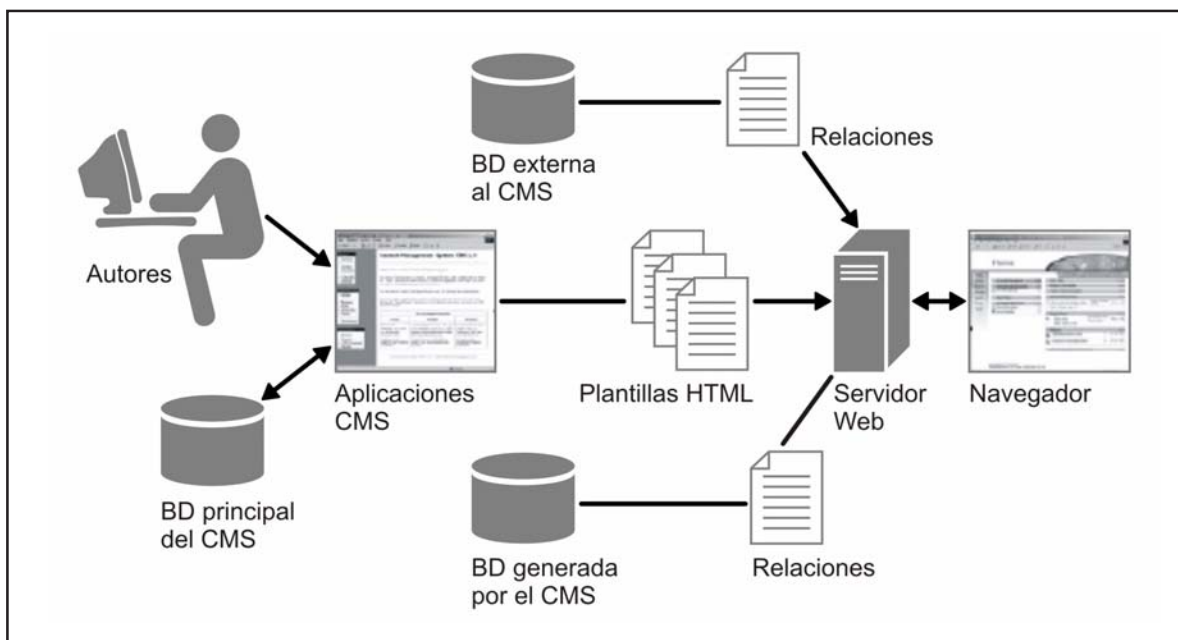


FIG. 1 ESQUEMA GENERAL DE UN CMS.

- E107 (PHP): CMS muy completo y fácil de administrar y usar, ideal para usuarios con conocimientos generales acerca de estos sistemas. Sencillo sistema de instalación, amplia selección de themes y módulos, muy flexible, backend muy bien ordenado, drop down menú agradable y organizado.
- Gekko (PHP): CMS en español muy fácil de configurar y altamente escalable.
- Jaws (PHP): Framework y CMS amigable para el usuario y desarrollador.
- Joomla (PHP / MySQL): Versión surgida de Mambo independiente de la empresa que está detrás de Mambo. Instalación muy sencilla y con muchas extensiones y módulos, la documentación es exhaustiva y concisa, interface de la administración muy intuitiva y poderosa, backend muy utilizable y editor.
- Mambo (PHP): CMS muy fácil de usar, pero con posibilidades un poco limitadas.
- MODxPHP MODx: Es un derivado (Fork) de Etomite, resulta ser un CMS más versátil que otros demasiado estructurados.
- Openflavor (PHP): Gestor de contenidos web en castellano.
- Plone (Zope/Python): Muy flexible y poderoso, excelente interface de usuario, instalación muy limpia, buena cantidad de addons, impresionante grado de personalización, integración con LDAP u otros sistemas de login.
- phpwcms (PHP): CMS orientado a la construcción de sitios web para profesionales y empresas.

- Postnuke (PHP): Poderoso CMS/Web Framework modular con motor de themes para una interfaz de usuario muy flexible y fácil de mantener, con gran cantidad de módulos para toda necesidad, con un Network Operations Center para soportar una gran comunidad de desarrollo muy activa, y con un código fuente muy limpio y de alta calidad.
- Post Revolution (PHP): CMS desarrollado en español, especializado en weblogs, de código libre bajo licencia GNU/GPL
- SPIP (PHP). Gestor de Contenido de licencia libre.
- Typo3 (PHP): CMS muy potente, pero muy difícil de utilizar pues utiliza un lenguaje de scripts para las plantillas.
- WebGUI (Perl): Ocupa más de 40 MB, flexible, adaptable, multilingüe.
- Xaraya (PHP): Es un CMS bastante potente y general, aunque con una elevada curva de aprendizaje.
- XOOPS (PHP): CMS modular. Instalación sencilla, gran soporte comunitario, gran cantidad de módulos y themes, mucha funcionalidad, sistema de permisos muy bueno.

Foros: sitio que permite la discusión en línea donde los usuarios pueden reunirse y discutir temas en los que están interesados.

Algunos CMS para Foros:

- phpBB (PHP/MySQL).
- MyBB (PHP/MySQL).
- SMF (PHP/MySQL).

Blogs: Publicación de noticias o artículos en orden cronológico con espacio para comentarios y discusión.

CMS para Blogs:

- WordPress (PHP/MySQL).
- b2evolution.net (PHP/MySQL).
- pMachine Pro (PHP/MySQL).
- bBlog (PHP).
- Simple PHP Blog (PHP).
- DotClear (PHP/MySQL).
- Serendipity (PHP/MySQL).
- BLOG: CMS (PHP/MySQL).
- Plone

Wikis: Sitio web dónde todos los usuarios pueden colaborar en los artículos. También permite espacio para discusiones. Indicado para material que irá evolucionando con el tiempo.

CMS para Wikis:

- MediaWiki (PHP CMS que permite que todos puedan modificar el contenido).
- TikiWiki (PHP).

eCommerce: Sitio web para comercio electrónico.

CMS para eCommerce:

- osCommerce (PHP/MySQL).

Galería: Permite administrar y generar automáticamente un portal o sitio web que muestra contenido audiovisual, normalmente imágenes.

CMS para galerías:

- Gallery (PHP/MySQL).
- coppermine (PHP/MySQL).

CMS para groupware:

- Webcollab (PHP/MySQL).

1.4.2 Características del CMS Plone.

1.4.2.1 CMS Plone.

Plone es un Sistema de Gestión de Contenidos o CMS por sus siglas en inglés (Content Management System), basado en Zope y programado en Python. Está basado en código abierto y puede utilizarse como servidor intranet o extranet, como un Sistema de Publicación de documentos y como una herramienta de trabajo en grupo para colaborar entre entidades distantes. Entre sus características más significativas podemos encontrar que:

- Es una herramienta atractiva y fácil de usar en la creación, y actualización de contenidos.
- Cuenta con la traducción en más de 50 lenguajes, además existen herramientas para crear contenidos en más de un lenguaje.
- Tiene licencia GPL, la cual permite la libre distribución, uso, y cambios de la aplicación.
- Cuenta con un equipo de cientos de desarrolladores de diferentes lugares del mundo.
- Usa estándares de usabilidad y accesibilidad. Usa los estándares web XHTML y CSS.
- Se puede extender integrándole nuevas funcionalidades desarrolladas por nuestros equipos o por otros de otras partes del mundo.
- La aplicación puede correr en Linux, Windows, Mac.

- Periódicamente se realizan actualizaciones de la tecnología, donde se agregan nuevas funcionalidades, y se corrigen defectos detectados por los usuarios.

Como servidor de aplicaciones web utiliza Zope servidor escrito en el lenguaje de programación Python. Puede ser manejado casi totalmente usando una interfaz de usuario basada en páginas Web.

Un sitio web de Zope está compuesto de objetos en lugar de archivos, como es usual con la mayoría de los otros sistemas de servidores web. Las ventajas de usar objetos en lugar de archivos son:

- Combinan el comportamiento y los datos en una forma más natural que los archivos de texto plano.
- Alientan el uso de componentes estándares que se ocupan de una parte particular de las que forman una aplicación Web, permitiendo flexibilidad y buena descomposición.
- Posibilitan procesos automáticos de gestión de información.

Python es un lenguaje de programación creado por Guido van Rossum en el año 1990. Es comparado habitualmente con TCL, Perl, Scheme, Java y Ruby. En la actualidad Python se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation. Python posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la licencia GPL. Esta licencia no obliga a liberar el código fuente al distribuir los archivos binarios.

1.4.3 Fundamentación de la selección del CMS.[13]

Después de haber analizado las características de los CMS y a partir de las características de la nueva Intranet a implementar se eligió el CMS Plone para su realización, este plantea ser ideal para servicios de Intranet y presenta diferentes herramientas colaborativas en sus disímiles módulos. Por estas y otras particularidades que se presentan a continuación se optó por su elección.

- (CMS) escrito en Python popular gracias a su estabilidad, escalabilidad y enorme potencial pues está construido sobre Zope, un poderoso servidor de aplicaciones que posee su propio Webserver, gestor de sesiones y base de datos.
- La interface de Plone es compatible con los estándares de CSS y de XHTML. EL uso intensivo de CSS permite de beneficiar de una interface ligera. En nativo, cada página del sitio es optimizada por la impresión. También cada página tiene su propio modo de visualización.
- Sobre la seguridad pone a disposición de los usuarios, roles y grupos con mucha flexibilidad. También es posible de manejar localmente la seguridad y no solamente al nivel del conjunto del sistema.
- Es publicado bajo la licencia GNU (General Public License).
- Enfocado a la producción de los contenidos y no en la tecnología.

Estas son solo algunas características que el CMS Plone ofrece para la realización de sitios web colaborativos.

1.5 Sistemas gestores de Base Datos (SGBD)

Los Sistemas de Gestión de Base de Datos (SGBD) se pueden definir como el «conjunto de herramientas que suministra a todos (administrador, analistas, programadores, usuarios) los medios necesarios para describir, recuperar y manipular los datos almacenados en la BD, manteniendo la seguridad, integridad y confidencialidad de los mismos». [14]

Entre los sistemas Gestores de Bases de Datos más usados actualmente encontramos MySQL, Oracle, SQL Server, Microsoft SQL Server 2000, PostgreSQL. Etc.

1.5.1 Características de los SGBD

Un sistema de gestión de bases de datos se puede definir como una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. Es un conjunto coordinado de programas, procedimientos, lenguajes, etc. que suministra, tanto a los usuarios no informáticos como a los analistas, programadores o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad.

Existen dos grandes modelos de sistemas de gestión de bases de datos:

- **Sistemas de Gestión de Bases de Datos Relacionales (SGBDR):** Las bases de datos que generan se construyen con información muy estructurada (datos) acerca de una organización o empresa determinada. Cuando un usuario realiza una consulta en una base de datos relacional, el sistema presenta como resultado la respuesta exacta a lo que se busca. A este tipo de bases de datos se les denomina bases de datos relacionales, y a los sistemas que las gestionan, Sistemas de Gestión de Bases de Datos Relacionales (SGBDR).
- **Sistemas de Gestión de Bases de Datos Documentales (SGBDD) o Sistemas de Recuperación de Información (SRI):** Las bases de datos que generan se construyen con información no estructurada tipo texto (documentos) sobre uno o varios temas. Cuando un usuario realiza una consulta en una base de datos documental, el sistema presenta como resultado, no una respuesta exacta, sino documentos útiles para satisfacer la pregunta del usuario. A este tipo de bases de datos se les denomina bases de datos documentales, y a los sistemas que las gestionan, Sistemas de Gestión de Bases de Datos Documentales (SGBDD) o Sistemas de Recuperación de Información (SRI). [15]

1.5.1.1 MySQL

MySQL es un SGBD multiplataforma, es un software de fuente abierta lo que significa que cualquier persona puede usarlo y modificarlo para satisfacer sus necesidades, puede ser utilizado gratuitamente. Se encuentra bajo la licencia GPL.

Por su sencillez y sus características es usado por muchas personas, consume muy pocos recursos, se usa tanto en aplicaciones sencillas como complejas. Es utilizado en aplicaciones Web como Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/

Python), y por herramientas de seguimiento de errores como Bugzilla, además sus conexiones son muy seguras.

1.5.1.2 Oracle (Relational Data Base Management System)

Es considerado uno de los sistemas de bases de datos más completos, por su estabilidad, soporte de transiciones, escalabilidad y ser multiplataforma, sin embargo el precio de Oracle es sumamente caro. Otro aspecto que ha sido criticado por algunos especialistas es la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre de 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años.

1.5.1.3 Microsoft SQL Server 2000.

Tiene capacidad para consultar la base de datos mediante un explorador y es compatible con el lenguaje de marcado extensible XML (del inglés *Extensible Markup Language*). Es un gestor de bases de datos totalmente habilitado para Web, exhibiendo sellos de referencia en cuanto a escalabilidad y confiabilidad. Se puede decir que este gestor permite lograr una gran velocidad en el procesamiento de transacciones, y agilidad en todas sus operaciones. Permite gran seguridad de los datos.

1.5.1.4 PostgreSQL

Es un magnífico gestor de bases de datos, es multiplataforma. Se encuentra bajo la licencia BSD. Permite una fácil gestión de los usuarios y de las bases de datos que contenga el sistema. Sirve de soporte a los lenguajes más populares como PHP, C, C++, Java, Python, Ruby, etc, y al protocolo de comunicación encriptado por SSL. El número de base de datos que puede contener es ilimitado. [16]

1.5.2 Fundamentación de la selección del SGBD.

A partir de las características que ofrecen los distintos SGBD se decidió escoger para la realización de las BD del portal a PostgreSQL como gestor, éste un fuerte sistema gestor de BD de código abierto. En más de 15 años de desarrollo presenta una poderosa arquitectura e integridad en sus datos, se ejecuta en cualquier Sistema operativo incluido Linux, Unix, Mac, Solaris y otros. Trae incluido diferentes tipos de datos: enteros, numéricos, caracteres, variables, intervalos y otros. También soporta objetos binarios, sonido, imágenes y videos. [17] permite agrupar un bloque de sentencias en el servidor y ejecutar cálculos. Al utilizar PostgreSQL se disipa cualquier dificultad a la hora de seleccionar el tipo de tabla que se va a utilizar debido a que cuenta con un único mecanismo de almacenamiento.

1.6 Metodologías de desarrollo del software.

1.6.1 Proceso de desarrollo de software.

Un proceso define «quién» está haciendo «qué», «cuándo» y «cómo» para alcanzar un determinado objetivo. Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto.

El proceso de desarrollo de software tiene la misión de transformar los requerimientos del usuario en un producto de software; de manera que los integrantes del equipo y todo aquel que pueda estar interesado en el producto final, tenga la misma visión y no ocurra cuando no se aplica un proceso de desarrollo.

Por lo tanto, las piedras angulares del proceso de desarrollo del software son: el proyecto, las personas y el producto; siendo las características del cliente, el entorno de desarrollo y las condiciones del negocio, elementos que influyen en el proceso. Existe una estrecha relación entre personas, proyecto, producto y proceso. Estos términos son conocidos como las cuatro «P» en el desarrollo de software.

El resultado final de un **proyecto** software es un producto, donde intervienen **personas** a través de un **proceso** de desarrollo de software que guía los esfuerzos de las personas implicadas en el proyecto. [18]

1.6.2 Metodologías.

En un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Una metodología es un proceso.

No existe una metodología de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable

En la actualidad existen varias metodologías OO(Orientada a Objetos) basadas en UML: Rational Unified Process (RUP), Extreme Programing (XP), OPEN, MÉTRICA 3, Microsoft Solution Framework (MSF) y otras más.

Sabiendo que (UML) Lenguaje Unificado de Modelado es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software y que este está compuesto por diversos elementos gráficos que se combinan para conformar diagramas.

Este lenguaje que permite la modelación de sistemas con tecnología orientada a objetos y describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

Ya planteado esto, se hará un estudio de algunas de las diferentes metodologías planteadas.

1.6.2.1 El Proceso Unificado de Desarrollo (RUP)

RUP es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los

clientes. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo.

Es una de las metodologías más generales y más usadas de las que existen en la actualidad, pues está pensada para adaptarse a cualquier proyecto. Constituye además, una propuesta de proceso para el desarrollo de software orientado a objeto, utilizando UML (del inglés Unified Model Language), para describir todo el proceso, basándose en componentes. Este lenguaje es estándar, con él se puede visualizar, especificar, construir y documentar los artefactos de un sistema.

Las principales características de esta metodología son: [19]

- **Centrado en los Modelos:** Los diagramas son un vehículo de comunicación más expresivo que las descripciones en lenguaje natural. Se trata de minimizar el uso de descripciones y especificaciones textuales del sistema.
- **Guiado por los casos de uso:** Los casos de uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.
- **Centrado en la arquitectura:** Los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar.
- **Iterativo e incremental:** Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo.

El RUP incluye cuatro etapas importantes que son: Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones. Estas etapas revelan que para producir una versión del producto en desarrollo se emplean todas las actividades de ingeniería pero con diferente énfasis; en las primeras versiones se hace más énfasis en el modelado del negocio, requisitos, análisis y diseño; mientras en las posteriores el énfasis recae sobre las actividades de implementación, pruebas y despliegue. Además contempla flujos de trabajo de soporte que involucran actividades de planificación de recursos humanos tecnológicos y financieros.

Como RUP es un proceso, en su modelación se define como sus principales elementos:

- **Trabajadores (¿Quién?):** Definen el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- **Actividades (¿Cómo?):** Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- **Artefactos (¿Qué?):** Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- **Flujo de actividades (¿Cuándo?):** Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

En él se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. [20]

Flujos de trabajo:

- **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a los largo del ciclo de vida.
- **Instalación:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

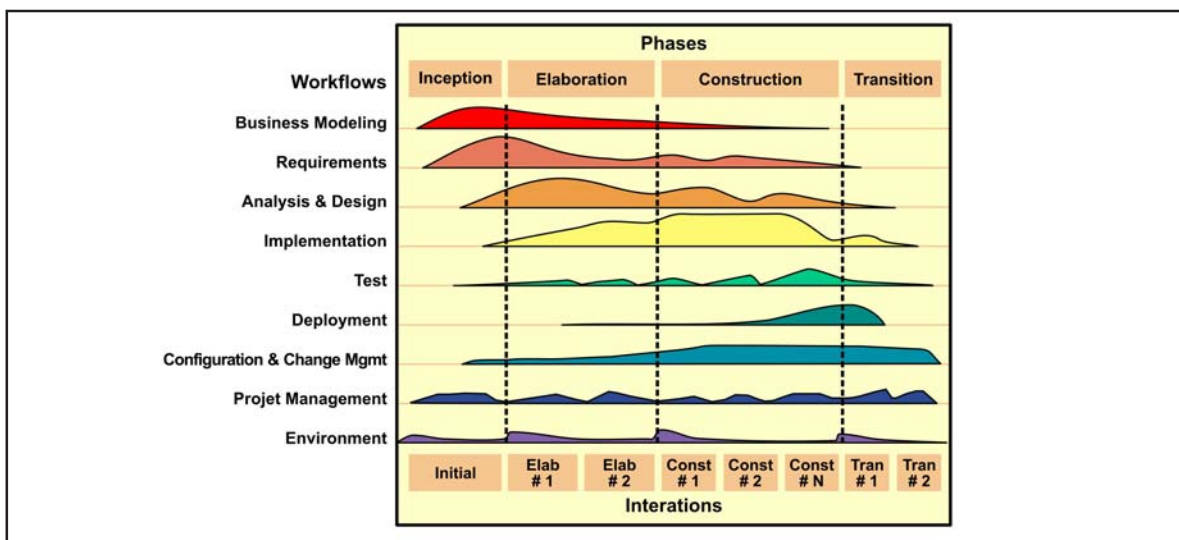


FIG. 2 REPRESENTACIÓN DEL PROCESO DE DESARROLLO DE SOFTWARE

1.6.2.2 Extreme Programing (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizadas para proyectos de corto plazo, corto equipo y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Características de XP, la metodología se basa en:

- **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- **Refabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

¿Qué es lo que propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.
- Derechos del Cliente.
 - Decidir que se implementa.
 - Saber el estado real y el progreso del proyecto.
 - Añadir, cambiar o quitar requerimientos en cualquier momento.
 - Obtener lo máximo de cada semana de trabajo.
 - Obtener un sistema funcionando cada 3 o 4 meses.

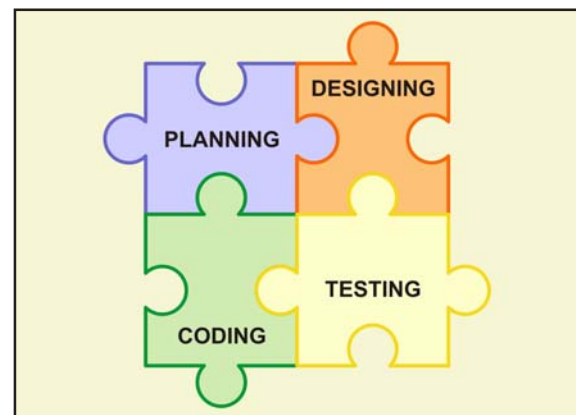


FIG. 3 METODOLOGÍA EXTREME PROGRAMING.

Entre los derechos del Desarrollador encontramos que este es capaz de decidir como se implementan los procesos y tiene la libertad de crear el sistema con la mejor calidad posible. Le puede pedir al cliente en cualquier momento aclaraciones de los requerimientos y estima

el esfuerzo para implementar el sistema. Cambiar los requerimientos en base a nuevos descubrimientos siempre que estén aparezcan en el ciclo de vida del software.

Lo fundamental en este tipo de metodología es:

- La comunicación, entre los usuarios y los desarrolladores
- La simplicidad, al desarrollar y codificar los módulos del sistema
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales

1.6.2.3 Microsoft Solution Framework (MSF).[21]

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

Entre las características más significativas de esta metodología encontramos:

- **Adaptable:** es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- **Escalable:** puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.
- **Tecnología Agnóstica:** porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

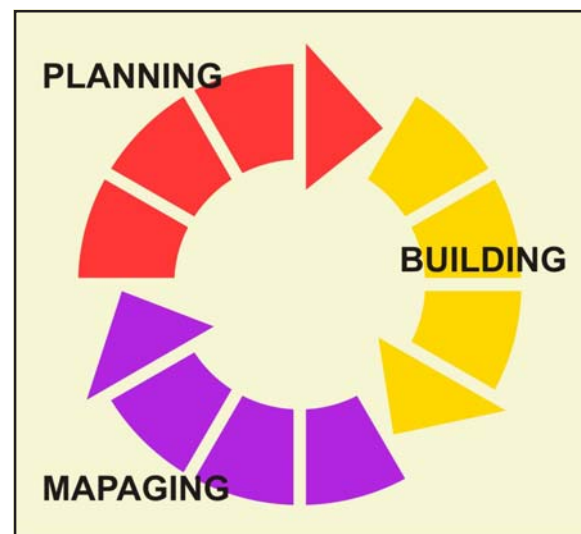


FIG. 4 METODOLOGÍA MSF.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto:

- Modelo de Arquitectura del Proyecto.
- Modelo de Equipo.
- Modelo de Proceso.
- Modelo de Gestión del Riesgo.
- Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

1.6.2.4 Métrica.[22]

La metodología MÉTRICA ofrece en sus últimas versiones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software dentro del marco que permite alcanzar los siguientes objetivos:

- Proporcionar o definir Sistemas de Información que ayuden a conseguir los fines de la Organización mediante la definición de un marco estratégico para el desarrollo de los mismos.
- Dotar a la Organización de productos software que satisfagan las necesidades de los usuarios dando una mayor importancia al análisis de requisitos.
- Mejorar la productividad de los departamentos de Sistemas y Tecnologías de la Información y las Comunicaciones, permitiendo una mayor capacidad de adaptación a los cambios y teniendo en cuenta la reutilización en la medida de lo posible.
- Facilitar la comunicación y entendimiento entre los distintos participantes en la producción de software a lo largo del ciclo de vida del proyecto, teniendo en cuenta su papel y responsabilidad, así como las necesidades de todos y cada uno de ellos.
- Facilitar la operación, mantenimiento y uso de los productos software obtenido.

Su estructura cubre distintos tipos de desarrollo: estructurado y orientado a objetos, facilitando a través de interfaces la realización de los procesos de apoyo u organizativos: Gestión de Proyectos, Gestión de Configuración, Aseguramiento de Calidad y Seguridad.

Presenta un enfoque orientado a los procesos, que se centra en la clasificación y definición del ciclo de vida del software.

Esta metodología descompone cada uno de los procesos en actividades, y éstas a su vez en tareas. Para cada tarea se describe su contenido haciendo referencia a sus principales acciones, productos, técnicas, prácticas y participantes.

El orden asignado a las actividades no debe interpretarse como secuencia en su realización, ya que éstas pueden realizarse en orden diferente a su numeración o bien en paralelo. Sin embargo, no se dará por acabado un proceso hasta no haber finalizado todas las actividades del mismo determinadas al inicio del proyecto.

Así los procesos de la estructura principal de MÉTRICA son los siguientes:

- Planificación de sistemas de información.
- Desarrollo de sistemas de información.
- Mantenimiento de sistemas de información.

1.6.3 Fundamentación de la selección de la metodología de desarrollo de software.

El software es un componente esencial de toda actividad basado en el uso de la informática, es por ello que la calidad en su desarrollo y mantenimiento resulta hoy en día uno de los principales objetivos estratégicos de las organizaciones. En los últimos años se han publica-

do diversos estudios en los que se presentan los principios que se deben seguir para mejorar los procesos de software, y de esta forma evitar las grandes catástrofes que conllevan al fracaso de un gran número de proyectos.

La metodología para el desarrollo del software surgió de la construcción del marco conceptual que integra el enfoque Constructivista, el Pensamiento Sistémico junto con la Dinámica de Sistemas. Ésta constituye un conjunto de procedimientos, reglas, herramientas y aspectos de formación para los desarrolladores de aplicaciones Informáticas. [23]

Para controlar, y concebir la propuesta de este sistema, se decide utilizar como metodología el Proceso Unificado de Modelado (RUP), por todas las ventajas de organización que brinda y debido a que goza de un grupo de características y facilidades, que hacen mas dinámico el desarrollo del trabajo.

1.6.4 Herramientas de modelo visual.

Las herramientas de modelaje visual permiten la representación de un producto de software en su totalidad mediante diagramas que se van desarrollando durante los diferentes ciclos de vida del proyecto y facilitan el desarrollo de un proceso. Entre las más usadas actualmente encontramos el Rational Rose herramienta CASE que proporciona mecanismos para realizar Ingeniería Directa e Inversa, posibilita la construcción de un modelo de casos de usos, identifica los objetos y representa cómo interactúan con los diagramas de secuencia y colaboración, así como otras operaciones. También encontramos otra herramienta muy popular por sus disímiles características, el Visual Paradigm producto de calidad, que soporta aplicaciones Web, es muy fácil de instalar y actualizar. Permite la generación de código para varios lenguajes y presenta un entorno de creación de diagramas para UML 2.0. Su diseño está centrado en casos de uso y enfocado al negocio generando un software de mayor calidad, presenta capacidades de ingeniería directa e inversa y disponibilidad en múltiples plataformas.

A partir de las herramientas de modelado conocidas y teniendo en cuenta las características de ambas se decidió escoger para la modelación del presente trabajo la herramienta Visual Paradigm partiendo que es una herramienta de software libre y por todas sus ventajas.

1.7 Análisis y Diseño de sistemas.

1.7.1 Conceptos.

1.7.1.1 Análisis de Sistemas.

El Análisis de Sistemas trata básicamente de determinar los objetivos y límites del sistema objeto de análisis, caracterizar su estructura y funcionamiento, marcar las directrices que permitan alcanzar los objetivos propuestos y evaluar sus consecuencias. Dependiendo de los objetivos del análisis podemos encontrarnos ante dos problemáticas distintas:

- Análisis de un sistema ya existente para comprender, mejorar, ajustar y predecir su comportamiento.
- Análisis como paso previo al diseño de un nuevo sistema-producto.

Las tareas que constituyen el análisis se dividen en una serie de etapas que se suceden de forma iterativa hasta validar el proceso completo:

- **Conceptualización:** Consiste en obtener una visión de muy alto nivel del sistema, identificando sus elementos básicos y las relaciones de éstos entre sí y con el entorno.
- **Análisis funcional:** Describe las acciones o transformaciones que tienen lugar en el sistema. Dichas acciones o transformaciones se especifican en forma de procesos que reciben una entrada y producen una salida.
- **Análisis de condiciones (o constricciones):** Debe reflejar todas aquellas limitaciones impuestas al sistema que restringen el margen de las soluciones posibles.
- **Construcción de modelos:** Una de las formas más habituales y convenientes de analizar un sistema consiste en construir un prototipo (un modelo en definitiva) del mismo.
- **Validación del análisis:** A fin de comprobar que el análisis efectuado es correcto y evitar en su caso la posible propagación de errores a la fase de diseño, es imprescindible proceder a la validación del mismo.

1.7.1.2 Diseño de Sistemas

El diseño de sistemas se ocupa de desarrollar las directrices propuestas durante el análisis en términos de aquella configuración que tenga más posibilidades de satisfacer los objetivos planteados tanto desde el punto de vista funcional como del no funcional (lo que antes hemos denominado constricciones). El proceso de diseño de un sistema complejo se suele realizar de forma descendente:

- Diseño de alto nivel (o descomposición del sistema a diseñar en subsistemas menos complejos).
- Diseño e implementación de cada uno de los subsistemas:
 - Especificación consistente y completa del subsistema de acuerdo con los objetivos establecidos en el análisis.
 - Desarrollo según la especificación.
 - Prueba.
- Integración de todos los subsistemas.
- Validación del diseño.

1.7.2 Tendencia del Diseño.

Realmente las tendencias del diseño son muy variadas porque existen diferentes tipos de diseños tal como el diseño gráfico, industrial, y los que nos interesa que es el Web y el de software.

El diseño web actualmente con el desarrollo de las tecnologías y la colocación de espacios en Internet como intranet, portales o simples sitios web que se dedican a comerciar o promocionar sus productos se han multiplicado de manera exponencial, sin embargo diariamente surgen y desaparecen nuevos sitios ante el difícil reto de conjugar las virtudes interactivas de la Red de Redes con una propuesta gráfica atractiva y un esquema de navegación intuitivo.

Tanto el diseño de sistemas, de aplicaciones web incluido el de Intranet se requiere de gran conocimiento de las herramientas que se usan para diseñar y se tiene aunque sea una idea de cómo se realiza aunque no se sigan patrones o estándares para el diseño.

A nivel mundial los diseñadores de sitios y sistemas de información son los que a diario no siguen estas reglas, incluso a veces no hay ni reglas para este diseño.

Y la primera premisa del diseño es contar con una buena imagen hacia quien se va a brindar dicho diseño o sea el usuario final que es el cliente o usuario que va recibir o va interactuar con este en el caso de un sitio Web. Por eso todo diseñador debe siempre regirse por distintas reglas para que el diseño sea aceptado por el usuario y estudiar lo que desean, porque a veces ni se toma en cuenta la opinión de estos. A medida que han pasado los años y con el actual desarrollo de la Internet el diseño se ha vuelto temas de discusión en foros y sitios de Internet.

Por colocar un sitio de una empresa u organización online requiere de gran trabajo y esfuerzo por parte del equipo o de la persona que realiza este diseño. Pero lo más importante para realizar un buen diseño es conocer sobre la actualidad y desarrollo de estándares de diseño o patrones (Design Patterns). Y por eso como al investigar sobre este tema encontramos varios casos y formas para diseñar software, sitios Web, que están muy relacionados porque los dos trabajan para lograr un mejor servicio al usuario.

A raíz del surgimiento de Internet los diseñadores todos los años se trazan nuevas metas y por eso es que vamos a ver en detalle como cambian las tendencias del diseñador medida que pasa el tiempo por eso en los últimos años han cambiado mucho por ejemplo para el año 2005 se centraba el diseño en disminuir la sencillez y centrarse en crear un diseño bien cargado en tres detalles fundamentales que transmitieran profesionalidad como: nuevos colores, nuevas estructuras, y nuevos aspectos.

En cuanto a estructura llegar a un estándar compuesto por cinco piezas específicas como:

- Cabecera.
- Menú superior.
- Menú lateral.
- Contenido.
- Pie.

Aunque el contenido puede cambiar debido a la aparición de nuevos estilos como el Wiki.

En cuanto a los colores que se usan para el diseño web seguramente predominen los colores marrones o tierra, y se dará un respiro a los tecnológicos gris y azules. Más amarillos y verdes, rojos y naranjas para hacer web de aspecto fresco.

En mi opinión la paleta de colores se esta empezando a quedar obsoleta y el amplio uso de colores frescos para que el usuario se sienta cómodo al navegar en ella.

El aspecto de las paginas web es el más difícil de solucionar pero por ejemplo para seleccionar el tipo de letra o tipografía casi todos los diseñadores apuestan a que se usara la Georgia para los títulos y Arial para el texto así como que hay concienciada en que las tablas ya no pueden ser usadas y que ya no aparezca en los entornos corporativos, también el flash se centrara en algo específico y dejaran de existir las páginas completas con esta tecnologías. En mi opinión en este año ya se ven claras las cosas de que el usuario que navega en Internet ya no es tan desconocedor de la tecnología sino que esta empapado en ella y por eso los diseñadores deben ser mucho mejores en sus diseños y aplicar los estándares e innovar respecto al diseño web.

Proseguimos con el avance de las tendencias del diseño web y ya para el año 2006 se trazaban nuevas debido a la aparición de nuevas tecnologías como IE 7 que posibilitara que nuestros diseños se adapten al nuevo navegador, principalmente los layout se acoplaran a resoluciones más altas como (1024x768), se incrementarán los layout líquidos y aparecerá el uso de los layout dinámicos que se basan en obtener el CSS específico para la resolución del monitor a través de funciones javascript.

Predominaran el diseño con una estructura de tres columnas donde el contenido estará en el lado izquierdo, en la columna central los menús y en la derecha la información adicional.

La tipografía aumentará de tamaño al aumentar las dimensiones del diseño y se empezará a utilizar (Times New Roman).

Los colores que más se usaran serán el negro, verde, celeste y blanco y los menús serán más simples y los flash solo se usaran como aplicaciones multimedia.

Así como el aumento del uso de los estándares XHTML y CSS, y la accesibilidad estará enmarcada la introducción y desarrollo de aplicaciones sociales y colaborativas con Ajax y Web 2.0.

En mi opinión este año hubo un vuelco en las tecnologías y avanza muchísimo la utilización de estándares web, con el predominio principalmente de las redes sociales.

En tal caso para el diseño de la primera versión de la Intranet utilizamos estándares como CSS y HTML.

Pero aun nos queda el año 2007 que veremos que depara este año para el diseño Web primero tenemos que se presentan grandes proyectos como el de la web semántica que se va ir introduciendo en ámbitos más institucionales y menos sociales al estilo de web 2.0. Se afianzara la web multimedia con videos y audio en las web.

Así como la aparición de varias tecnologías que harán la red de redes más navegable y más accesible por ejemplo:

- Los Blogs aumentaran sobre todo en las redes sociales adaptándose al mercado muchos de los elementos añadidos a los CMS.
- Los wikis hicieron su aparición en el recién terminado 2006. Ahora, en 2007, seguirán creciendo y activando nuevas propuestas en servidor. Debido a que el uso de wikis es muy flexible, con lo cuál muchos centros educativos lo usarán como página web.
- Aumentará la velocidad de navegación con la aparición del **VDSL**.

Así como la aparición de tecnologías como RSS que es muy usada para todo tipo de sitios que se actualicen frecuentemente y que socialicen información. También aparecerá la multimedia y la virtualidad y la puesta en marcha de la TV-Internet en fin este año estará enmarcado en pasos agigantados hacia el uso más frecuente de la tecnología de la información y las comunicaciones o TIC's y por eso los diseñadores tendrán que ampliar sus conocimientos en aras de lograr una web más accesible y más usable para el usuario que la explota.

El diseño de software tiende a utilizarse hoy en día en casi todas las empresas del mundo y no solo en las que se encargan el negocio de la informática sino en todos aquellos que deseen diseñar un producto con calidad, por eso la Ingeniería de software se dice que contiene todas las ingenierías.

El diseño de software incluye metodologías y herramientas que se pueden usar para cualquier tipo de software ya sea una aplicación web o aplicaciones de bancos, entre otras y es que el diseño de software se usa para todo tipo de software que se quiera producir, el uso de el lenguaje de modelado, UML par el diseño de software que incluye las principales metodologías de desarrollo de software orientadas a objetos hace que su pueda usar en cualquier aplicación que se vaya a implementar.

1.7.3 Patrones de Diseño.

1.7.3.1 Reseña Histórica.

El surgimiento de los Patrones de Diseño (Desing Patterns) comienza en el 1979 cuando el arquitecto Christopher Alexander aportó al mundo de la arquitectura el libro The Tímeless Way of Building, en él proponía el aprendizaje y uso de una serie de patrones para la construcción de edificios de una mayor calidad.

Posteriormente, en 1987, Ward Cunningham y Kent Beck usaron varias ideas de Alexander para desarrollar cinco patrones de interacción hombre-ordenador (HCI) y publicaron un artículo en OOPSLA-87 titulado Using Pattern Languages for OO Programs.

Pero no fue hasta principios de los 90's cuando los patrones de diseño tuvieron un gran éxito en el mundo de la informática a partir de la publicación del libro Design Patterns escrito por el GoF (Gang of Four) compuesto por Erich Gamma, Richard Helm, Ralph Johnson y John Vlisodes, en el que se recogían 23 patrones diseño comunes.

Uno de los conceptos de patrones de diseño más definido es el propuesto por el arquitecto **Christopher Alexander cuando dice que un patrón de diseño es: «una solución a un**

problema que se usa repetidamente en contextos similares con algunas variantes en la implementación». [24]

Existen patrones de diseño para cada rama de la ciencia y la ingeniería que estén involucradas en diseño de sistemas pero los fundamentales son los patrones de diseño web y de software.

1.7.4 Patrones de diseño web.

El principal problema de los diseñadores es que existen infinitudes de formas de realizar un diseño web, por eso a diario resulta muy difícil darse cuenta de si usamos patrones o no. Además hay millones de formas de resolver un problema y porque todos no pensamos y hacemos nuestro diseño como mejor nos convenga.

1.7.4.1 Tipos de patrones de diseño web.

Los patrones de diseño web pueden aplicarse a un nivel jerárquico, nivel de componente y nivel de hipertexto (de navegación). [25] Así como otros que veremos en esta sección.

Cuando en una web se requiere la funcionalidad del proceso de datos, puede hacerse uso de los patrones de diseño arquitectónicos a nivel de componentes.

Los patrones de diseño a nivel de hipertextos se centran en el diseño de las características de navegación que le permitan al usuario moverse por el contenido de la web.

Hay de hecho muchos pero los más importantes s patrones de diseño hipertexto son:

1. **Ciclo:** Configuración que devuelve al usuario al contenido visitado anteriormente.
2. **Anillo de web:** Una configuración que implementa un gran número de ciclos que enlaza hipertextos enteros viajando por un tema.
3. **Contorno:** Un patrón que aparece cuando varios ciclos inciden en otro, permitiendo navegar por rutas definidas por los ciclos.
4. **Contrapunto:** añade comentarios hipertextos interrumpiendo la narrativa del contenido para proporcionar más información o más indagación.
5. **Mundo espejo:** el contenido se presenta utilizando diferentes hilos narrativos, cada uno con un punto de vista o perspectiva diferente.
6. **Tamiz:** una configuración que va guiando al usuario a través de una serie de opciones (decisiones) con el fin de llevar al usuario a un contenido específico e indicado por la sucesión de opciones elegidas o decisiones tomadas.
7. **Vecindario:** una configuración que abarca un marco de navegación uniforme por todas la paginas Web para permitir que un usuario tenga una guía de navegación consecuente independientemente de la localización de la Web.

Estas configuraciones se pueden ir reutilizando a medida que el contenido va adquiriendo el formato para la navegación por la web.

Pero existen normas, reglas y formas de hacer más fácil el diseño en la web. Primero debemos saber que para hacer un buen diseño debemos dividir el proceso en pequeños pedazos independientes que sean los detalles importantes dentro de nuestros requerimientos. Christopher Alexander fue el que promovió esta idea y a esto le llama «pedazos de patrones».

Los siguientes pasos son de mucha importancia para la realizar fácilmente el diseño y comenzamos por:

1. Hacer una lista de componentes.

Esto significa sacar cualquier parte que compone la página y que la hacen funcionar. Es decir que pueda llevar algún diseño como que información debe proporcionar la página, y acciones que debe llevar.

2. Definir prioridades.

Definan cuáles son los grupos o pedazos más importantes desde el punto de vista funcional, que sean importantes para el usuarios. Sin reordenar la lista de los componentes solo con agruparlos basta.

Ejemplo:

Importantes: A, B, C, D

Necesarios: E, F, G

Detalles: H

3. Diseñar cada pedazo.

Ya hemos agrupados todos los pedazos y separados en categorías a partir de su importancia pueden pasara diseñar sin ningún problema comenzando por los que más importancia tienen y así sucesivamente. El diseño puede ser en una herramienta de diseño o hasta en un papel. Recuerden que siempre comenzamos por el más importante, así que aquí comenzaríamos por diseñar el Diagrama A.

4. Unir los pedazos.

Cuando tengan todos los diagramas comenzamos a unirlos y a ubicarlos por como quedarían en la web y por la prioridad que tienen los pedazos. Si una web lo primero que tiene es el nombre del sitio entonces lo primero que ubicaremos será el nombre. Pueden hacer un borrador del diseño en un papel y luego lo harán como quedaría unir todos los diagramas en un prototipo HTML, o sea de cómo se vería en un documento HTML.

5. Producción.

Cuando tengan el resultado final pueden darse cuenta de que algunas cosas no funcionan tan bien en la pantalla como lo hacían en el papel o en lo que lo hayan diseñado. Y lo podrán cambiar gracias a que lo diseñamos independientes.

1.7.4.2 Patrón de diseño MVC (Model View Controller) o modelo vista controlador.

Uno de los patrones más usado para aplicaciones Web es el patrón de diseño MVC (Model View Controller) o modelo vista controlador.

Este es un patrón de diseño aportado originariamente por el lenguaje SmallTalk a la Ingeniería del Software. Se basa principalmente en dividir las aplicaciones en tres partes:

1. Controlador.
2. Modelo.
3. Vistas.

El **controlador** es el encargado de redirigir o asignar una aplicación (un modelo) a cada petición; el controlador debe poseer de algún modo, un «mapa» de correspondencias entre peticiones y respuestas (aplicaciones o modelo) que se les asignan.

El **modelo** sería la aplicación que responde a una petición, es la lógica de negocio a fin de cuentas.

Una vez realizadas las operaciones necesarias el flujo vuelve al controlador y este devuelve los resultados a una **vista** asignada.

Existe gran diferencia entre los modelos convencionales y el MVC.

Modelo Convencional.



FIG. 5 ESTRUCTURA DE UN MODELO CONVENCIONAL.

En el modelo convencional tenemos una entrada o parámetros que llegan (INPUT) se procesan y se obtiene un resultado (OUTPUT).

Modelo Vista Controlador (MVC).

El procesamiento se lleva a cabo entre sus tres componentes. El controller recibe una orden y decide quien va a llevar a cabo el modelo. Ya q una vez que el modelo (lógica del negocio) termina sus operaciones devuelve el flujo y vuelve al controller y este envía el resultado a la capa de presentación. [Ver Anexo 1, Fig. 30]

El controller de cierta forma debe tener un registro de la relación de las órdenes que le pueden llegar y la lógica del negocio que le corresponde.

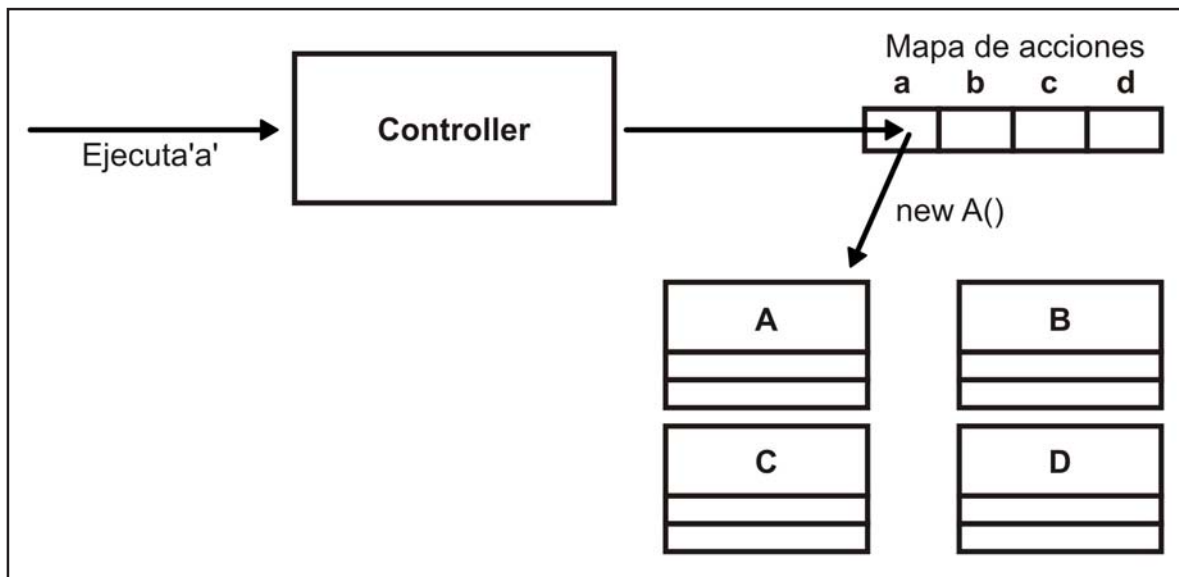


FIG. 6 FUNCIONAMIENTO DEL CONTROLLER.

1.7.4.2.1 Ventaja del MVC.

Este modelo tiene una gran ventaja porque logra una total separación entre la lógica del negocio y la presentación. Se le pueden aplicar opciones como el multilinguaje, distintos diseños de presentación, sin alterar la lógica del negocio.

Esta separación entre la capa de presentación, lógica del negocio, acceso a datos, es muy importante para el desarrollo de arquitecturas consistentes, reutilizables y más fácilmente mantenibles, lo que ahorra un gran tiempo en el desarrollo de proyectos posteriores.

Ejemplos de algunas aplicaciones que aplican este modelo tenemos a los STRUTS, framework que aplica este modelo con dirección en <http://jakarta.apache.org/struts>.

1.7.5 Patrones de Diseño de Software.

Un patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada un millón de veces sin hacerlo siquiera dos veces de la misma forma. [26]

Ahora un patrón de diseño de software es una determinada forma de crear software o de fabricar el software.

Una de las mejores bibliografías del tema lo trata el libro (Design Patterns: Elements of Reusable Object-Oriented Software. En español (Patrones de diseño: Elementos reusables en software orientado a objetos) que contiene varios modelos de diseño de software y cuyos autores son Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, conocidos como «The Gang of Four» (la banda de los cuatro).

Existen dos formas de clasificar los patrones de diseño de software:

1. Según su propósito se clasifican en:
 - a. Patrones de creación: Tratan la creación de instancias.
 - b. Patrones estructurales: Tratan la relación entre las clases, la combinación de clases y la formación de estructuras de mayor complejidad.
 - c. Patrones de comportamiento: Tratan la interacción y cooperación entre clases.
2. Según su ámbito:
 - a. De clase: Basados en Herencia de clases.
 - b. De objetos: Basados en la utilización dinámica de objetos.

Existen dos tipos fundamentales de patrones de diseño de software.

1. Patrones de diseño de software. (GRAPS).
2. Patrones de diseño de software. (GoF).

1.7.5.1 Patrones de diseño de software. (GRAPS).

Los patrones de asignación de responsabilidades (GRAPS, General Responsibility Assignment Software Patterns), describen los principios fundamentales de la asignación de responsabilidades a objetos. [27]

Constituyen el fundamento de cómo se va a diseñar el sistema finalmente. Es importante que el diseñador de software domine y aplique estos conocimientos durante la realización de un diagrama de interacción.

Los principales patrones de asignación de responsabilidades son:

Experto.

Problema: No existe un principio a seguir en la asignación de responsabilidades en el DOO (diseño orientado a objetos).

Solución: Asignar una responsabilidad al experto en información: es decir a la clase que cuenta con la información necesaria para cumplir la responsabilidad.

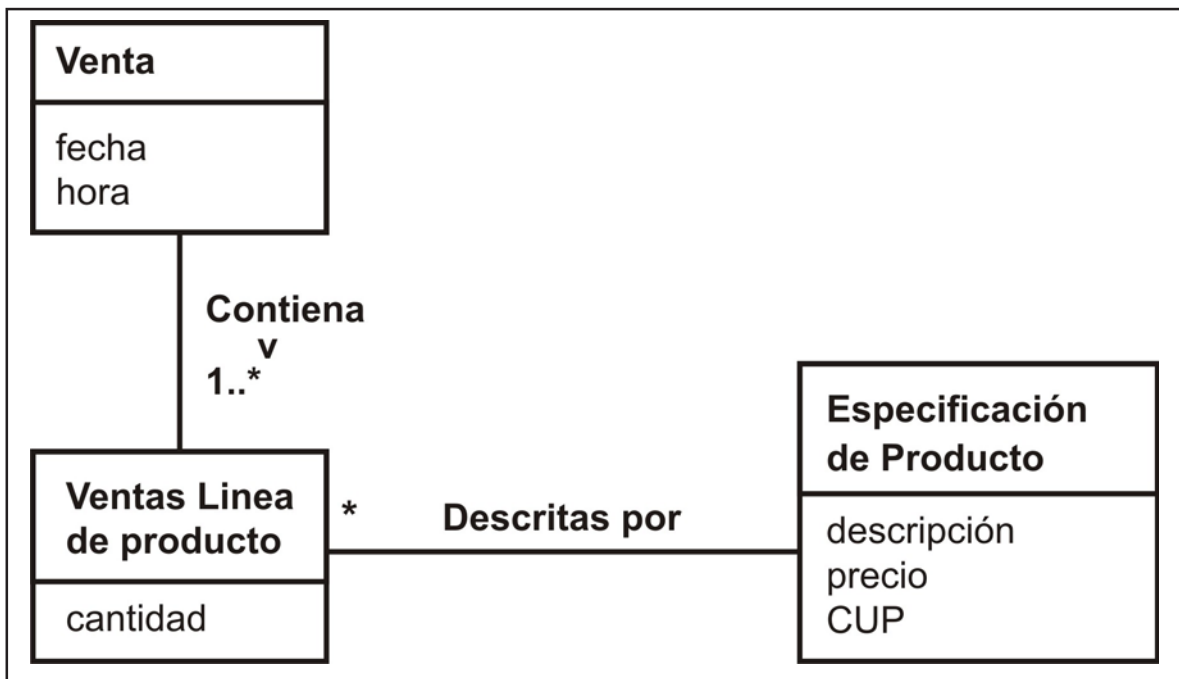


FIG. 7 ESTRUCTURA DE CLASES.

Aquí no sabemos quien es el responsable de conocer el gran total de las ventas.

Por eso asignamos a Venta la responsabilidad o lo que es lo mismo un nuevo método.

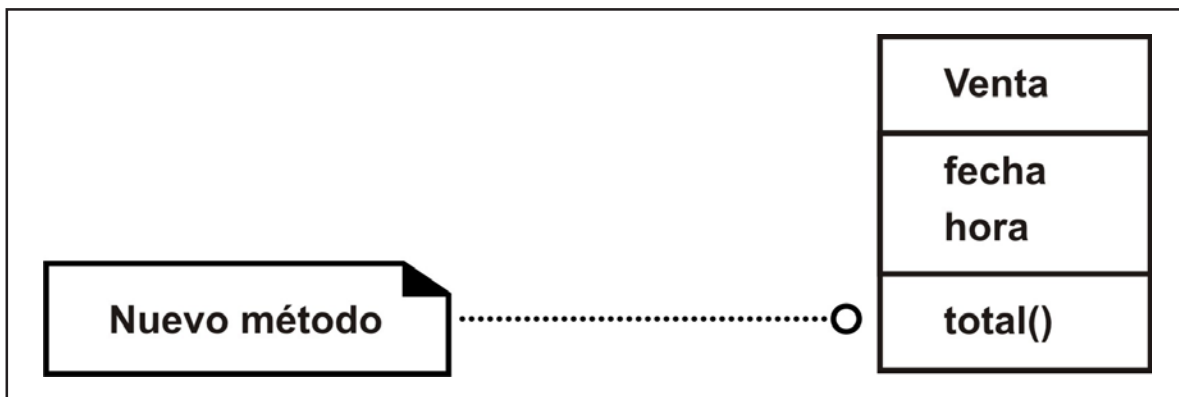


FIG. 8 ESTRUCTURA DE LA CLASE VENTA CON UN NUEVO MÉTODO.

Beneficios: Conserva el encapsulamiento ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece tener sistemas más robustos y de fácil mantenimiento.

Este comportamiento se distribuye entre las clases que cuentan con la información requerida, posibilitando así definiciones de clases «sencillas» y más **cohesivas** que son más fáciles de comprender y de mantener. Así se brinda soporte a una alta cohesión.

Creador

Problema: Aquí se busca quien es el responsable de crear una nueva instancia de otras clases.

Solución: se asigna la responsabilidad de la clase B cree un objeto de la clase A solo cuando:

- B agrega los objetos A.
B contiene los objetos A.
- B registra las instancias de los objetos A.
- B utiliza específicamente los objetos A.
- B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado (así que B es un Experto respecto a la creación de A).

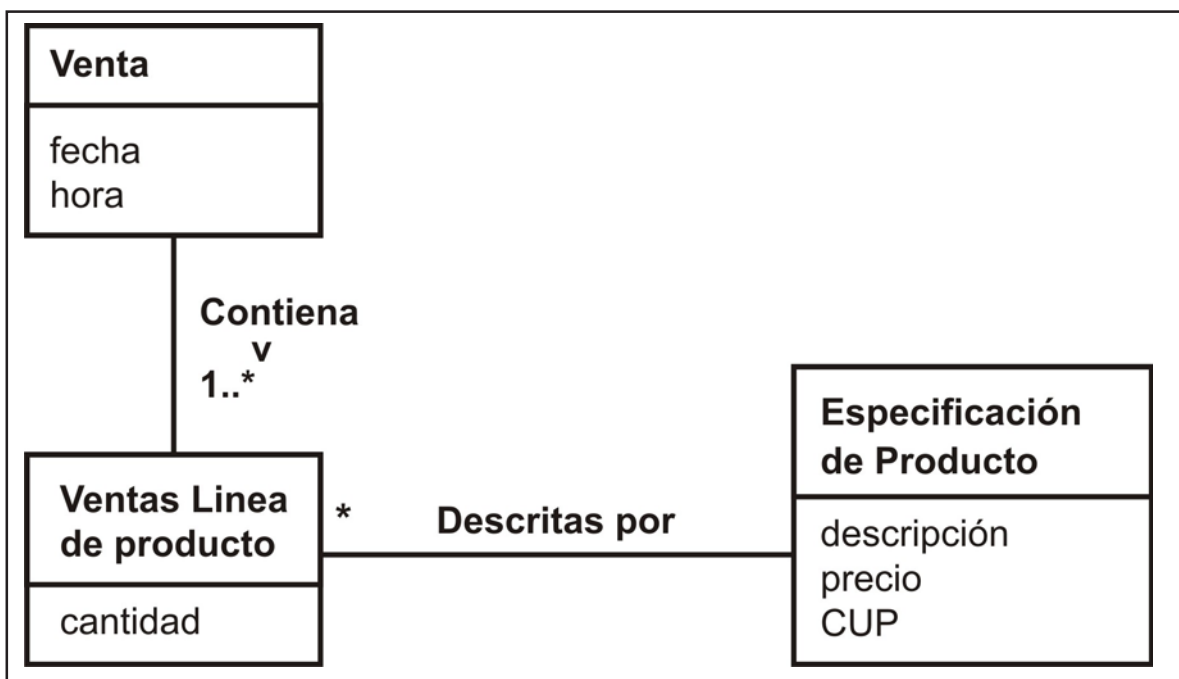


FIG. 9 ESTRUCTURA DE CLASES.

La pregunta es quien debe crear *VentasLineadeProducto*.

Una *Venta* contiene (en realidad, agrega) muchos objetos *VentasLineadeProducto*; por ello, el patrón Creador sugiere que *Venta* es idónea para asumir la responsabilidad de crear las instancias *VentasLineadeProducto*.

Esto brinda soporte a un bajo acoplamiento, lo cual pone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización. Es probable que el acoplamiento no aumente, pues la clase *creada* tiende a ser visible a la clase *creador*, debido a las asociaciones actuales que nos llevaron a elegirla como el parámetro adecuado.

Bajo Acoplamiento.

Pregunta: Como dar soporte a una dependencia escasa y un aumento de reutilización.

Por ejemplo: Los cambios de las clases afines ocasionan cambios locales, es más difícil de entender, es más difícil de reutilizar porque se requiere la presencia de otras clases de las que depende y herencias muy profundas

Solución: Asignar una responsabilidad para mantener pocas dependencias entre las clases.

Beneficios: No se afectan por cambios de otros componentes. Fáciles de entender por separado. Fáciles de reutilizar.

Alta cohesión.

Problema: Mantener la complejidad dentro de límites manejables.

Solución: Cada elemento debe realizar una labor única dentro del sistema, que no desempeñan ninguna otra.

Ejemplos:

De baja cohesión: clases que realizan muchas cosas.

De un buen diseño: agrupar las clases por paquetes de servicio o funcionalidades (bien por su uso o por su herencia).

Controlador.

Problema: Atender un evento del sistema.

Solución: Asignarlas a clases específicas.

Beneficios: Facilita centralizar las actividades (validación, seguridad y otras).

El controlador no realiza estas actividades sino que las delega a otras clases con las que mantiene un modelo de alta cohesión.

El controlador representa el sistema global ya que cuando una tarea específica podemos crear una clase que se llame así.

También representa la organización global.

No hables con extraños.

Problema: A quien debe invocar un método.

Solución: Solo invocara a métodos de si mismo (this). De su área de parámetros. Un objeto creado en su propio ámbito.

Ejemplo: En las aplicaciones Web es crear variables globales (estáticas) o el acceso desde muchos puntos desordenadamente a variables de sesión o aplicación (contexto).

1.7.5.2 Patrones de diseño de software GOF.

1.7.5.2.1 Tipos de Patrones de diseño de software.GOF

1. Patrones creacionales o de creación:

Estos patrones de software se centran en la forma de crear las clases y sus instancias, así como el uso que recibirán una vez creadas. Ejemplos de patrones de este tipo son el patrón Singleton o el patrón Abstract Factory, Builder, Factory Method, Prototype. [28]

Tipos de patrones creacionales o de creación.[29]

Singleton: asegura que una clase tendrá solo una instancia, y provee un mundo global de acceso a la misma. (Ver Fig. 11).

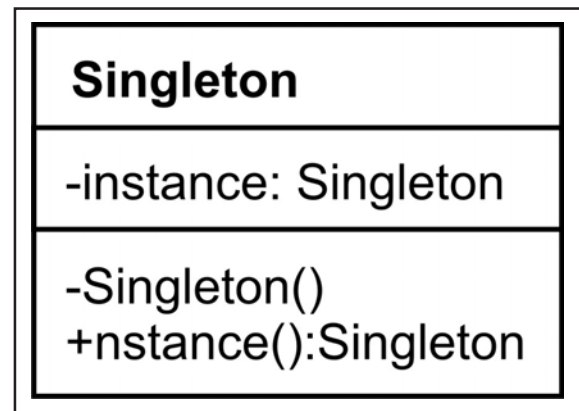


FIG. 10 ESTRUCTURA GENERAL DEL PATRÓN

Observen que el constructor es privado.

El método instance() es el que devuelve la única instancia de esta clase, o la crea si no existe.

Abstract Factory: provee un interfaz para crear familias de objetos dependientes o relacionados sin necesidad de especificar sus clases concretas. [Ver Anexo 1, Fig. 31]

La clase cliente («Client»), selecciona la clase factoría concreta en el momento de la creación.

Sólo guarda una referencia a la clase base abstracta de ProductA y ProductB.

De esta forma el Cliente y los productos son totalmente independientes.

Builder: separa la construcción de un objeto complejo de su representación, de tal manera que el mismo proceso de construcción puede crear diferentes representaciones (por ejemplo: conversores de formatos de texto).

Factory Method: define una interfaz para crear un objeto, pero deja que las subclases decidan qué clase instanciar. Este patrón delega la instanciación de una clase a sus subclases (como un constructor virtual). [21]

Prototype: especifica las clases de objetos a crear usando un ejemplar prototípico, y crea nuevos objetos copiando este prototipo.

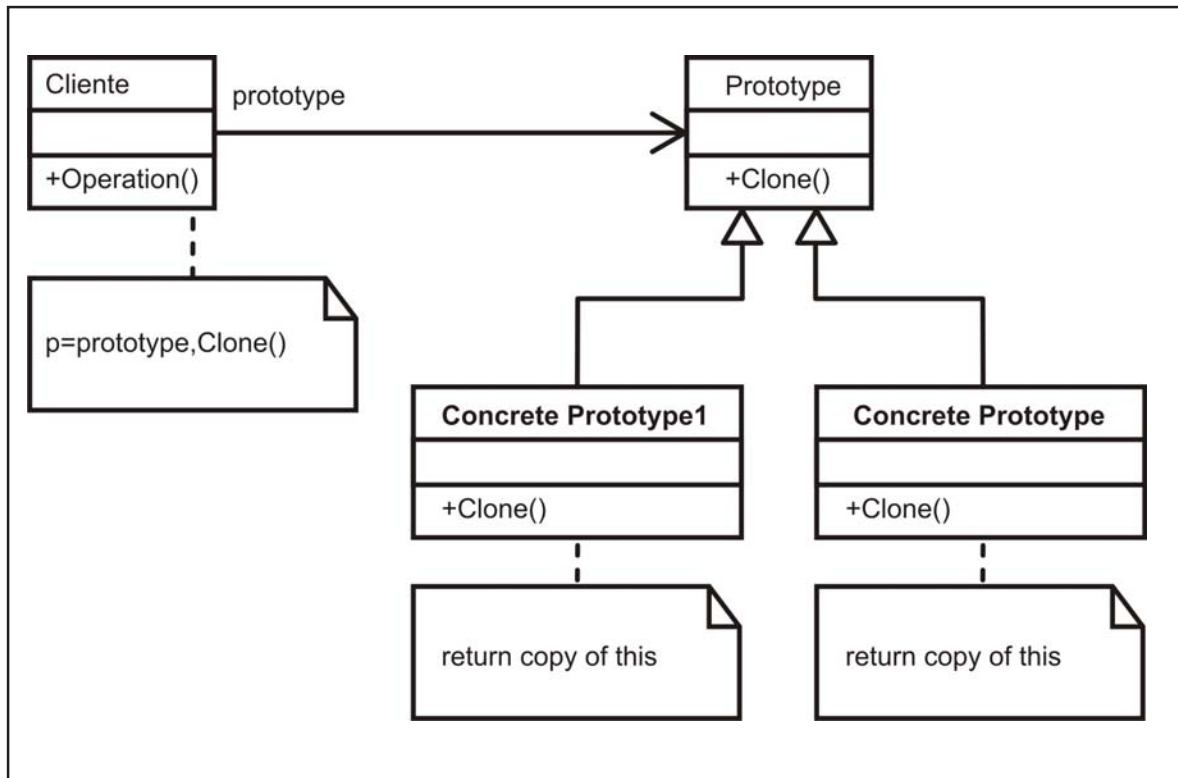


FIG. 11 ESTRUCTURA GENERAL DEL PATRÓN DE DISEÑO PROTOTYPE.

El cliente («Client»), en lugar de crear objetos, clona otros ya creados.

Nótese que `clone()` debe ser un método polimórfico, por lo que no será necesario conocer exactamente al objeto.

2. Patrones Estructurales: Se centran en la composición y estructura de las clases y la herencia entre ellas. Por ejemplo, el patrón Adapter o el patrón Facade, Bridge, Composite, Decorador, Flyweight, Proxy. [30]

Tipos de patrones estructurales.[31]

Adapter: Encapsula una clase de manera que es posible normalizar su interfaz para su uso con otras clases. El problema aparece cuando en una aplicación reaparece una clase que no tiene la interfaz requerida.

La clase adaptadora se interpone entre el cliente y la clase reutilizada. La clase adaptadora presenta la interfaz que espera la aplicación, y se comunica con la clase reutilizada, «traduciendo» los mensajes.

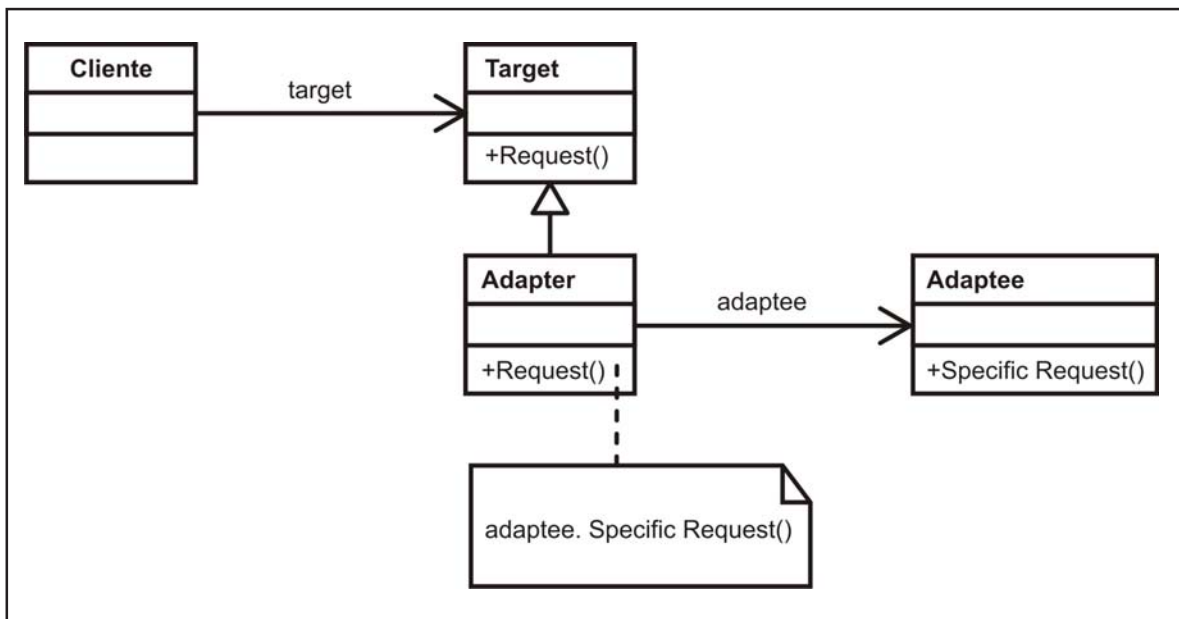


FIG. 12 ESTRUCTURA GENERAL DEL PATRÓN DE DISEÑO ADAPTER.

El cliente («Client»), se comunica con la clase adaptadora que actúa de intermediaria, en lugar de con la clase que realmente proporciona la funcionalidad («Adaptee»).

Bridge: Permite separar dos aspectos distintos de una jerarquía de clases. Cuando una jerarquía implementa una la funcionalidad real, las otras representan diversos aspectos, como representación en diferentes medios.

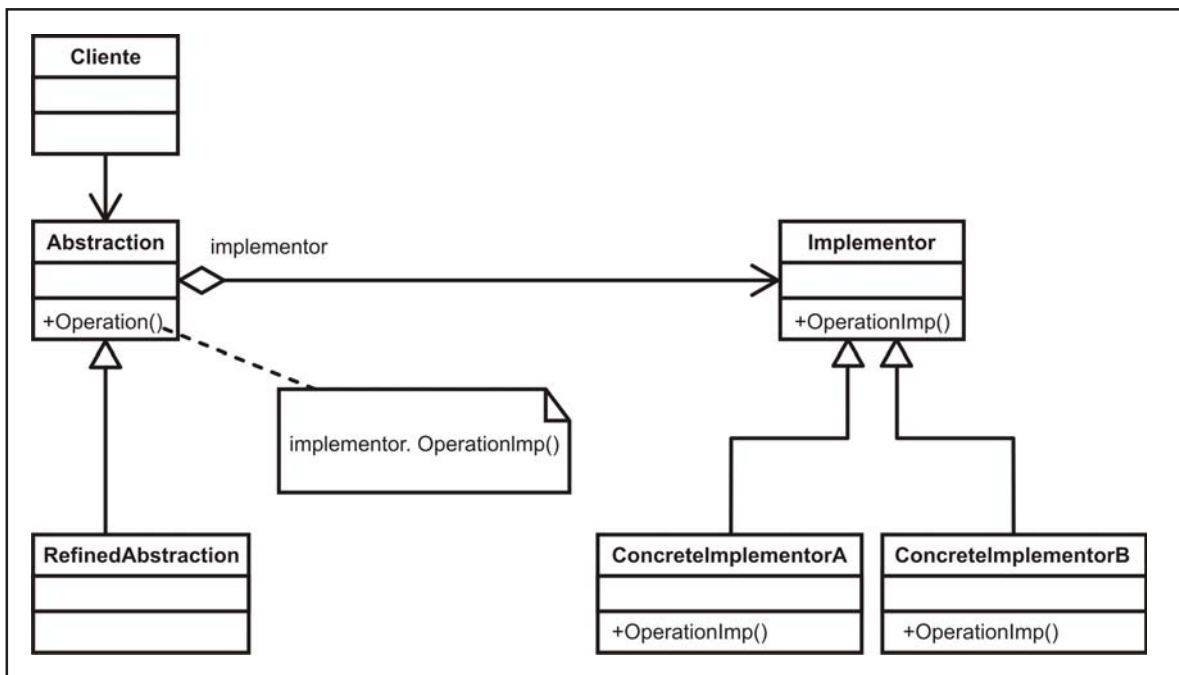


FIG. 13 ESTRUCTURA GENERAL DEL PATRÓN DE DISEÑO BRIDGE.

Existen varias implementaciones distintas para un método dado (ConcreteImplementorA y ConcreteImplementorB). Se trata de separar diferentes aspectos.

Facade:

Típicamente, un diseño adecuado nos llevará a una factorización en clases más allá de la estrictamente necesaria para un sistema determinado.

En ese caso, es posible construir una clase facade, (fachada), que tenga la interfaz esperada y sea el que realmente se comunique con la estructura de clases real, cuyas interfaces pueden variar.

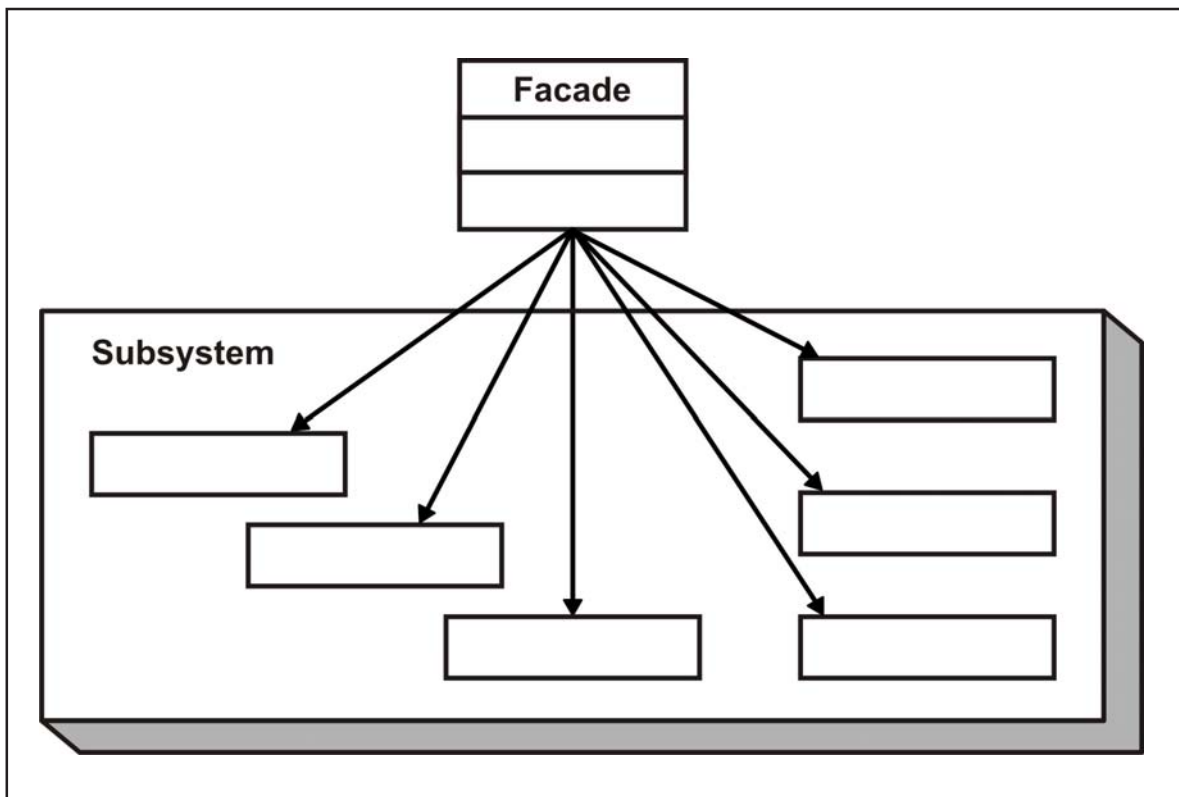


FIG. 14 ESTRUCTURA DEL PATRÓN DE DISEÑO FACADE.

La clase fachada se interpone entre el cliente (no mostrado) y el subsistema que proporciona la funcionalidad deseada. Facade simplifica la interfaz del subsistema.

3. Patrones de comportamiento[32]: Los que se basan en la comunicación entre las distintas clases y objetos. Por ejemplo, patrón Observer o el patrón Strategy, Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Template Method, Visitor.[30]

Chain of responsibility:

En múltiples ocasiones, un cliente necesita que se realice una función, pero o no conoce al servidor concreto de esa función o es conveniente que no lo conozca para evitar un gran acoplamiento entre cliente y servidor. La petición se lanza a una cadena de objetos que se «la van pasando» hasta que uno de los objetos la maneja. [21]

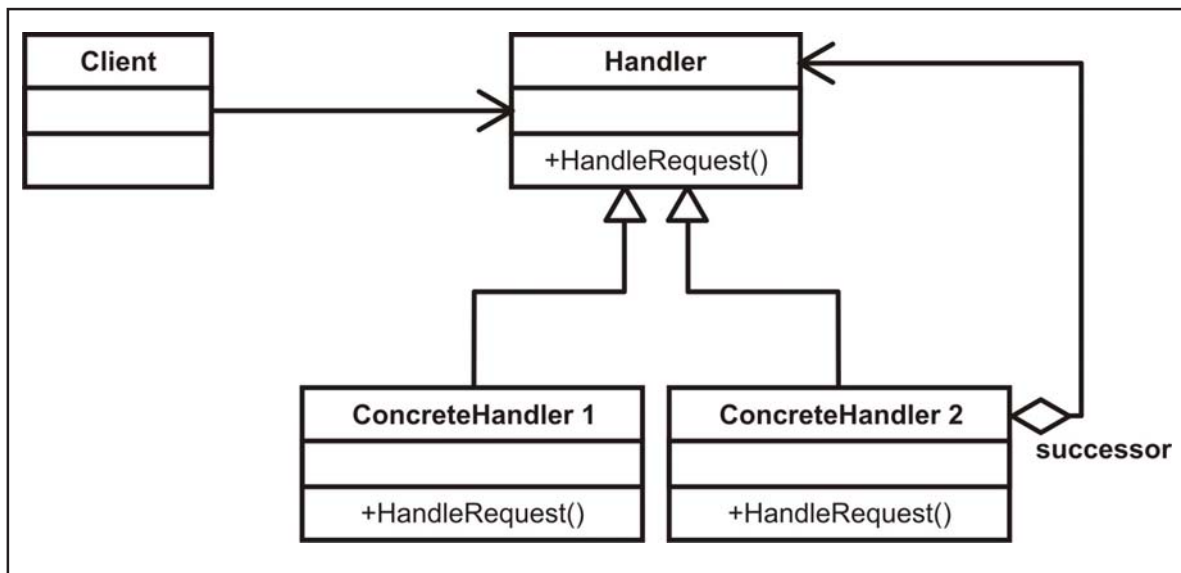


FIG. 15 ESTRUCTURA DEL PATRÓN DE DISEÑO CHAIN OF RESPONSABILITY.

El cliente («Client»), conoce a un gestor que es el que lanza la petición a la cadena hasta que alguien la recoge.

Iterator:

Es un patrón ampliamente utilizado. Se trata de tener varias colecciones de objetos (contenedores como lista, vector), que pueden ser recorridas utilizando la misma abstracción: el iterador. Permite recorrer cada elemento de un contenedor secuencialmente. Es ampliamente utilizado en la STL de C++.

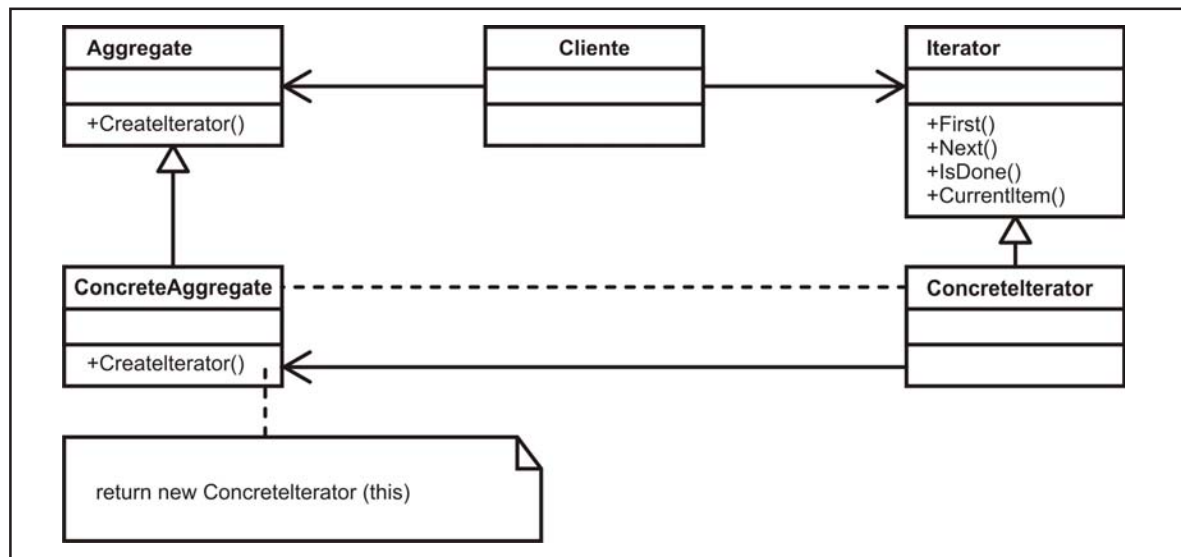


FIG. 16 ESTRUCTURA DEL PATRÓN DE DISEÑO ITERATOR.

El cliente («Client»), puede incluso utilizar un iterador sin saber sobre qué colección concreta se está ejecutando.

Observer:

Se trata de poder notificar de un cambio a varios objetos. Al fin y al cabo, es una manera de establecer una relación de uno a muchos.

Un ejemplo claro puede encontrarse cuando, teniendo abiertas dos ventanas de exploración sobre un mismo directorio, desde una de ellas borramos un archivo, y la segunda se actualiza automáticamente. Podrían definirse ambos exploradores como dos observadores del sistema de ficheros. [Ver Anexo 1, Fig. 32]. En cada observador puede tomarse una acción distinta según se haya producido un cambio.

En el capítulo se realizó un amplio estudio de todos los temas teóricos que soportan la investigación científica. Se realizó un resumen a grandes rasgos de las principales características y procesos presentes en una Intranet. Se fundamentó la selección de las tecnologías, metodologías y herramientas utilizadas en el desarrollo del sistema. Además se trataron los conceptos de análisis y diseño de sistemas, también se planteó las tendencias del diseño y los diferentes patrones del diseño.

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

En el siguiente capítulo se aborda las características que tendrá el sistema, para cumplir con ellas se realizó un estudio de todos los servicios y funcionalidades que presenta el Portal de la Intranet de la Universidad de las Ciencias Informáticas, sus objetivos y procesos que lo soportarán. Se realizará la descripción de los procesos a automatizar, así como un análisis crítico de cómo se ejecutan estos procesos actualmente y las causas que dieron origen a la Situación problemática.

2.1 Objeto de automatización.

Los principales procesos que serán objeto de automatización son:

La publicación de noticias, avisos, efemérides, sitios web y páginas informativas. Los cuales serán aprobados previamente a su publicación.

La gestión de usuario que recogerá toda la información del usuario registrado en el portal. Su estado de cuenta, salario laboral, información correspondiente al área que pertenece, dirección y datos personales serán algunos de los campos que se podrán encontrar.

También se realizará un buscador de toda la información existente dentro del Portal, en él se verán reflejados los intereses de todos los usuarios y soportará todos los servicios y sus procesos fundamentales estarán orientados a describir, organizar e informar a la comunidad universitaria.

2.2 Información que se manejará en el Portal de la Intranet 2.

En el Portal/Intranet se manejará todas las informaciones de todas las actividades y vida diaria de la Universidad tales como:

- Información de todas las facultades de la UCI.
- Noticias de la universidad y de actualidad nacional e internacional.
- Publicaciones de avisos.
- Publicaciones de banner.
- Publicación de efemérides.
- Información personal de los usuarios de la Intranet 2.
- Estadísticas del portal.
- Información de la organización, historia, reglamentos y directivas.
- Producción de software.
- Vida y funcionamiento interno.

- Sobre las organizaciones de masas de la UCI, UJC, PCC, FEU, CTC.
- Servicios que se prestan en la UCI, transporte, gastronomía, comercio, Serv. Técnicos, Serv. de apoyo a la residencia, Serv. de seguridad y protección.
- Buscadores de personas, de aplicaciones, telefónicos y de áreas.
- Guía de estilos, pautas de diseño, normas editoriales.
- Mapa del Portal.
- Sistema de opiniones.

2.3 Propuesta de sistema.

El sistema en primer lugar es una Intranet que debe funcionar para los 10 mil estudiantes que existen en la UCI y que viven bastante tiempo en la universidad y tienen un amplio dominio de la tecnología. Los trabajadores podrán acceder siempre y cuando tengan acceso al dominio UCI.

Por tanto nuestro sistema debe ser antes de todo accesible para todos los usuarios que navegan en ella, pero siempre atendiendo que el portal deberá funcionar para usuarios que tienen una interacción continua con la tecnología y como será implementado para una universidad de informática entonces solo llevará estándares, que le faciliten al usuario acceso y usabilidad del portal, algunos de los estándares a utilizar serán:

1. Proporcionar alternativas equivalentes para los contenidos auditivo y visual.
2. Garantizar que las páginas utilicen nuevas tecnologías
3. Asegurar que el usuario controla los cambios del contenido sensibles al tiempo.
4. Garantizar la accesibilidad directa de las interfaces
5. Proporcionar información de contexto y de orientación.
6. Asegurar que los documentos sean claros y sencillos.

El objetivo es crear un portal que trabaje mejor para todos, con sitios accesibles a más personas y que funcionen desde cualquier área de la universidad. Basa su funcionamiento fundamentalmente orientado a los servicios que brindará hacia la comunidad, entre estos encontramos el servicio de editar todo lo relacionado con la información de las facultades y su quehacer diario tanto en la docencia como en las demás actividades relacionada con la misma. También estará centrada en la realización de los siguientes procesos: publicar noticias, avisos, efemérides, sitios Web, páginas informativas, buscar información contenida en el Portal, así como mostrar una estadística del portal orientada a la cantidad de usuarios visitantes, sitios más visitados, servicios más accedidos y cualquier otra información que necesite ser mostrada numéricamente en el portal.

También existe la posibilidad de adicionar, actualizar y aprobar los diferentes contenidos agrupados por secciones como son: noticias, avisos y efemérides.

Las noticias presentan diferentes categorías entre ellas tenemos las de ciencia y técnica, culturales, deportivas, internacionales, nacionales, productivas y universitarias. Estas serán adicionadas por el productor quien se encargará de ponerles la fecha de publicación y de terminación de la misma. A las noticias se le deben poder adjuntar diferentes materiales multimedia (imágenes, videos, sonidos).

Al igual que las noticias los avisos presentan categorías como son: de trabajadores, de estudiantes y relevantes. Ellos serán emitidos a partir de su aprobación.

Cada día se publicará las efemérides correspondientes en textos breves. Se podrán recuperar a partir de la fecha de publicación, palabras del título, así como la navegación cronológica de todas las efemérides o solamente de un grupo específico.

Otros de los procesos que se describen son la publicación de Sitios Web, a partir de su solicitud y la publicación de Páginas Informativas que recogerá toda la información referente a las diferentes áreas, así como la información perteneciente al personal de la universidad.

La Intranet tendrá acceso a muchos de los servicios que brinde la universidad como correo, paging, inter-nos y otros, a partir de la dirección url y un link se accederán a ellos.

Se podrán realizar búsquedas de cualquier información contenida en la Intranet.

Todos los contenidos publicados en el Portal serán aprobados a partir del cumplimiento de las normas y manual de estilo.

En nuestro país existen muchas Intranets, pero no todas están orientadas a la gestión del conocimiento que cuente con los servicios necesarios que responden a las necesidades primarias de la organización. Bajo este precepto es que se desea la realización del Portal de la Universidad de las Ciencias Informáticas.

2.4 Modelo del Dominio.

El proceso de modelar permite obtener una visión de la organización, permitiendo definir los procesos y roles relacionados con la obtención de requerimientos y del análisis-diseño. Si se determina que no es necesario un modelo completo del negocio se realizará lo que se conoce como un modelo del dominio.

Un modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema. [33]

Debido al bajo nivel de estructuración que presenta el negocio que se está estudiando y que está altamente centrado en las tecnologías informáticas, se propone un modelo del dominio ayudando a los usuarios, clientes, desarrolladores y demás interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema.

El modelo del dominio se describe mediante diagramas UML, específicamente con un diagrama de clases conceptuales significativas en el dominio del problema. Este va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.

Se identifican como conceptos utilizados en el diagrama a los siguientes términos del glosario:

- **Intranet:** Una Intranet es una red privada que la tecnología Internet usó como arquitectura elemental y que utiliza los protocolos TCP/IP para su transporte básico. [34]
- **Servicios:** Proporcionar lo que realmente requiere el cliente. [35] Es el conjunto de actividades interrelacionadas que se le ofrecen al usuario para que este obtenga las funcionalidades que le sean de interés y ayuda para desarrollarse en la comunidad universitaria.
- **Áreas:** Diferentes espacios y departamentos dentro de la infraestructura de la universidad que se agrupan para formar las diferentes áreas, por ejemplo alimentos, residencia, entre otras.
- **Sitio Web:** Conjunto de páginas relacionadas por hipervínculos y con un tema en común. Están presentes los diferentes sitios Web de la Universidad.
- **Página Informativa:** Publicación de información descriptiva de un determinado subsistema de la universidad (estructural o funcional).
- **Categoría:** Los diferentes tipos de noticias y avisos que podemos encontrar, las categorías que encontramos en las noticias son: Ciencia y técnica, culturales, deportivas, Internacionales, nacionales, productivas y universitarias. En el caso de los avisos serán: De trabajadores, de estudiantes y relevantes.
- **Efemérides:** Diferentes acontecimientos ocurridos en el día en curso, recogidos en un texto breve.
- **Noticias:** Eventos, acontecimientos y actividades recogidas en textos y categorizadas en dependencia del contexto de la noticia.
- **Avisos:** Textos breves con avisos dirigidos a los estudiantes, trabajadores y pueden ser relevantes.
- **Productor:** Actor encargado de la adición de contenido en el portal. Rol desempeñado por un periodista o Resp. de Área.
- **Editor:** Actor encargado de la edición, corrección y arreglos realizados a los diferentes contenidos que serán publicados en el portal.
- **Revisor:** Encargado de la revisión y de la puesta en función por el editor de las normas y estilos del manual de publicación de la universidad.
- **Publicador:** Rol desempeñado por un actor encargado de publicar y administrar el portal.

2.4.1 Diagrama Conceptual Modelo del dominio.

Para facilitar la comprensión de esta tesis se incluyó el modelo de dominio en los Anexos. [Ver Anexo 2]

2.5 Especificación de los requisitos de software.

El flujo de especificación de los requisitos de software radica su mayor esfuerzo en el reconocimiento del problema como lo ve el usuario. Se define una evaluación del problema y se dan respuesta a las funcionalidades que tendrá el sistema. Con él se pretende entender el comportamiento del software y se establecen las características de la interfaz y el descubrimiento de restricciones adicionales de diseño.[36]

Reúne todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema. Los requisitos se pueden clasificar en: funcionales y no funcionales.

2.5.1 Requerimientos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, no alteran la funcionalidad del producto y se mantienen invariables sin importarle con que propiedades o cualidades se relacionen. Son el punto de partida para identificar qué debe hacer el sistema. [37]

A continuación se relacionan los que deben cumplir el portal de la Intranet 2:

R1. Autenticar usuario.

- El sistema debe ser capaz de autenticar los usuarios por el dominio.
- El sistema debe ser capaz de identificar los grupos de usuarios en dependencia al rol que desempeñan

Esta información requiere:

- a-) Nombre de usuario.
- b-) Contraseña.

R2. Adicionar noticias.

- El sistema debe ser capaz de adicionar las noticias por categorías, confeccionadas por los periodistas o encontradas en los diferentes medios de comunicación.

Esta información requiere:

- a-) Categoría.
- b-) Título.
- c-) Resumen.
- d-) Texto de la noticia.
- e-) Fotos.
- f-) Fuentes.
- g-) Archivos.

h-) Fecha de publicación.

i-) Fecha de retirada.

j-) Autor.

R3. Actualizar noticias.

- El sistema debe ser capaz de actualizar las noticias diariamente en dependencia de las diferentes categorías.

Esta información requiere

a-) Categoría.

b-) Título.

c-) Resumen.

d-) Texto de la noticia.

e-) Fotos.

f-) Fuentes.

g-) Archivos.

h-) Fecha de publicación.

i-) Fecha de retirada.

j-) Autor.

R4. Aprobar noticias.

- El sistema debe ser capaz de decidir si la noticia cumple con los requisitos para ser publicada.

a-) Categoría.

b-) Autor o fuente.

c-) Tema.

d-) Palabras en el título.

e-) Fecha de publicación.

f-) Fecha de caducidad.

g-) Palabras claves.

R6. Adicionar Avisos.

- El sistema debe ser capaz de adicionar avisos creados por los responsables de áreas que pueden ser de interés para ciertos usuarios o para todos en general.

- El sistema debe ser capaz de identificar los avisos según sus categorías.
- Pueden estar enmarcados de importancia o de prioridad.

Esta información requiere:

- a-) Autor.
- b-) Nivel de prioridad.
- c-) Categorías.
- d-) Fechas de publicación.
- e-) Categoría.
- f-) Título.
- g-) Texto.
- h-) Fecha de retirada.
- i-) Solicitud de área.

R7. Actualizar avisos.

- El sistema debe ser capaz de actualizar los avisos según vayan caducando en la fecha de vigencia.
- Debe ser capaz de identificar los avisos por categorías y nivel de prioridad.

Esta información requiere:

- a-) Contenido nuevo.
- b-) Categoría.
- c-) Nivel de prioridad.
- f-) Título.
- g-) Texto.
- h-) Fecha de publicación.
- i-) Fecha de retirada.
- j-) Autor.
- k-) Solicitud de área.

R8. Aprobar avisos.

- El sistema debe ser capaz de aprobar los avisos que serán publicados a partir de que cumplan con los requisitos para la publicación.

Esta información requiere:

- a-)** Autor.
- b-)** Título.
- c-)** Nivel de prioridad.
- d-)** Categorías.
- e-)** Fechas de publicación.
- f-)** Fecha de retirada.
- g-)** Solicitud del área.

R10. Adicionar efemérides.

- El sistema debe ser capaz de adicionar efemérides de acuerdo al día en curso.

Esta información requiere:

- a-)** Texto.
- b-)** Fecha.
- c-)** Título.
- e-)** Aprobación.
- f-)** Fecha de publicación.
- g-)** Archivos.

R11. Actualizar efemérides.

- El sistema debe ser capaz de actualizar diariamente las efemérides en correspondencia a la fecha.

Esta información requiere:

- a-)** Texto.
- b-)** Fecha.
- c-)** Título.
- e-)** Aprobación.
- f-)** Fecha de publicación.
- g-)** Archivos.

R12. Aprobar efemérides.

- El sistema debe ser capaz de aprobar las efemérides que serán publicadas diariamente.

Esta información requiere:

- a-)** Texto.
- b-)** Fecha.
- c-)** Título.
- e-)** Aprobación.
- f-)** Fecha de publicación.
- g-)** Archivos.

R14. Enviar solicitud de publicación SW.

- El sistema debe ser capaz enviar las solicitudes confeccionadas por los diferentes Resp. de aéreas.

Esta información requiere:

- a-)** Descripción del sitio.
- b-)** Nombre del solicitante.
- c-)** Clasificación del sitio.

R15. Aprobar solicitud de publicación SW.

- El sistema debe ser capaz de aprobar los sitios web que se publicarán.

Esta información requiere:

- a-)** Descripción del sitio.
- b-)** Clasificación.

R16. Publicar Sitios Web.

- El sistema debe ser capaz de publicar los sitios web de las diferentes aéreas de la universidad de acuerdo con la aprobación del consejo editor.

Esta información requiere:

- a-)** Descripción del sitio.
- b-)** Clasificación.

R17. Enviar solicitud de publicación de páginas Informativas.

- El sistema debe ser capaz de recibir las solicitudes enviadas por los diferentes responsables de aéreas para publicar las páginas informativas, estas recogen la información de determinado usuario, área o servicio perteneciente a la universidad.

Esta información requiere:

- a-)** Título.

b-) Fuente (Área de la universidad).

c-) Resumen del contenido de la página (Texto).

R18. Aprobar solicitud de publicación de páginas Informativas.

- El sistema debe ser capaz de enviar la aprobación de la solicitud de publicación de páginas Informativas.

Esta información requiere:

a-) Título.

b-) Fuente (Área de la universidad).

c-) Resumen del contenido de la página (Texto).

d-) Respuesta.

R19. Publicar Páginas informativas.

- El sistema debe ser capaz de publicar las páginas informativas que han sido aprobadas.

Esta información requiere:

a-) Título.

b-) Fuente (Área de la universidad).

c-) Página a publicar.

R20. Adicionar usuario.

- El sistema debe ser capaz de adicionar usuario en dependencia del rol a desempeñar

Esta información requiere:

a-) Nombre del usuario.

b-) Fecha de creación.

c-) Área a la pertenece.

d-) Rol.

R21. Modificar usuario.

- El sistema debe ser capaz de modificar los usuarios, en dependencia del rol que desempeñe en el momento.

Esta información requiere:

a-) Nombre del usuario.

b-) Fecha de creación.

c-) Área a la pertenece.

d-) Rol.

R22. Eliminar usuario

- El sistema debe ser capaz de eliminar usuario y rol correspondiente.

Esta información requiere:

a-) Nombre del usuario.

b-) Fecha de creación.

c-) Área a la pertenece.

d-) Rol.

R23. Buscar información del contenido del portal.

- El sistema debe ser capaz de realizar una búsqueda y recopilación de cualquier información perteneciente a la universidad existente en el portal.

Esta información requiere:

a-) Información que se va a buscar.

b-) Palabras claves.

c-) Autor.

R24. Mostrar estadísticas del portal.

- El sistema debe ser capaz de mostrar las estadísticas del portal, usuario que han visitados, sitios más visitados, información más buscada y servicios más utilizado.

Esta información requiere:

a-) Usuarios.

b-) Sitios visitados.

c-) Servicios.

2.5.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Representan las características que hacen al producto atractivo, usable, rápido o confiable, son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales.

Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto. [38]

Entre las diferentes categorías de los requerimientos no funcionales se encuentran:

1. Usabilidad.

- El Portal podrá ser accedido por todos los usuarios de la UCI.
- Debe tener un diseño sencillo.
- La navegación debe ser fácil para el usuario.
- No se debe recargar de contenido que resulte molesto.

2. Portabilidad.

- El sistema será multiplataforma.

3. Seguridad.

- El sistema se regirá por el reglamento disciplinario de navegación de la universidad, así como la seguridad que ofrece PLONE.
- Debe estar protegida con la seguridad de Linux. Los usuarios que accedan a la aplicación solo tienen acceso según el rol que cumplan dentro de esta.

4. Implementación.

- Utilizar el servidor Apache para reemplazar el Zope Web Server. Para manejar las BD utilizar PostgreSQL en reemplazo de ZopeDB. Usar CMS Plone y el lenguaje que trae incorporado Python para el desarrollo de la aplicación.

5. Apariencia.

- La apariencia del sistema debe ser lo más accesible y usable posible, cumplir las normas de la UCI, en dependencia del rol que cumpla el usuario.

6. Software.

Cliente:

- Cualquier sistema operativo con interfaz gráfica y red.
- Navegador Web.

Servidor:

- Linux cualquier distribución (Debian, Ubuntu, KDE, etc.).
- Servidor de BD PostgreSQL en reemplazo de ZopeDB.
- Apache para reemplazar el Zope Web Server.

7. Hardware.

Para el desarrollo:

- Pentium IV o superior.

- 256 de memoria RAM o superior.
- Disco duro con capacidad de 80 GB o superior.

Para la explotación:

Clientes:

- Pentium IV o superior.
- 240 de RAM o superior.

8. Legales.

- El sistema contara con normas de de estilos y pautas de diseño para cada uno de los servicios de la intranet aprobadas por el consejo de dirección de la Universidad.

9. Confiabilidad.

- El sistema solo será visitado por los usuarios que cuenten con acceso al dominio UCI.

10. Interfaz.

- La interfaz del sistema será muy legible.
- También será fácil de usar.
- Debe regirse por el manual de estilo de la Universidad.

11. Ayuda y documentación en línea.

- El sistema cuenta con ayuda para navegar en el portal, una guía de estilos y pautas de diseño.

2.6 Definición de los casos de uso.

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema. Entre ellos se encuentran entre las actividades a automatizar. [39]

2.6.1 Definición de los actores.[40]

Actores del sistema

Cada trabajador del negocio (inclusive si fuera un sistema ya existente) que tiene actividades a automatizar es un candidato a actor del sistema. Si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema.

Los actores del sistema:

- No son parte de él.

- Pueden intercambiar información con él.
- Pueden ser un recipiente pasivo de información.
- Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado. [4]

TABLA 1 ACTORES DEL SISTEMA

Nombre del Actor	Descripción
Productor	Usuario que realiza la acción de crear los contenidos que se publicarán en el portal.
Editor	Usuario encargado de los arreglos, edición y de cumplir con las normas y estilos establecidos por la UCI, para la publicación de contenidos.
Revisor	Consejo editorial, compuesto por directivos y periodistas, que se encargan de decidir que contenido será publicado en el portal y de que cumpla con las normas de estilos establecidas por la UCI.
Publicador	Usuario con permisos de administración, que publica el contenido en el portal.

2.6.2 Listado de casos de uso del sistema.

TABLA 2 CU AUTENTICAR USUARIO (RESUMEN)

CU-1	Autenticar usuario
Actor	Usuario
Descripción	El caso de usos lo inicia el usuario que accederá al sistema entrando su login y contraseña.
Referencia	R1

TABLA 3 CU ADICIONAR USUARIO (RESUMEN)

CU-2	Adicionar Contenido
Actor	Productor
Descripción	El caso de uso se inicia cuando el productor confecciona el contenido que será publicado en el portal. Este consta de diferentes categorías: noticias, efemérides y avisos. El CU termina cuando el contenido es enviado al editor.
Referencia	R2,R6,R10

TABLA 4 CU ACTUALIZAR CONTENIDO (RESUMEN)

CU-3	Actualizar Contenido
Actor	Editor
Descripción	El caso de uso se inicia cuando el editor recibe el contenido a publicar, este es corregido y arreglado según el manual de estilos, y termina cuando el contenido es actualizado en el SGC y enviado al revisor para su aprobación.
Referencia	R2,R3,R8,R7,R10,R12

TABLA 5 CU APROBAR CONTENIDO (RESUMEN)

CU-4	Aprobar Contenido
Actor	Revisor
Descripción	El caso de uso se inicia cuando el revisor recibe el contenido actualizado, a partir de la categoría de este es seleccionado el revisor correspondiente. Si esta cumple con todas las normativas de publicación es aprobado, sino es enviado nuevamente al editor para aplique nuevas correcciones.
Referencia	R3,R4,R7,R8,R11,R12

TABLA 6 CU ENVIAR SOLICITUD DE PUBLICACIÓN (RESUMEN)

CU-5	Enviar solicitud de publicación
Actor	Productor
Descripción	El caso de se inicia cuando el productor envía una solicitud de publicación a partir de las diferentes opciones de solicitudes: Sitios Web o Páginas Informativas, con la descripción y pequeño resumen del contenido y finaliza con una respuesta del sistema donde se confirma entrega de solicitud.
Referencia	R14,R17

TABLA 7 CU APROBAR SOLICITUR DE PUBLICACIÓN (RESUMEN)

CU-6	Aprobar solicitud de publicación
Actor	Revisor
Descripción	El caso de uso se inicia cuando el revisor recibe la solicitud de publicación ya sea de SW o de Pág. Informativa, la aprueba según el contenido y finaliza cuando el sistema le envía confirmación de publicación al productor.
Referencia	R14,R15,R17,R18

TABLA 8 CU PUBLICAR SITIO WEB (RESUMEN)

CU-7	Publicar SW
Actor	Publicador
Descripción	El caso de uso se inicia cuando el productor recibe las solicitudes de los SW que han sido aprobados y finaliza con la publicación de los sitios.
Referencia	R15,R16

TABLA 9 CU PUBLICAR PÁGINA INFORMATIVA (RESUMEN)

CU-8	Publicar Página Informativa
Actor	Publicador
Descripción	El caso de uso se inicia cuando el productor recibe las solicitudes de las páginas informativas que han sido aprobadas y finaliza con la publicación de las mismas.
Referencia	R18,R19

TABLA 10 CU GESTIONAR USUARIO (RESUMEN)

CU-9	Gestionar usuario
Actor	Publicador
Descripción	El Caso de uso se inicia cuando el publicador realiza la gestión de usuario. Este puede agregar, modificar o eliminar a un usuario determinado.
Referencia	R20,R21,R22

TABLA 11 CU BUSCAR INFORMACIÓN DEL CONTENIDO DEL PORTAL (RESUMEN)

CU-10	Buscar información del contenido del Portal
Actor	Usuario
Descripción	El caso de uso se inicia cuando el usuario realiza una búsqueda a partir de un texto, palabra o frase que aparece en cualquiera de los contenidos que están publicados o se encuentran archivados y finaliza cuando el sistema devuelve la información buscada.
Referencia	R23

TABLA 12 CU MOSTRAR ESTADÍSTICAS DEL PORTAL (RESUMEN)

CU-11	Mostrar estadísticas del portal
Actor	Publicador
Descripción	El caso de uso se inicia cuando el publicador realiza una estadísticas de los sitios más visitados, cantidad de usuarios que han visitados el portal, servicios más visitados e información más buscada y finaliza cuando se muestran estas cifras en el portal.
Referencia	R24

2.6.3 Diagrama de Casos de Uso del Sistema.

En los diagrama de casos de uso del sistema representan las relaciones de incluye, extend y generalización/especialización entre los casos de uso y entre los actores, así como los casos de uso asociados con la seguridad. Estos se pueden estructurar en paquetes y poner el diagrama con las dependencias entre ellos.

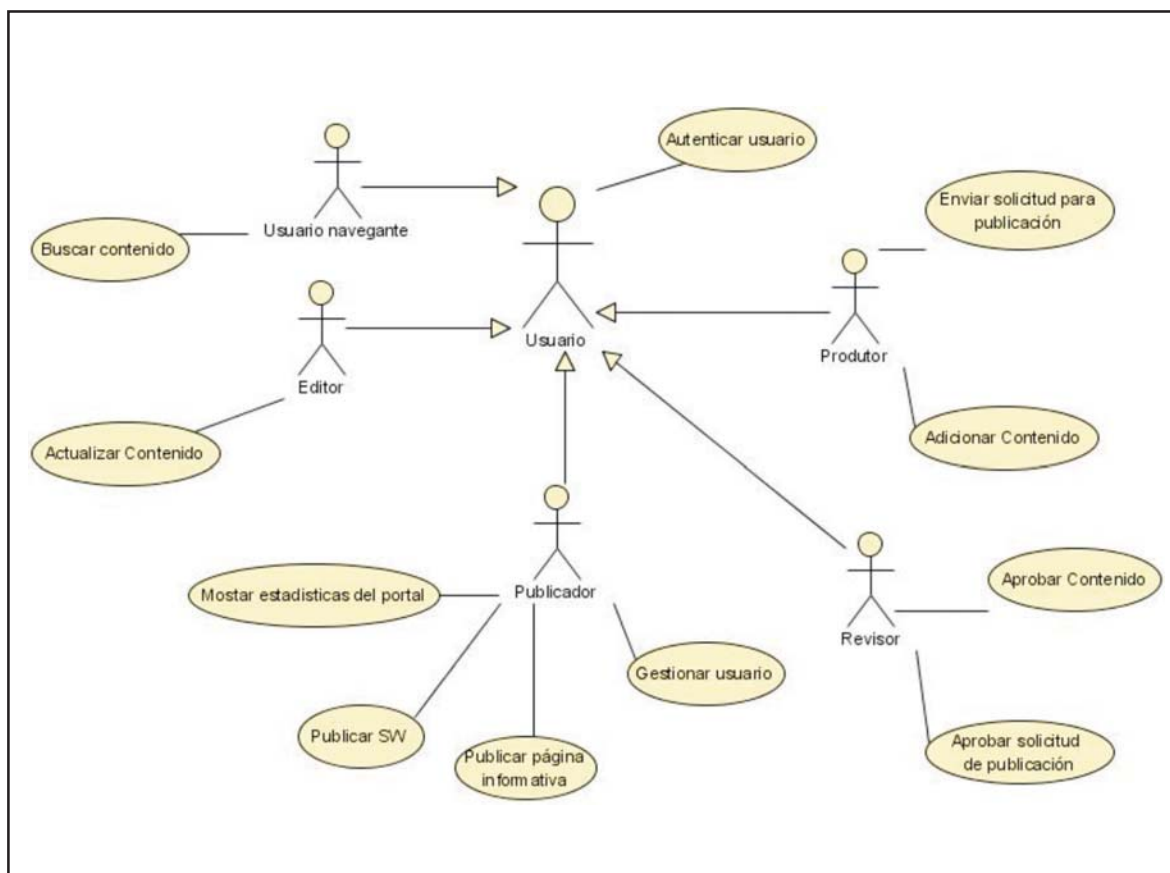


FIG. 17 DIAGRAMA DE CUS.

2.6.4 Casos de uso expandidos.

TABLA 13: CU AUTENTICAR USUARIO

Caso de uso	
CU-1	Autenticar usuario
Propósito	Permitir la entrada de los usuarios al sistema.
Actores	Usuario(Inicia)
Resumen El caso de usos lo inicia el usuario que accederá al sistema entrando su login y contraseña.	
Referencias	R1,RNF10
Acción del actor	Repuesta del sistema
1- El usuario accede al portal	2- El sistema solicita nombre de usuario y contraseña
3- El usuario inserta los datos solicitados	4- El sistema comprueba que el usuario esté en el dominio, en caso de que no pertenezca al dominio. Ver CA1 .
Caso alternativo	
CA1- Muestra mensaje de error y termina el caso de uso.	

TABLA 14: CU ADICIONAR CONTENIDO

Caso de uso	
CU-2	Adicionar contenido
Propósito	Adicionar los contenidos que serán publicados en el portal.
Actores	Productor(Inicia)
Resumen El caso de uso se inicia cuando el productor confecciona el contenido que será publicado en el portal. Este consta de diferentes categorías: noticias, efemérides y avisos. El CU termina cuando el contenido es enviado al editor.	
Referencias	R2,R6,R10
Acción del actor	Repuesta del sistema
1- El productor confecciona el contenido	2- El sistema muestra las categorías del contenido.Ver anexo 5 (Fig. 48)

3- El productor elige categoría del contenido a adicionar.	4 a-) Si el contenido es una noticia ver Sección Noticia. b-) Si el contenido es un aviso ver Sección Avisos. c-) Si el contenido es una efeméride ver Sección Efeméride.
Sección «Noticia»	
1- El productor selecciona adicionar una nueva noticia.	2- El sistema muestra la Sección de insertar los datos de la noticias. Ver anexo 5(Fig. 49)
3- El productor inserta los datos de la noticia a partir de los medios de comunicación o tomando en cuenta los sucesos ocurridos últimamente en la Universidad y el país.Oprime el botón guardar.	4- El sistema adiciona las noticias en el SGC.
Sección «Aviso»	
1-El productor selecciona insertar un nuevo aviso.	2- El sistema muestra una interfaz para insertar los datos del aviso.Ver anexo 5 (Fig. 51)
3-El Productor inserta los datos solicitados y oprime el botón guardar.	4- El sistema adiciona el aviso en el SGC
Sección «Efeméride»	
1-El productor selecciona insertar una nueva efeméride.	2- El sistema muestra una interfaz para adicionar la efeméride. Ver anexo 5 (Fig. 53)
3- El productor inserta los datos y oprime el botón guardar.	4-.El sistema adiciona las efemérides en el SGC.

TABLA 15: CU ACTUALIZAR CONTENIDO.

Caso de uso	
CU-3	Adicionar contenido
Propósito	Actualizar el contenido que será publicado en el portal.
Actores	Editor(Inicia)
Resumen El caso de uso se inicia cuando el editor recibe el contenido a publicar, este es corregido y arreglado según el manual de estilos, y termina cuando el contenido es actualizado en el SGC y enviado al revisor para su aprobación.	

Referencias	R2,R3,R8,R7,R10,R12
Acción del actor	Repuesta del sistema
1- El editor verifica en el SGC el contenido a publicar.	2-El sistema muestra el contenido por categorías encontradas en el SGC.
3- El editor elige categoría del contenido.	4 a-) Si el contenido es una noticia ver Sección Noticia. b-) Si el contenido es un aviso ver Sección Avisos. c-) Si el contenido es una efeméride ver Sección Efeméride.
Sección «Noticia»	
1- El editor selecciona actualizar una noticia.	2- El sistema muestra una interfaz con todas las noticias y sus datos. Ver anexo 5 (Fig.50)
3- El editor selecciona la noticia que va a actualizar.	4- El sistema muestra una interfaz con la noticia.
5- El editor aplica los cambios a la noticia según el manual de normas y estilos. Oprime el botón guardar.	6- El sistema actualiza la noticia en el SGC.
Sección «Avisos»	
1- El editor selecciona actualizar un aviso.	2- El sistema muestra una interfaz con todos los avisos y sus datos. Ver anexo 5 (Fig.52)
3-El editor selecciona el aviso que va a actualizar.	4- El sistema muestra una interfaz con el aviso.
5- El editor corrige el aviso.	6- El sistema actualiza los avisos en el SGC.
Sección «Efemérides»	
1- El editor selecciona actualizar efeméride	2- El sistema muestra una lista con todas las efemérides y sus datos.
3- El editor selecciona la efeméride que va a actualizar.	4- El sistema muestra una interfaz con la efeméride.
5- El editor corrige las efemérides.	6- El sistema actualiza la efeméride en el SGC.

TABLA 16: CU APROBAR CONTENIDO.

Caso de uso	
CU-4	Aprobar contenido
Propósito	Aprobar el contenido que será publicado en portal.
Actores	Revisor(Inicia)
Resumen El caso de uso se inicia cuando el revisor recibe el contenido actualizado, a partir de la categoría de este es seleccionado el revisor correspondiente. Si esta cumple con todas las normativas de publicación es aprobado, sino es enviado nuevamente al editor para aplique nuevas correcciones.	
Referencias	R3,R4,R7,R8,R11,R12
Acción del actor	Repuesta del sistema
1- El revisor verifica el contenido actualizado en el SGC.	2- El sistema muestra el contenido actualizado por categoría encontrado en el SGC.
3- El revisor elige categoría del contenido.	4 a-) Si el contenido es una noticia ver Sección Noticia. b-) Si el contenido es un aviso ver Sección Avisos. c-) Si el contenido es una efeméride ver Sección Efeméride.
Sección «Noticia»	
1- El revisor selecciona las noticias actualizadas en el SGC	2- El sistema muestra una interfaz con las noticias actualizadas y sus datos. Ver anexo 5 (Fig. 56)
3- El revisor selecciona la noticia a aprobar.	4- El sistema muestra una interfaz con la noticia.
5- El revisor a partir de la categoría de la noticia aplica el manual de las normas editoriales de la intranet para este tipo de contenido y lo aprueba.	6- El sistema actualiza las noticias en el SGC, si es aprobada sino ver CA1 .
CA1	
	1-El sistema envía la noticia al editor para aplicar nuevas correcciones.

CA1	
	1- El sistema envía la noticia al editor para aplicar nuevas correcciones.
Sección «Avisos»	
1- El revisor selecciona los avisos actualizados en el SGC.	2- El sistema muestra una interfaz con los avisos y sus datos. Ver anexo 5 (Fig. 56)
3- El revisor selecciona el aviso a aprobar.	4- El sistema muestra una interfaz con el aviso.
5- El revisor a partir de la categoría de los avisos aplica el manual de normas editoriales de la intranet y lo aprueba.	6- El sistema actualiza los avisos en el SGC, si son aprobados sino ver CA1 .
CA1	
	1- El sistema envía el aviso al editor para aplicar nuevas correcciones.
Sección «Efemérides»	
1- El revisor selecciona las efemérides actualizadas en el SGC.	2- El sistema muestra una interfaz con todas las efemérides y sus datos. Ver anexo 5 (Fig.56)
3- El revisor selecciona la efeméride a aprobar.	4- El sistema muestra una interfaz con la efeméride.
5- El revisor aplica las normas editoriales y aprueba la efeméride.	6- El sistema actualiza las efemérides en el SGC, si son aprobados sino ver CA1 .
CA1	
	1- El sistema envía las efemérides al editor para aplicar nuevas correcciones.

A partir de la extensión de las descripciones de los CUS siguientes, se decidió crear una documentación para el proyecto encargada de guardar estos artefactos y solo aparecerán en el cuerpo del documento los relacionados con la publicación de contenido; servicio primario del Portal .

En este capítulo se comenzó a profundizar en el desarrollo de la propuesta de solución, obteniéndose una lista de las funcionalidades que debe tener el sistema, las mismas fueron representadas mediante un Diagrama de Casos de Uso, y finalmente fueron descritas todas las acciones que realizan los actores y el sistema en general. A partir de aquí se puede comenzar a construir el sistema, cumpliendo con todos los requerimientos y las funcionalidades que se consideraron en este capítulo.

CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA

En el siguiente capítulo se aborda la propuesta del análisis y diseño de la Intranet 2 por lo que es más esquemático que teórico, se tratan temas referentes a como debe ser diseñada la Intranet. Se usan herramientas y metodologías que incluyen principios y diagramas que ilustran la forma en que los componentes de software interactúan y se comportan dentro del sistema. Se tratan temas específicos sobre los modelos de análisis y diseño y sobre su construcción como la propone RUP para el desarrollo de software, para lograr un software robusto y de gran calidad.

3.1 Modelo de análisis.

3.1.1 Definición del modelo de análisis.[41]

El Modelo de análisis contiene clases de análisis y sus objetos organizados en paquetes que colaboran.

Las clases de análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización/especialización y tipos asociativos.

RUP propone clasificar las clases en:

- Clases de Interfaz: Modelan la interacción entre el sistema y sus actores.
- Clases entidad: Modelan información que posee larga vida y que es a menudo persistente.
- Clases de control.: Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

3.1.1.1 Diagramas de clases de análisis.

Para cada caso de uso del sistema se realizó un diagrama de clases de análisis, pero para mayor organización del documento los restantes diagramas se encuentran en la documentación del proyecto.

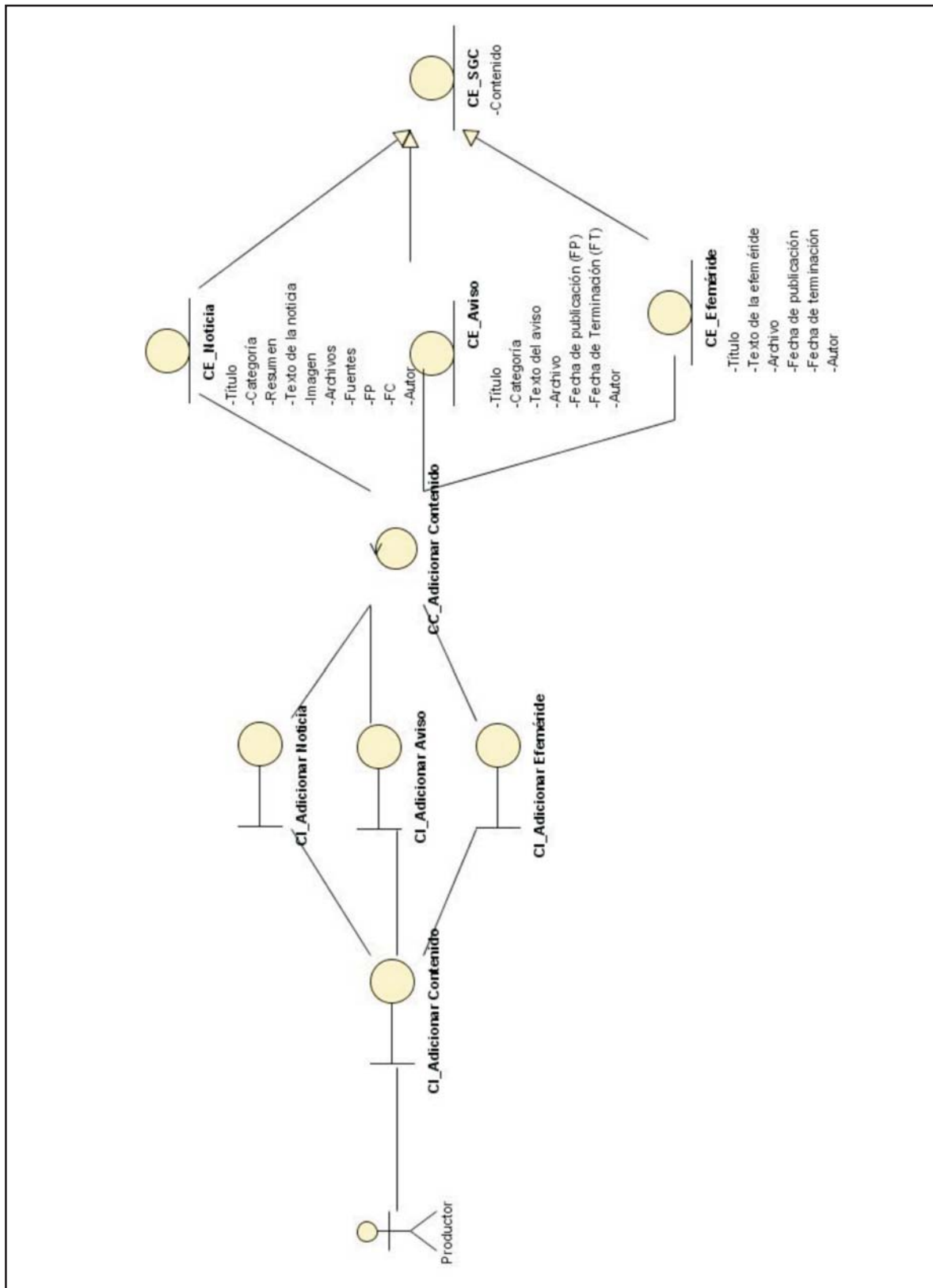


FIG. 18 DIAGRAMA DE CLASES DE ANÁLISIS DE CUSADICIONAR CONTENIDO.

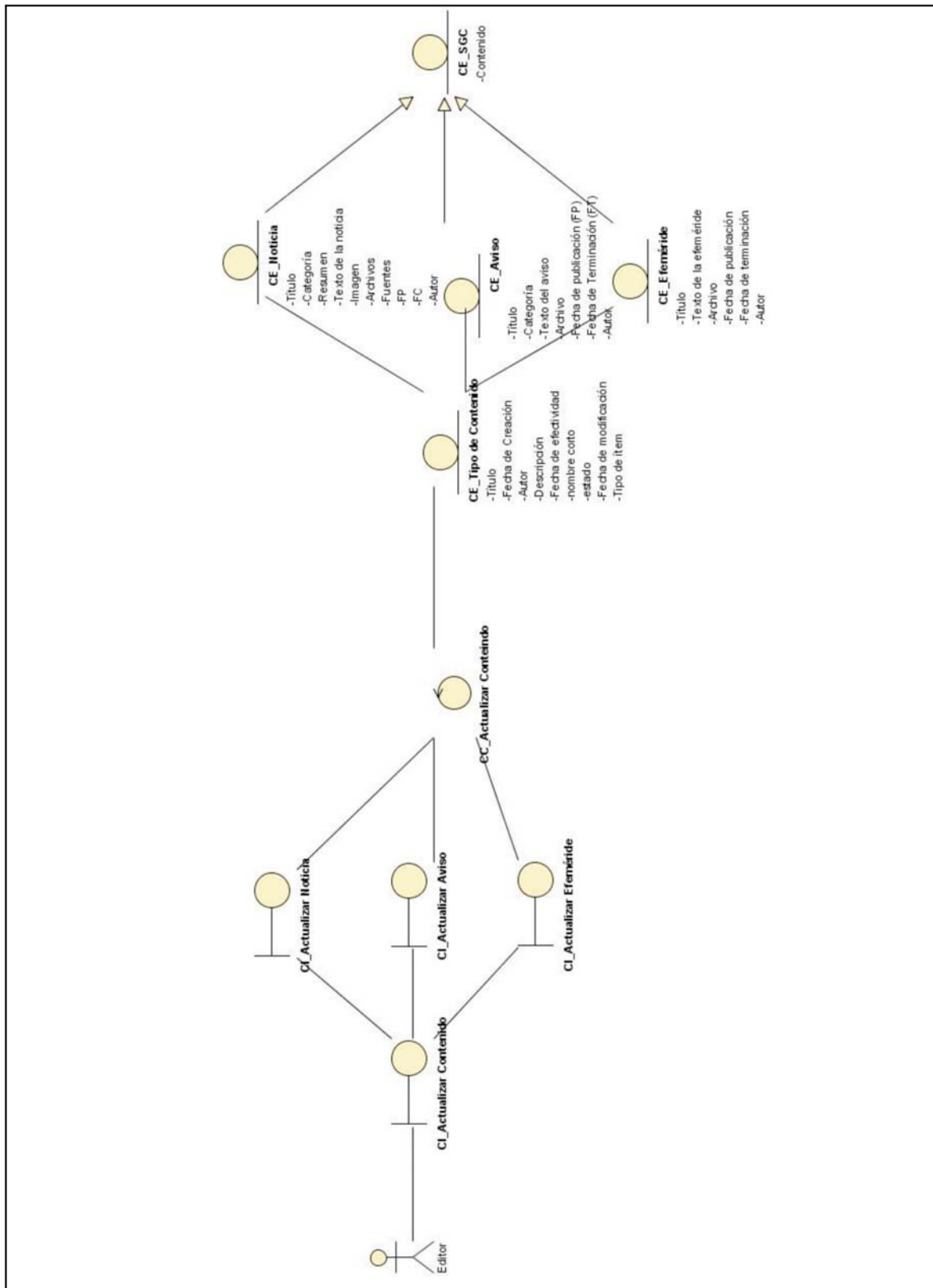


FIG. 19 DIAGRAMA DE CLASES DE DEL CUS ANÁLISIS ACTUALIZAR CONTENIDO.

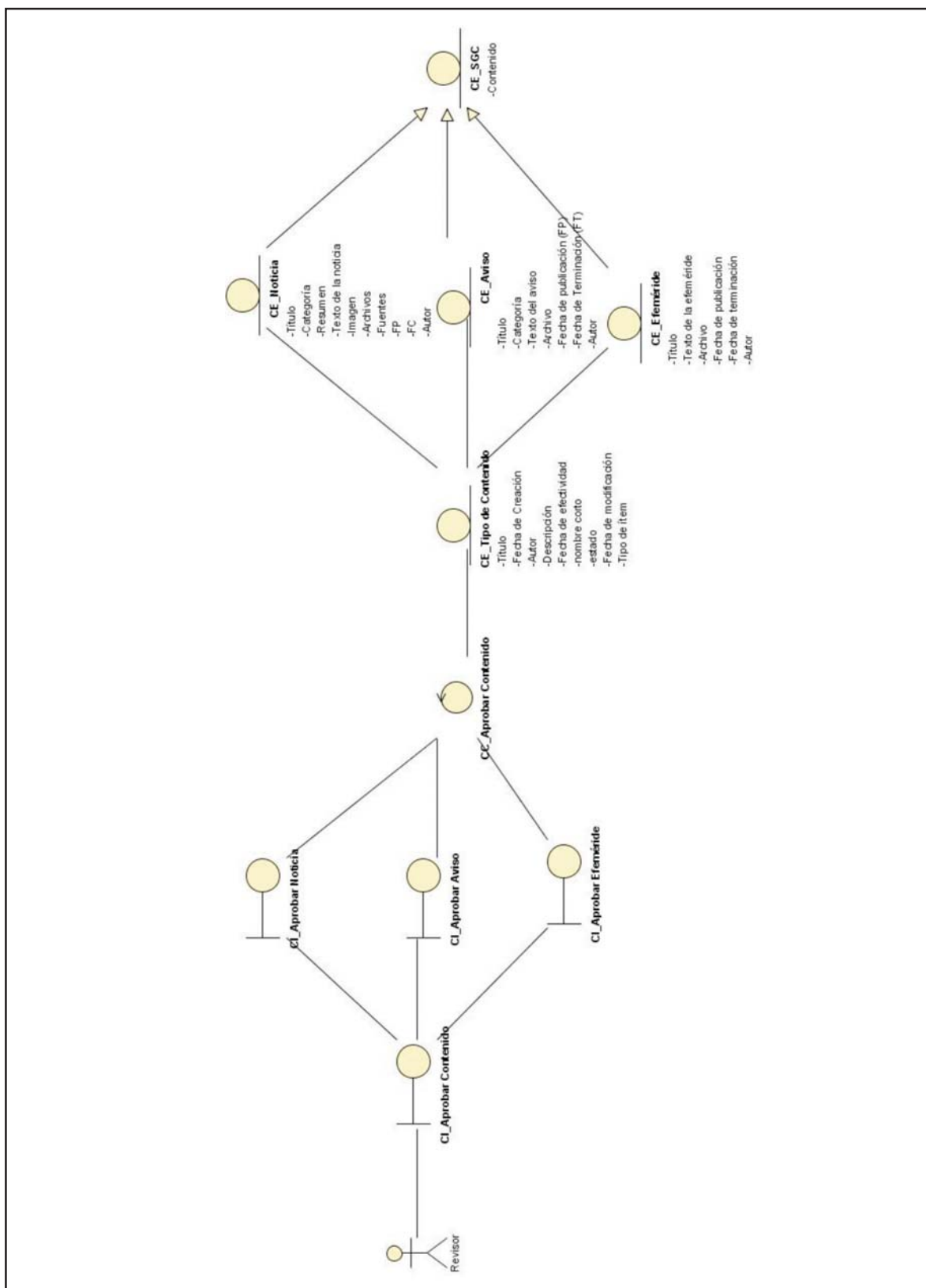


FIG. 20 DIAGRAMA DE CLASES DE ANÁLISIS DEL CUS APROBAR CONTENIDO.

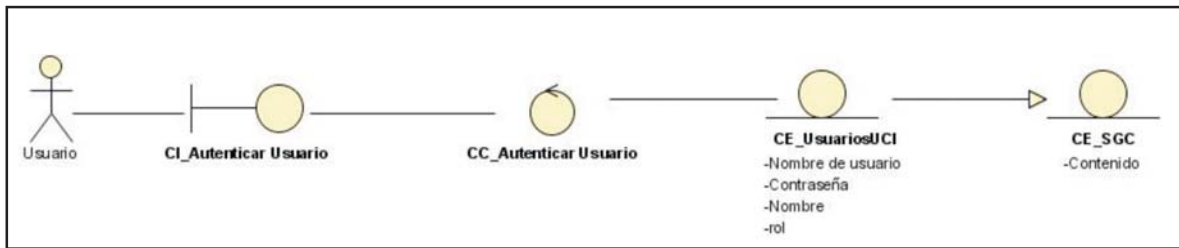


FIG. 21 DIAGRAMA DE CLASES DE ANÁLISIS DEL CUS AUTENTICAR USUARIO.

3.2 Patrones de diseño.

Para el diseño de software se usaron los patrones de asignación de responsabilidades o GRAPS de estos se usaron principalmente los patrones experto y creador, pues son los más aplicables para este tipo de trabajo.

En el diseño web se usará el patrón MVC por las ventajas que propone para el funcionamiento de una aplicación web. [Ver Capítulo 1 Sección 1.9]

3.3 Modelo de Diseño.

3.3.1 Definición del modelo de diseño.[42]

El modelo de diseño es un modelo que describe los objetos que:

- Describe la realización física de los casos de uso.
- Se centra en como los requisitos funcionales y no funcionales tienen impacto en el sistema.

RUP propone que el artefacto Modelo de Diseño básicamente contenga:

- Introducción: Una descripción textual que sirve como breve introducción al modelo.
- Paquetes y Subsistemas de Diseño: Los paquetes y subsistemas del diseño representados en una jerarquía y una breve descripción de ellos.
- Diagramas: los diagramas de clases del diseño y diagramas de interacción (colaboración y/o secuencia) del diseño. Estos últimos también llamados realización de casos de uso.
- Clases, interfaces, relaciones, contenidas en los paquetes y una breve descripción de ellos.

3.3.2 Diagramas de interacción.[43]

Los Diagramas de interacción muestran una interacción concreta: un conjunto de objetos y sus relaciones, junto con los mensajes que se envían entre ellos.

- Modelan el comportamiento dinámico del sistema; el flujo de control en una operación.

- Describen la interacción entre objetos; los objetos interactúan a través de mensajes para cumplir ciertas tareas.
- Las interacciones proveen un «comportamiento» y típicamente implementan un Caso de Uso.

Existen dos tipos de diagramas de interacción en UML:

- Diagramas de Secuencia (dimensión temporal).
- Diagramas de Colaboración (dimensión estructural).

3.3.2.1 Diagrama de Secuencia.[44]

Un diagrama de secuencia muestra las interacciones entre objetos, ordenadas en secuencia temporal durante un escenario concreto.

Si los casos de uso tienen varios flujos o subflujos distintos, suele ser útil crear un diagrama de secuencia para cada uno de ellos.

Los diagramas de Secuencia orientados a la publicación de contenidos se encuentran en: [Ver Anexo 3], los restantes diagramas están suscritos a la documentación del proyecto.

3.4 Diagrama de clases de diseño.

Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones.

Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema, esto incluye modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas. Los diagramas de clases también son la base para un par de diagramas relacionados: los diagramas de componentes y los diagramas de despliegue. Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

Contienen normalmente los siguientes elementos:

- Clases.
- Interfaces.
- Colaboraciones.
- Relaciones de dependencia, generalización y asociación.

Al igual que los demás diagramas, los diagramas de clases pueden contener notas y restricciones. Los diagramas de clases también pueden contener paquetes o subsistemas, los cuales se usan para agrupar los elementos de un modelo en partes más grandes. A veces se colocarán instancias en los diagramas de clases, especialmente cuando se quiera mostrar el tipo (posiblemente dinámico) de una instancia. [45]

3.4.1 Diagrama de clases Web.

Para modelar las aplicaciones Web no se usa la forma tradicional de modelar las clases, se usa varios mecanismos de extensión que fueron definidos para extender su semántica en aquellas situaciones en las que no es posible capturar todas las características de una determinada arquitectura o dominio en particular. [46]

En esta tesis se elaboró un diagrama de clases Web para cada uno de los casos de uso del sistema [Ver Anexo 4], los orientados a la publicación de contenido, los restantes diagramas se pueden consultar en la documentación del proyecto.

3.5 Análisis y diseño en la calidad del software.

En el análisis se mide la calidad con el estudio de viabilidad que se basa fundamentalmente en que muchas veces cuando se emprende un proyecto de software no se cuenta con el tiempo y los recursos necesarios para su materialización sin tener pérdidas económicas y frustración profesional. [47]

La viabilidad y el análisis de riesgos están relacionados de muchas maneras, si el riesgo del proyecto es alto, la viabilidad de producir software de calidad se reduce, sin embargo se deben tomar en cuenta cuatro áreas principales de interés:

- Viabilidad económica: Una evaluación de los costos de desarrollo, comparados con los ingresos netos o beneficios obtenidos del producto o Sistema desarrollado.
- Viabilidad Técnica: Un estudio de funciones, rendimiento y restricciones que puedan afectar la realización de un sistema aceptable.
- Viabilidad Legal: Es determinar cualquier posibilidad de infracción, violación o responsabilidad legal en que se podría incurrir al desarrollar el Sistema.

Por el contrario el papel del diseño es muy importante en la calidad del software, tal es así que el diseño de software se puede definir en una sola palabra Calidad.

Pues es el diseño donde se fomenta la calidad del software, el diseño es la única manera de materializar los requerimientos del cliente. [48]

Para medir la calidad se deben definir un conjunto de criterios técnicos para un buen diseño como:

- Debe presentar un orden jerárquico que haga un uso inteligente del control entre los componentes del software.
- El diseño debe ser modular, es decir, se debe hacer una partición lógica del software en elementos que realicen funciones y subfunciones específicas.
- Un diseño debe contener abstracciones de datos y procedimientos.
- Debe producir módulos que presenten características de funcionamiento independiente.
- Debe conducir a interfaces que reduzcan la complejidad de las conexiones entre los módulos y el entorno exterior.

- Debe producir un diseño usando un método que pudiera repetirse según la información obtenida durante el análisis de los requisitos de software.

Siguiendo todos estos pasos logramos obtener un software con alta calidad, en nuestra tesis a la hora de realizar el análisis y diseño del software, lo hicimos guiándonos por información que contamos sobre la descripción de los requisitos que debe tener la intranet2 porque para un mejor trabajo necesitábamos la arquitectura de información de la intranet pero esta no estaba completa.

Nos guiamos por una descripción de algunos servicios que debía tener la intranet y seguimos todos los pasos antes descritos, usando para ello una metodología de desarrollo de software en específico RUP.

Y como herramienta para el análisis y diseño de software el visual paradigm [ver capítulo 1] que utiliza el UML [ver capítulo 1] como lenguaje de modelado.

3.6 Principios de diseño que se usaran.

3.6.1 Estándares de la interfaz de la aplicación.

Para el diseño web se usara un formato que conste con cinco partes fundamentales (Ver Fig. 29.):

- Cabecera.
- Menú superior
- Menú lateral.
- Contenido
- Pie.

Donde cada uno de ellos tienen contenidos distintos, por ejemplos:

La Cabecera tendrá el logo de la UCI y otras cosas para que la presentación de la Intranet se distinga de otras, se podrá acceder a la página principal con solo dar click en el logo.

El menú superior contara con los servicios que brindara la Intranet.

El menú lateral se pondrá a la derecha o a la izquierda en caso de lo que se quiera mostrar, y de la importancia que tenga, si son cosas que van a estar permanentes se aconseja ponerlos del lado derecho pues es más fácil acceder a ellos.

El contenido mostrara todo aquel contenido de la Intranet en dependencia del rol del usuario que este logeado.

En el pie lo que se acostumbra a poner es los autores, o aunque también se puede poner la compatibilidad con los navegadores, o la herramienta con que se construyó el software.



FIG. 22 ESTRUCTURA DEL DISEÑO WEB PROPUESTO PARA LA INTRANET2.

En este capítulo se dió paso a construir la propuesta del sistema obtenida en el capítulo 2, se construyó el diagrama de clases de análisis, diagramas de interacción, en este caso el de secuencia, además de las clases de diseño Web. Para la construcción de estos diagramas se utilizaron patrones de diseño para mejorar la calidad del software. Con éstos quedan sentadas las bases para seguir trabajando en la implementación de la Intranet 2.

CONCLUSIONES GENERALES

Terminado el estudio y la realización del análisis y diseño del sistema quedaron pautadas las normas para su posterior implementación, lográndose identificar pautas y normativas de seguridad así como de diseño. Se prevé una interfaz fácil de entender y accesible gracias a que cumple con estándares de diseño y patrones de diseño actualizados para este tipo de aplicación.

Para el desarrollo se realizó el análisis de las tecnologías y metodologías de desarrollo de software más usadas en la actualidad, concluyéndose en la utilización del RUP como metodología de desarrollo de software, el Visual Paradigm como herramienta para realizar el modelo visual de los artefactos generados en los flujos de análisis y diseño.

Se modeló el dominio, se definieron requerimientos del sistema, tanto funcionales como no funcionales de forma estructurada, se realizó el modelo de caso de uso de sistema.

El análisis y diseño se realizó a través de diagramas de clases de análisis, diagramas de interacción, en específico diagramas de secuencia por ser más claro al ahora de representarlo, diagramas de clases de diseño Web.

Por todo lo anterior se concluye que los objetivos propuestos para el presente proyecto han sido cumplidos satisfactoriamente, incluyendo una serie de recomendaciones que deben tenerse en cuenta para el trabajo futuro.

RECOMENDACIONES

.....

Con la finalización del trabajo se recomienda para el futuro desarrollo e implementación de la Intranet 2:

- Realizar un profundo estudio de las tendencias actuales en cuanto a las Intranets colaborativa, para que la nueva Intranet cumpla con ese principio.
- Mayor seriedad en la entrega de la información necesaria para realizar los diferentes artefactos de los flujos de trabajos encontrados en la metodología seleccionada.
- Se realice el levantamiento de los procesos, servicios e información de las diferentes áreas y direcciones de la Universidad.
- Continuar con el proyecto Intranet 2, para que sirva como soporte para los diferentes sitios web de la universidad en la migración al software libre.
- En vista a mejorar la imagen visual de la Intranet se realice un manual de estilos para esta y los diferentes sitios web de la Universidad.
- Integrar nuevos servicios a la Intranet que faciliten la vida interna en la universidad.
- Continuar con el análisis y diseño de la nueva Intranet y adicionar el flujo de implementación para la misma.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Rodrigo Fuentes. *Intranet*.2002. [Disponible en: <http://www.monografias.com/trabajos12/intrants/intrants.shtml#VISION>]
- [2] Patricia Cascante. Mirna Fonseca. *Intranet: Una ventana a la comunicación interna de las organizaciones*. Revista de Derecho y Tecnologías de la Información.UNED, Costa Rica, N° 4, (Pág. 3- Pág. 5)
- [3] IDEM 1
- [4] IDEM 2
- [5] IDEM 1
- [6] IDEM 1
- [7] Patricia Cascante. Mirna Fonseca. *Intranet: Una ventana a la comunicación interna de las organizaciones*. Revista de Derecho y Tecnologías de la Información. UNED, Costa Rica, N° 4, (Pág. 6- Pág. 12)
- [8] Andreas Kaiser. Software Libre. 2000. [Disponible en: <http://www.atela.net/contenido/articulos/software-libre.html>]
- [9] Ingeniería del software 1.Fase de Inicio. Flujo de trabajo de requerimientos, 2006-2007
- [10] IDEM 9
- [11] Dra. Lourdes García Ávilalng. Leidy Fernández Sánchez. Procedimiento para el desarrollo del proceso de ingeniería de requisitos en un proyecto software (PROCIR). Departamento de Sistemas y Computación. Instituto Tecnológico de Morelia. México.Pag.17.
- [12] IDEM 9
- [13] Heymans Xavier. *Qué es Plone: introducción y estudios de casos* (slides).2006 [Disponible en: http://www.communesplone.org/Members/xavier_heyman/sp/Plone-portalesExtranet.pdf/view?set_language=es-es]
- [14] Ro y Norrick, Aprenda Bases de Datos con MS SQL SERVER 2000, 2003, [Disponible en: <http://usuarios.lycos.es/cursosgbd/>]
- [15] Mario G. Fundamentos y modelos de bases de datos.1997
- [16] About PostgreSQL.2006. [Disponible en: <http://www.postgresql.org/>]
- [17] IDEM 14
- [18] Ingeniería del software 1.*Introducción a la Ingeniería de Software*.2006-2007 [Disponible en: <http://www.eubd.ucm.es/html/personales/enred/mantonia/docauto/tema5/tema5.htm>]

[19] Barrientos Enríquez, Aleida Mirian. *El proceso Unificado de Modelado (RUP)*. (Noviembre-2005). [Disponible en: <http://www.monografias.com/trabajos16/lenguaje-modelado-unificado/lenguaje-modelado-unificado.shtml#PROCESO>]

[20] IDEM 16

[21] IDEM 19

[22] María A. Mendoza Sánchez. *Metodologías De Desarrollo De Software*. (Junio -2004) [Disponible en: <http://www.informatizate.net>]

[23] MÉTRICA. Metodología de planificación y desarrollo de Sistemas de Información. (2000). [Disponible en: <http://www.csi.map.es/csi/pg5m42.htm>]

[24] Ricardo Devis Botella, Tecnología de Objetos PATRONES DE DISEÑO, 1996, [Disponible en: <http://www.arrakis.es/~devis/patrones.ppt>]

[25] Roger S.Pressman. Ingeniería de Software: Un enfoque práctico. Madrid y Carachelejo (España), 2001. 614 pág.

[26] SOFTWARE, D.D.I.D. Patrones. Conferencia Patrones 2.0, 2004-2005. 21 pág.

[27] LARMAN, C. UML Y patrones. Mexico. 1999. 536 pág.

[28] Patrones de diseño: Introducción. 2006. [Disponible en: http://www.thealphasite.org/patrones_de_diseño_introduccion].

[29] Colectivo de autores, Materiales, 2007, [Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=10996>].

[30] IDEM 28

[31] IDEM 29

[32] IDEM 29

[33] Fase de Inicio. Modelo del Negocio, 2006-2007.

[34] Rodrigo Fuentes. Intranet, 1997. [Disponible en: <http://www.monografias.com/trabajos12/intrants/intrants.shtml#VISION>].

[35] Héctor Fabio Alban. *El concepto de servicio*. 2007. [Disponible en: http://www.wikilearning.com/el_concepto_de_servicio-wkccp-11501-2.htm]

[36] IDEM 9

[37] IDEM 9

[38] IDEM 9

[39] IDEM 8

[40] IDEM 8

[41] D. D. I. Software, «Conferencia 5,» *Fase de Inicio. Flujo de Análisis y Diseño. Modelo de Análisis*, 2006

[42] D.D.I.Software, «Conferencia- _Diseño», 2007, [Disponible en: <http://teleformacion.uci.cu/course/view.php?id=43>].

[43] D.D.I.Software,«Conferencia_6», [Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=10896>].

[44] IDEM 43

[45] IDEM 42

[46] D.D.I.Software, «Extensiones para Web», *Extensiones para Web*, 2006, [Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=10918>].

[47] Pedro Concepción Nova, «Análisis y Diseño de Sistemas», 2003, [Disponible en: <http://www.ilustrados.com/publicaciones/EpyppplVpEbjHkcuHD.php#>].

[48] IDEM 47

BIBLIOGRAFÍA

.....

About PostgreSQL.2006. [Disponible en: <http://www.postgresql.org/>]

ALONSO, J. M. Estándares Web del W3C, 2005. [Disponible en: <http://www.w3c.es/Presentaciones/2005/0314-estandares-JA/>]

Andreas Kaiser. Software Libre. 2000. [Disponible en: <http://www.atela.net/contenido/articulos/software-libre.html>]

ANÓNIMO. Diseño Centrado en el Usuario, DCU. [Disponible en: <http://griho.udl.es/mpiua/dcu.htm>]

ANÓNIMO. Diseño de Sistemas. [Disponible en: <http://iteso.mx/~adrianay/sesion14.ppt#256,1,DISEÑO DE SISTEMAS>]

ANÓNIMO. Grandes mentiras: la accesibilidad no es un coste añadido, 2007. [Disponible en: <http://www.grancomo.com/weblog2.php?idpost=97>]

ANÓNIMO. ¿Que es esto del CMS? [Disponible en: <http://www.simplewebportal.net/host/>]

Barrientos Enríquez, Aleida Mirian. *El proceso Unificado de Modelado (RUP)*. (Noviembre-2005). [Disponible en: <http://www.monografias.com/trabajos16/lenguaje-modelado-unificado/lenguaje-modelado-unificado.shtml#PROCESO>]

Colectivo de autores, Materiales, 2007, [Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=10996>].

D. D. I. Software, «Conferencia 5,» *Fase de Inicio. Flujo de Análisis y Diseño. Modelo de Análisis*, 2006

D.D.I.Software, «Conferencia- _Diseño», 2007, [Disponible en: <http://teleformacion.uci.cu/course/view.php?id=43>].

D.D.I.Software,«Conferencia_6», [Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=10896>].

D.D.I.Software, «Extensiones para Web», *Extensiones para Web, 2006*, [Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=10918>].

Dra. Lourdes García ÁvilaIng. Leidy Fernández Sánchez. Procedimiento para el desarrollo del proceso de ingeniería de requisitos en un proyecto software (PROCIR). Departamento de Sistemas y Computación. Instituto Tecnológico de Morelia. México.Pag.17.

DUEÑAS, J. B. Cómo configurar Squid: Parámetros básicos para Servidor Intermediario (Proxy). , 2007. [Disponible en: <http://www.linuxparatodos.net/portal/staticpages/index.php?page=19-0-como-squid-general>]

Fase de Inicio. Modelo del Negocio, 2006-2007.

Héctor Fabio Alban. *El concepto de servicio*.2007. [Disponible en: http://www.wikilearning.com/el_concepto_de_servicio-wkccp-11501-2.htm]

Heymans Xavier. *Qué es Plone: introducción y estudios de casos* (slides).2006 [Disponible en: http://www.communesplone.org/Members/xavier_heyman/sp/Plone-portalesExtranet.pdf/view?set_language=es-es]

Ingeniería del software 1.Fase de Inicio. Flujo de trabajo de requerimientos, 2006-2007.

Ingeniería del software 1.*Introducción a la Ingeniería de Software*.2006-2007 [Disponible en: <http://www.eubd.ucm.es/html/personales/enred/mantonia/docauto/tema5/tema5.htm>].

LARMAN,C.UML Y patrones.Mexico.1999.536 pág.

María A. Mendoza Sánchez. *Metodologías De Desarrollo De Software*. (Junio -2004) [Disponible en:<http://www.informatizate.net>]

Mario G. Fundamentos y modelos de bases de datos.1997.

MARTÍNEZ, A. Y. Una metodología para el diseño de sistemas de información, basada en el estudio de sistemas blandos, 2004. [Disponible en: <http://www.revistaespacios.com/a04v25n02/04250231.html>]

MARIÑÁN, M. P. Los peligros de los patrones de diseño. [Disponible en: <http://www.solotuweb.com/fs-id~6723.html>]

MÉTRICA. Metodología de planificación y desarrollo de Sistemas de Información. (2000). [Disponible en: <http://www.csi.map.es/csi/pg5m42.htm>].

Patricia Cascante. Mirna Fonseca. *Intranet: Una ventana a la comunicación interna de las organizaciones*. Revista de Derecho y Tecnologías de la Información.UNED, Costa Rica, Nº 4, (Pág. 3- Pág. 5).

Patricia Cascante. Mirna Fonseca. *Intranet: Una ventana a la comunicación interna de las organizaciones*. Revista de Derecho y Tecnologías de la Información. UNED, Costa Rica, Nº 4, (Pág. 6- Pág. 12).

Pedro Concepción Nova, «Análisis y Diseño de Sistemas», 2003, [Disponible en: <http://www.ilustrados.com/publicaciones/EpyppplVpEbjHkcuHD.php#>].

Patrones de diseño: Introducción. 2006. [Disponible en: http://www.thealphasite.org/patrones_de_diseño_introduccion].

Ricardo Devis Botella, Tecnología de Objetos PATRONES DE DISEÑO, 1996, [Disponible en: <http://www.arrakis.es/~devis/patrones.ppt>]

Rodrigo Fuentes. Intranet, 1997. [Disponible en: <http://www.monografias.com/trabajos12/intrants/intrants.shtml#VISION>].

Ro y Norrick, Aprenda Bases de Datos con MS SQL SERVER 2000, 2003, [Disponible en: <http://usuarios.lycos.es/cursosgbd/>]

RUMBAUGH., I. J. G. B. J. El Proceso Unificado de desarrollo de software. Pearson Education S.A. Madrid, 2000. 464

ANEXO 1

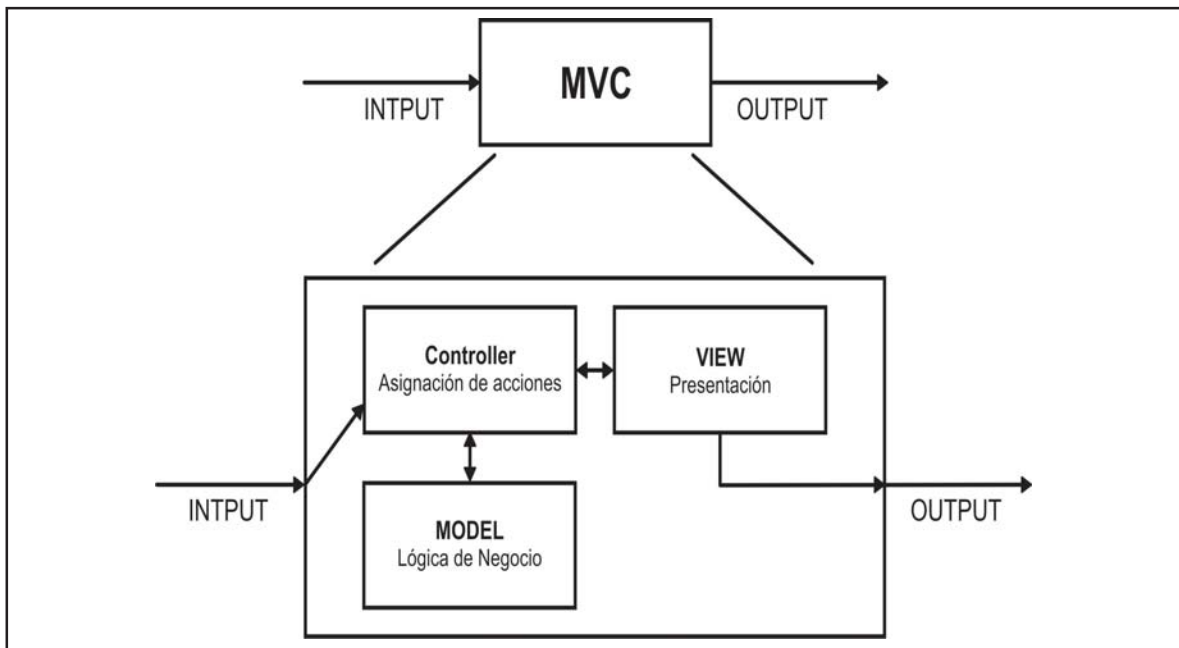


FIG. 23 ESTRUCTURA DEL MODELO VISTA CONTROLADOR (MVC).

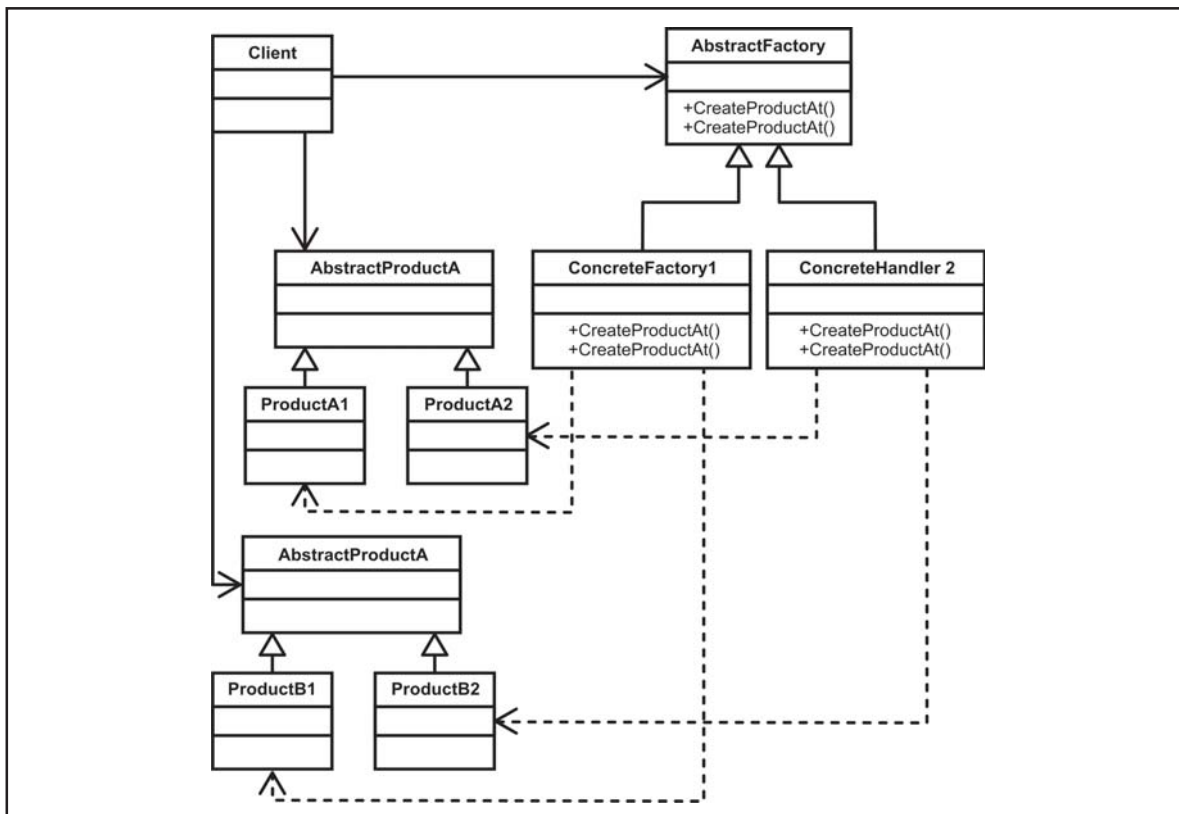


FIG. 24 ESTRUCTURA GENERAL DEL PATRÓN DE DISEÑO ABSTRACT FACTORY.

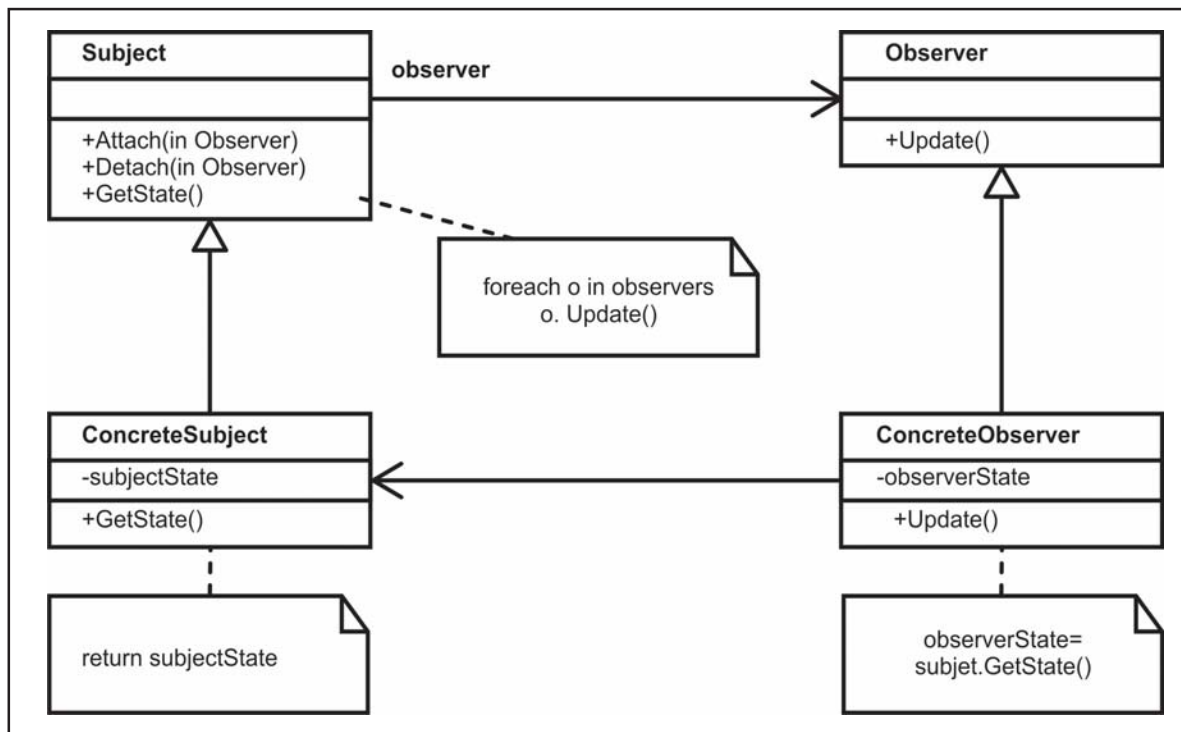


FIG. 25 ESTRUCTURA DEL PATRÓN DE DISEÑO OBSERVER.

ANEXO 2 MODELO DE DOMINIO

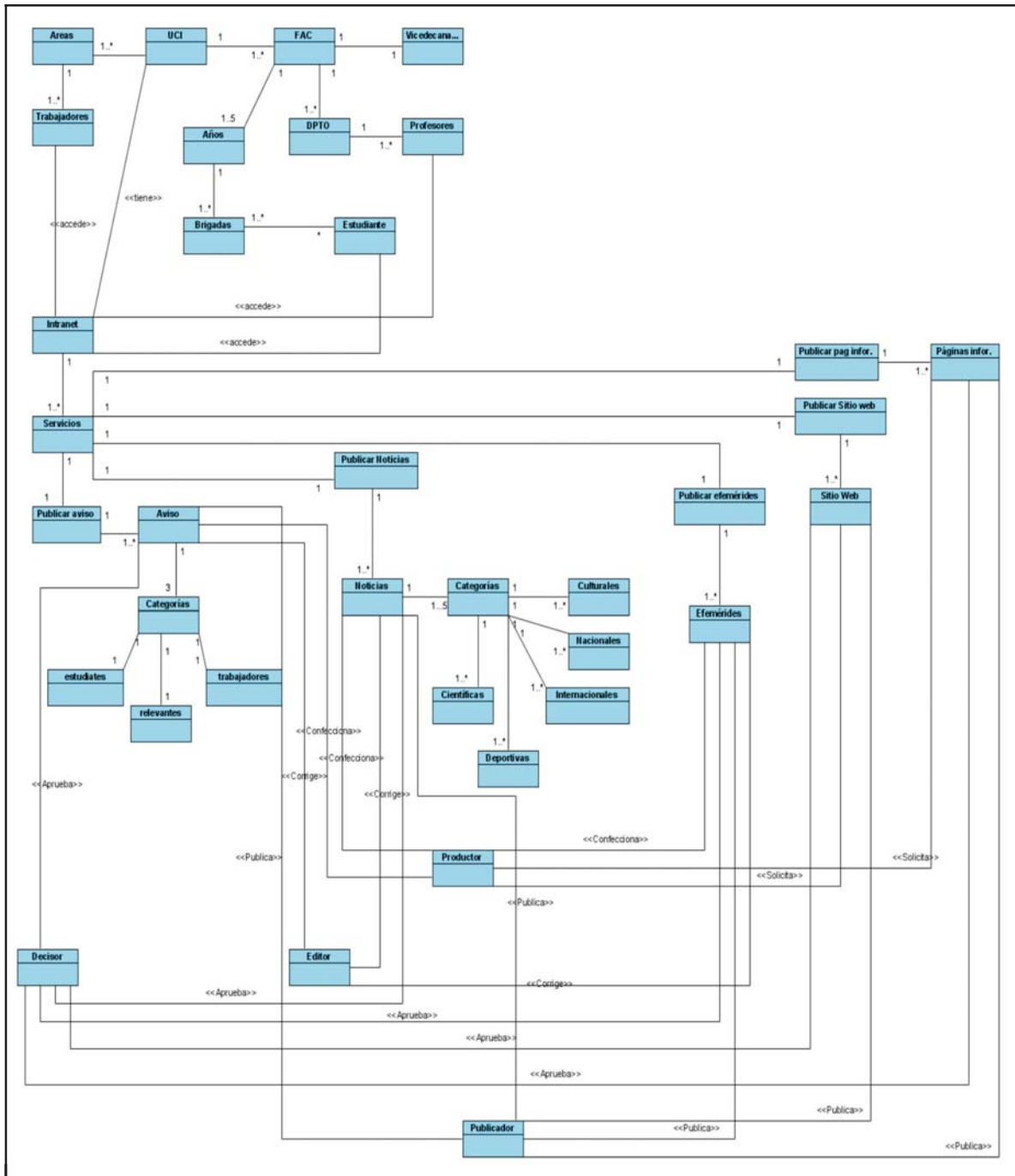


FIG. 26 MODELO DE DOMINIO.

ANEXO 3 DIAGRAMAS DE SECUENCIA.

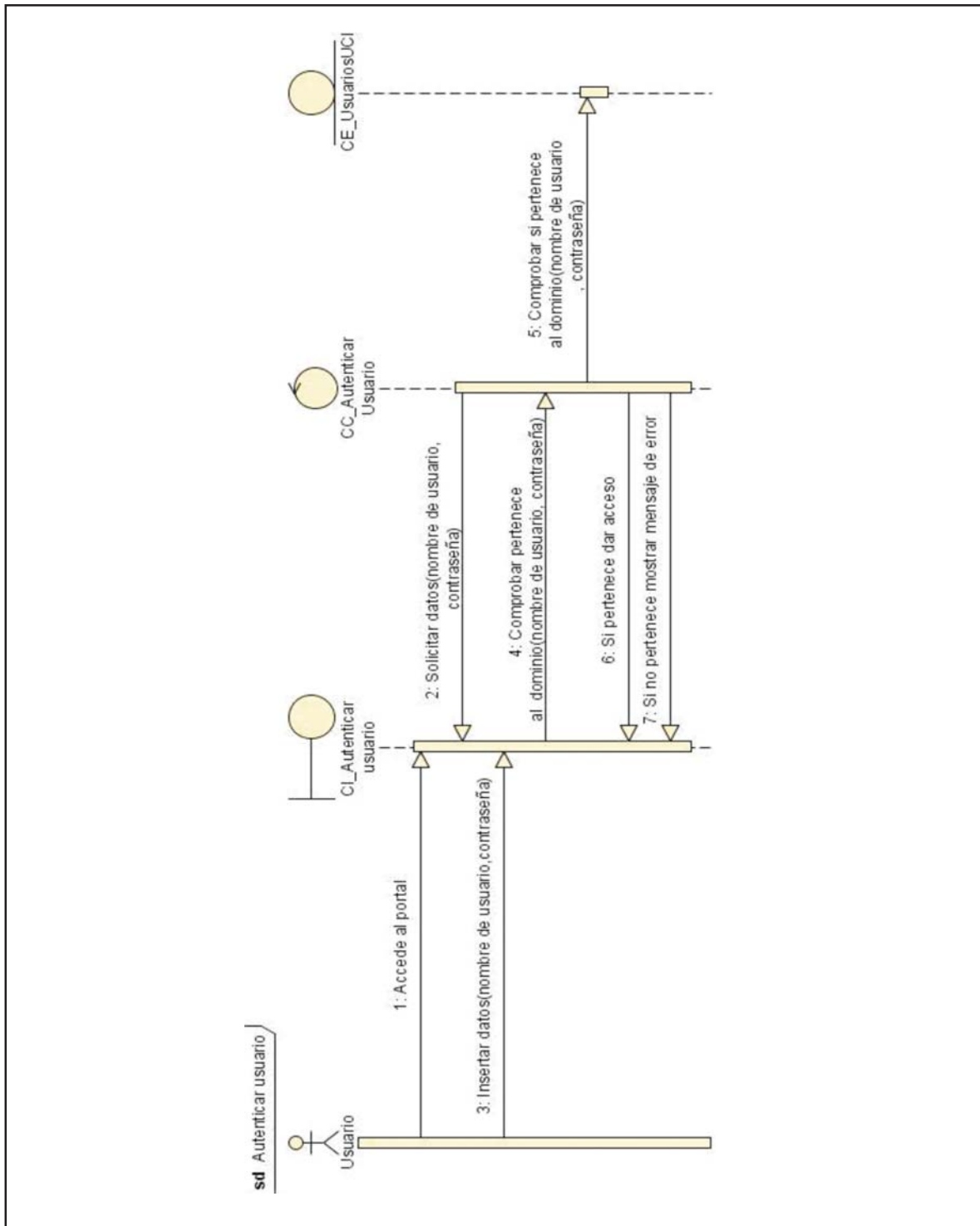


FIG. 27 DIAGRAMA DE SECUENCIA DEL CASO DE USO «AUTENTICAR USUARIO».

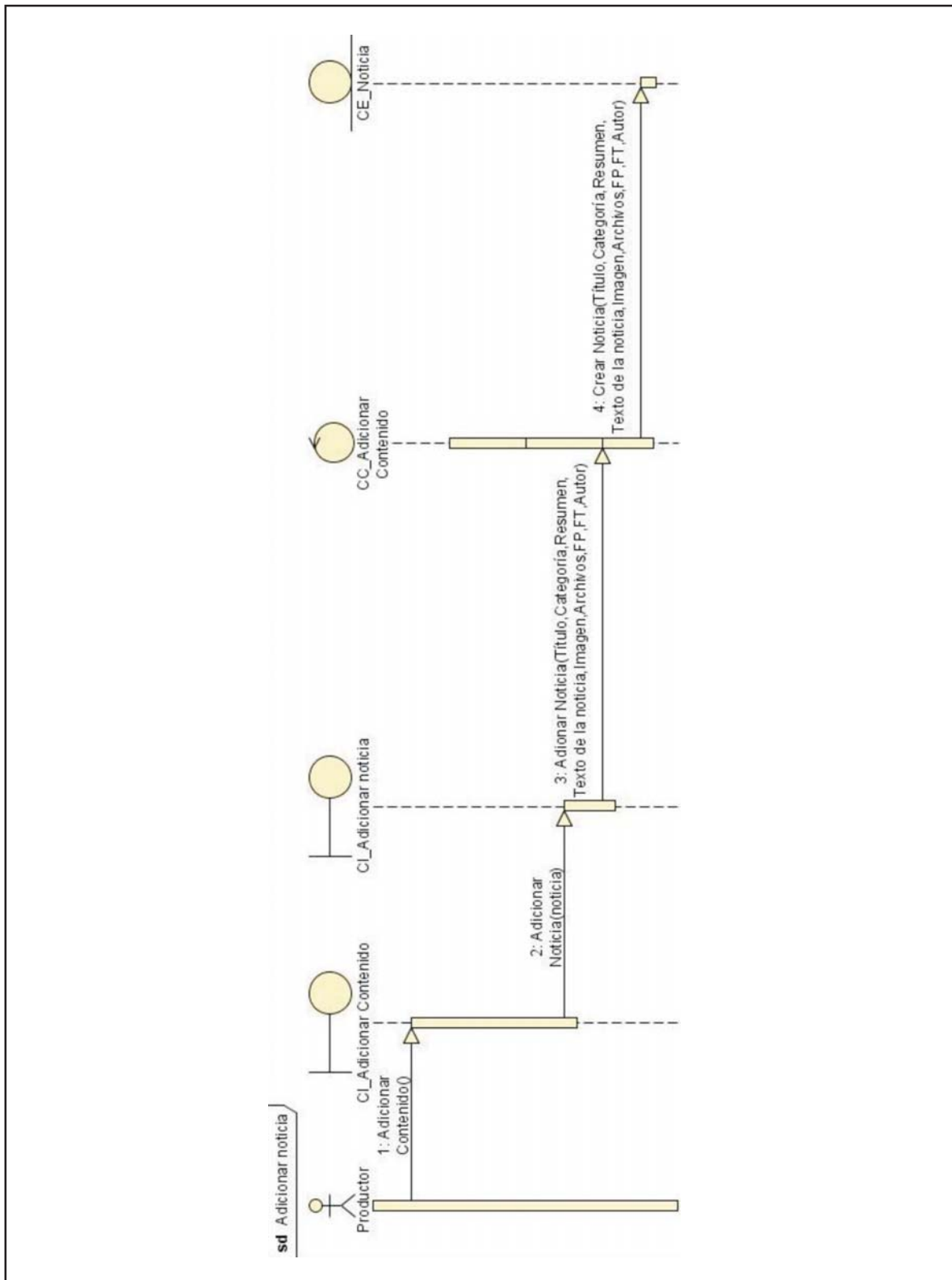


FIG.28 DIAGRAMA DE SECUENCIA DEL CASO DE USO «ADICIONAR CONTENIDO».SECCIÓN «ADICIONAR NOTICIA».

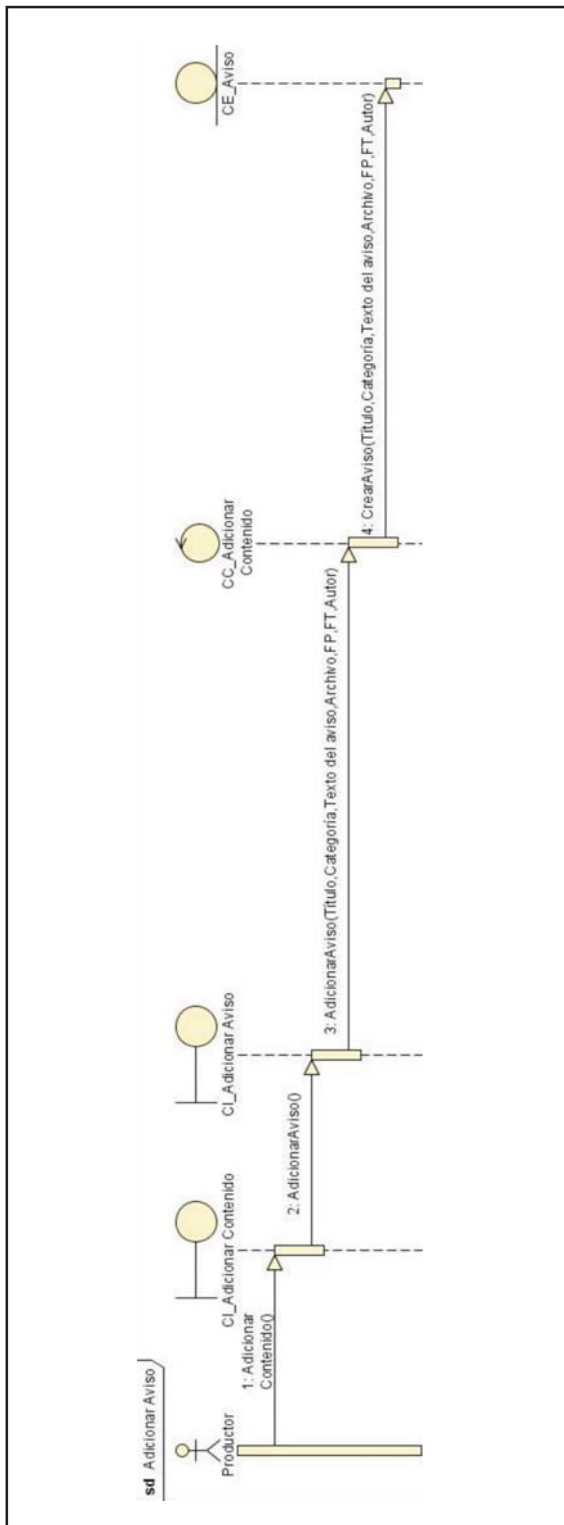


FIG. 29 DIAGRAMA DE SECUENCIA DEL CASO DE USO «ADICIONAR CONTENIDO».SECCIÓN «ADICIONAR AVISO».

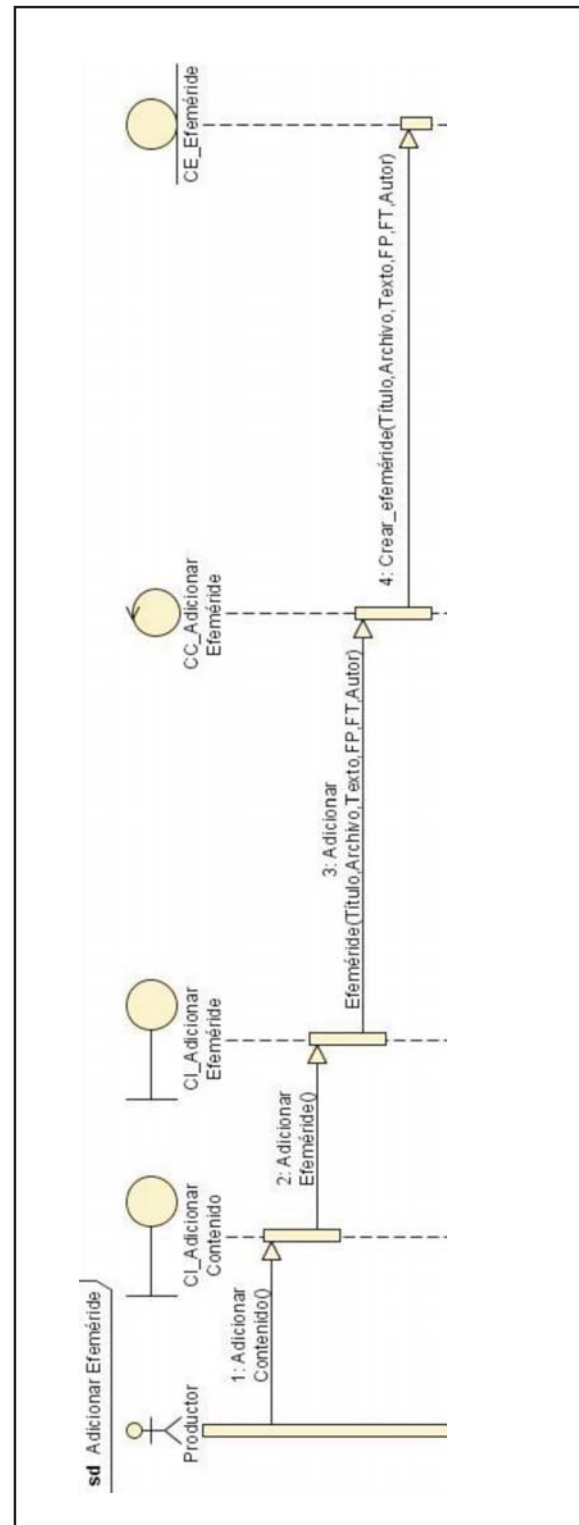


FIG. 30 DIAGRAMA DE SECUENCIA DEL CASO DE USO «ADICIONAR CONTENIDO».SECCIÓN «ADICIONAR EFEMÉRIDE».

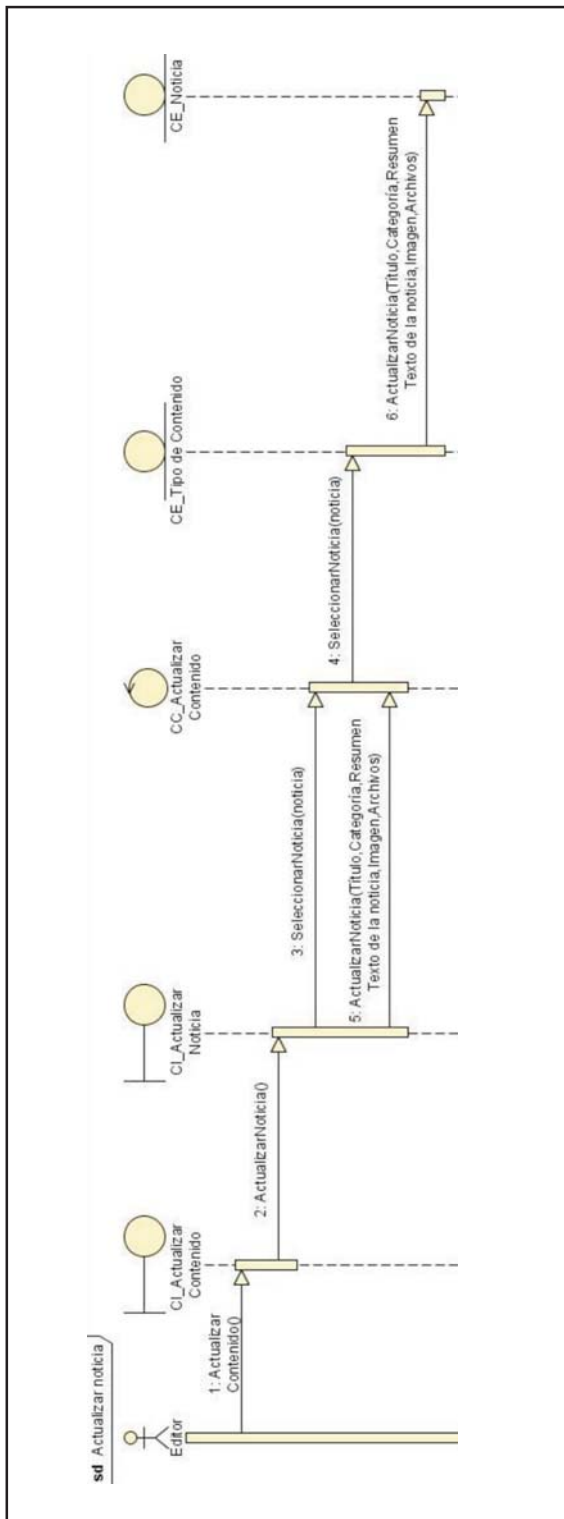


FIG. 31 DIAGRAMA DE SECUENCIA DEL CASO DE USO «ACTUALIZAR CONTENIDO».SECCIÓN «ACTUALIZAR NOTICIA».

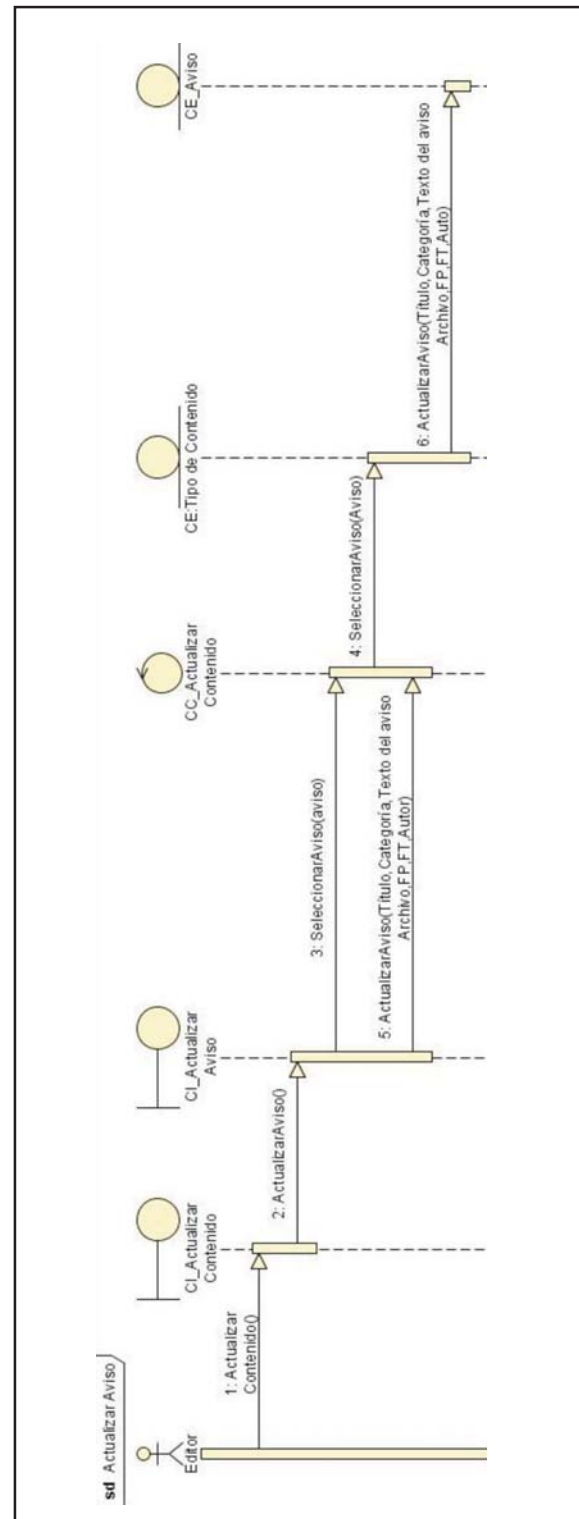


FIG. 32 DIAGRAMA DE SECUENCIA DEL CASO DE USO «ACTUALIZAR CONTENIDO».SECCIÓN «ACTUALIZAR AVISO».

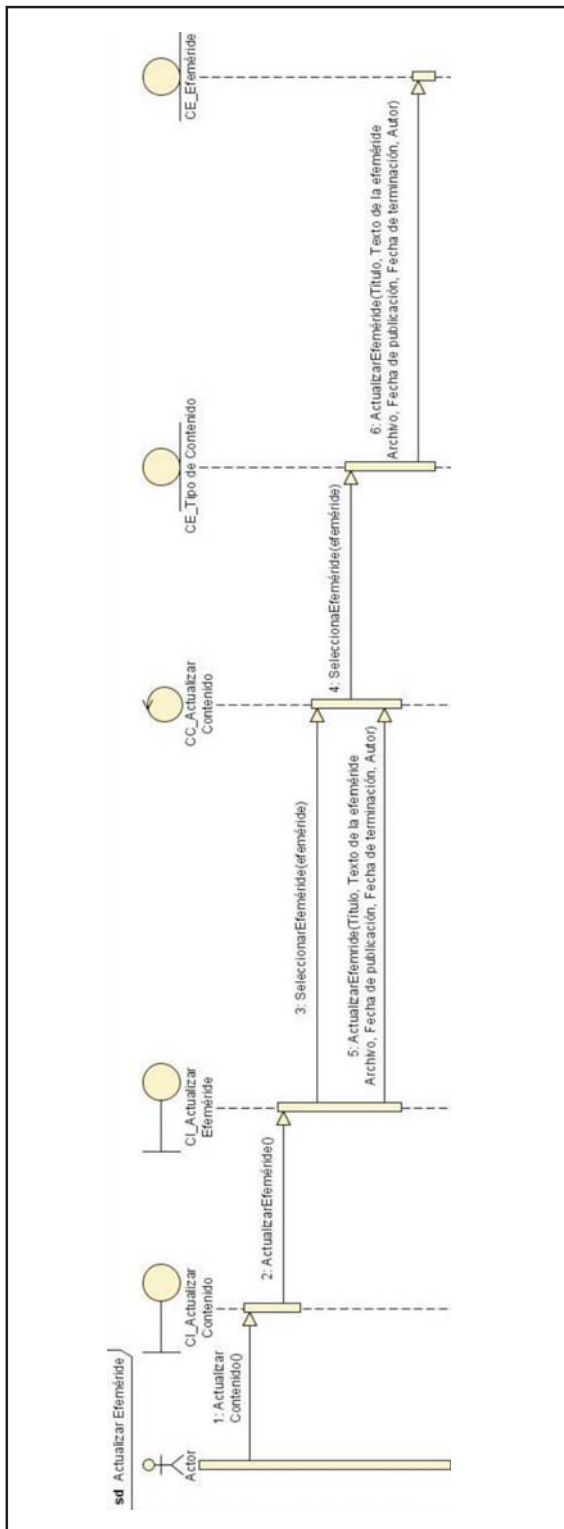


FIG. 33 DIAGRAMA DE SECUENCIA DEL CASO DE USO «ACTUALIZAR CONTENIDO».SECCIÓN «ACTUALIZAR EFEMÉRIDE».

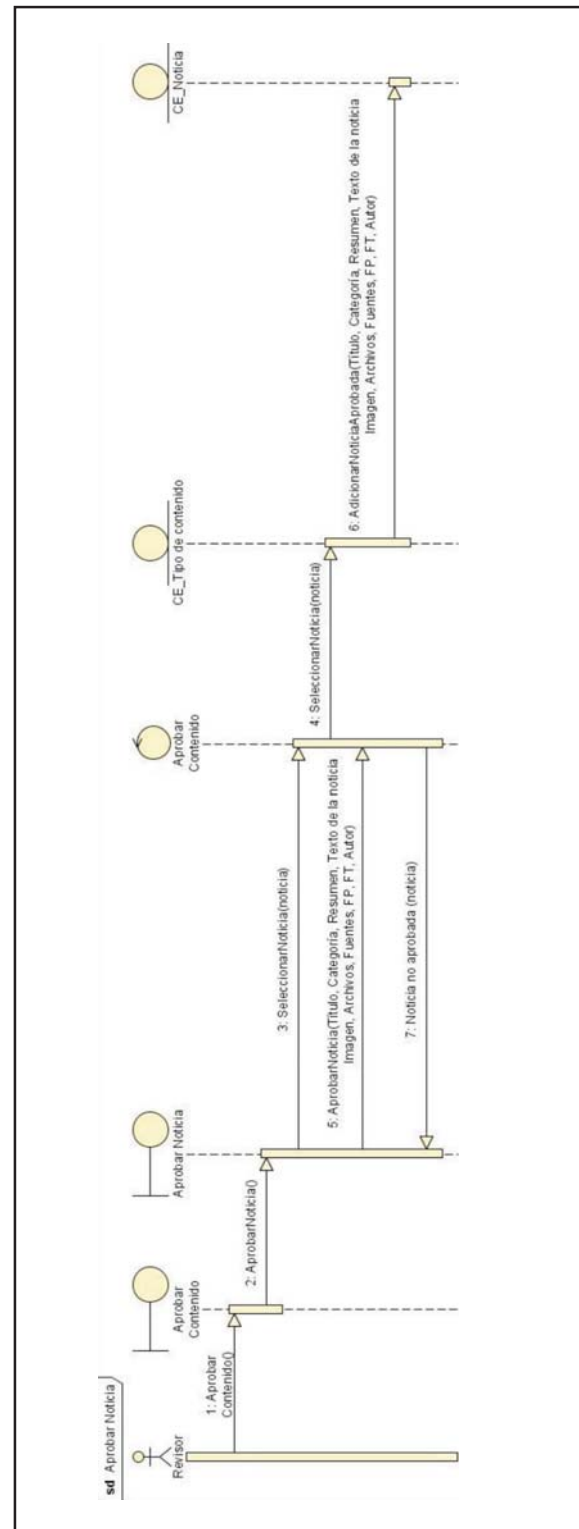


FIG. 34 DIAGRAMA DE SECUENCIA DEL CASO DE USO «APROBAR CONTENIDO».SECCIÓN «APROBAR NOTICIA».

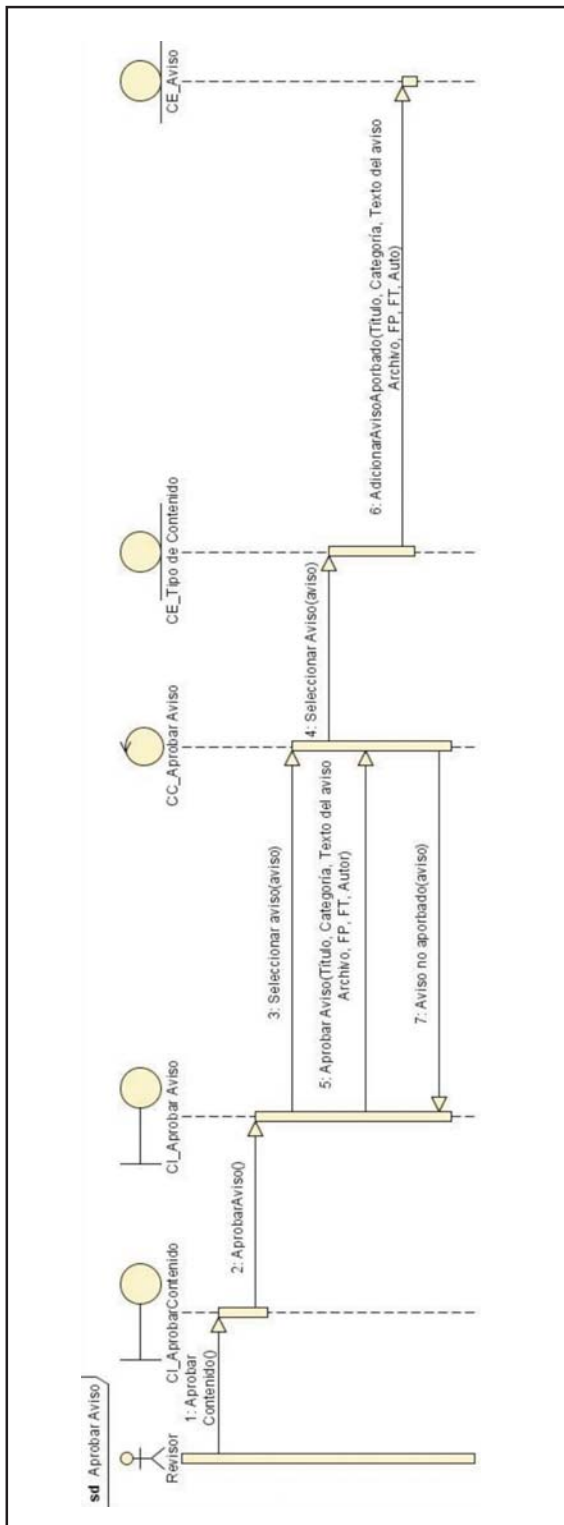


FIG. 35 DIAGRAMA DE SECUENCIA DEL CASO DE USO «APROBAR CONTENIDO».SECCIÓN «APROBAR AVISO».

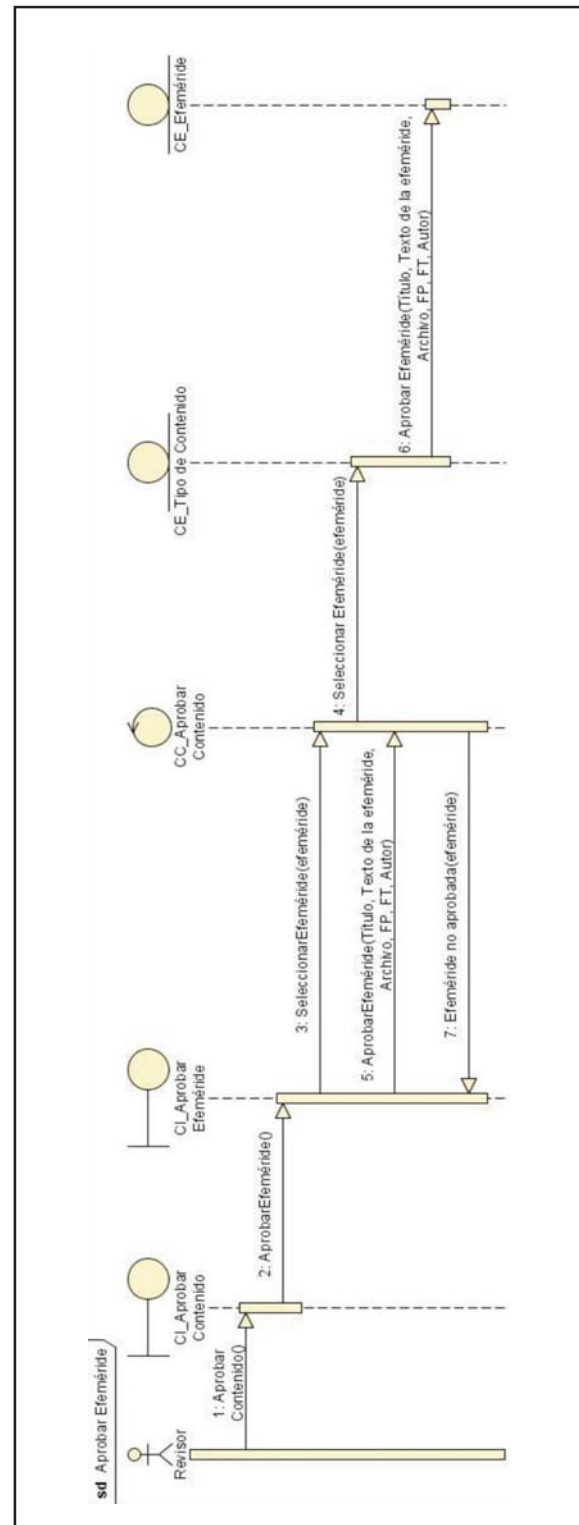


FIG. 36 DIAGRAMA DE SECUENCIA DEL CASO DE USO «APROBAR CONTENIDO».SECCIÓN «APROBAR EFEMÉRIDE».

ANEXO 4 DIAGRAMAS DE CLASES WEB

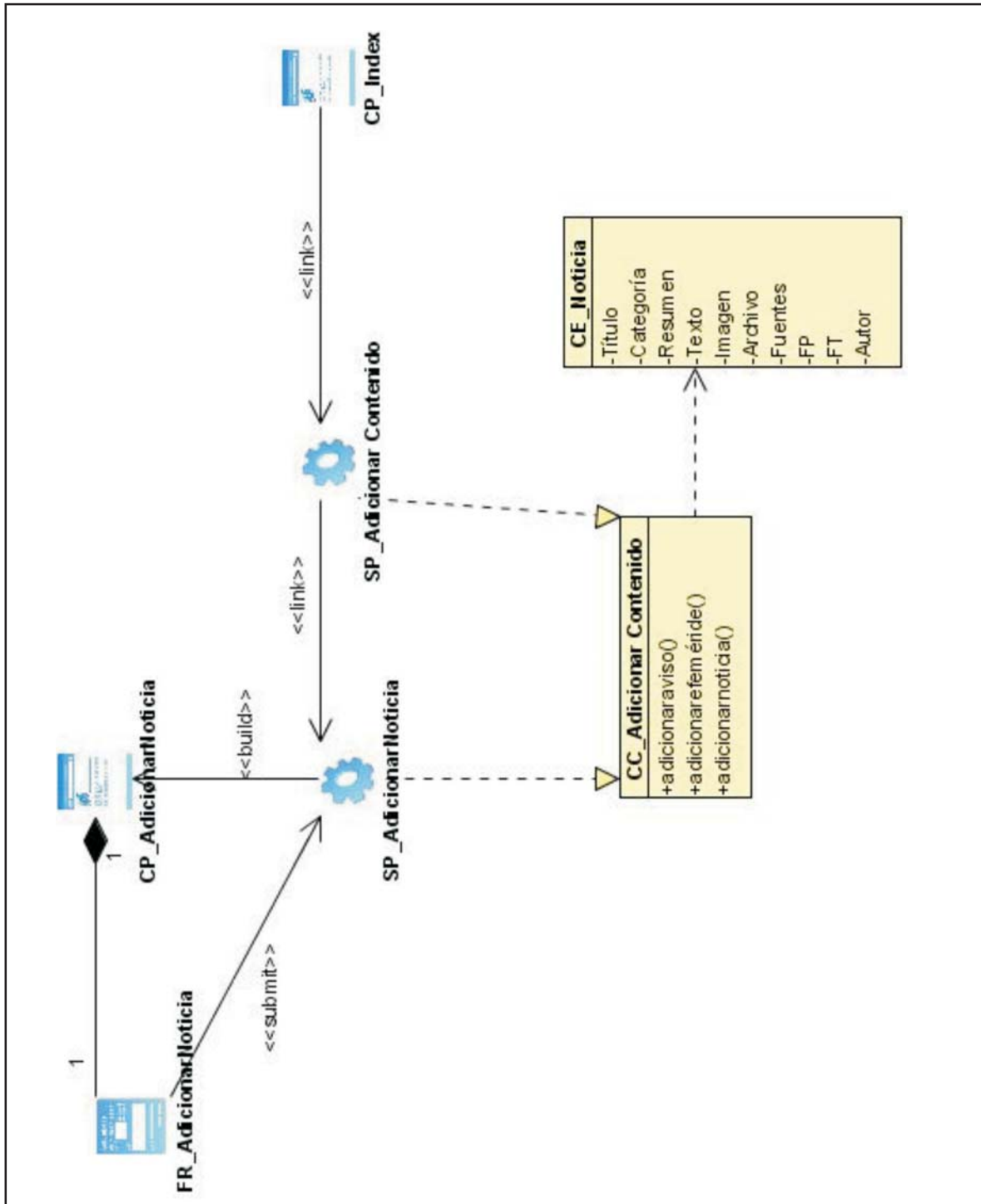


FIG. 37 DIAGRAMA DE CLASES DE DISEÑO WEB CU «ADICIONAR CONTENIDO».SECCIÓN «ADICIONAR NOTICIA».

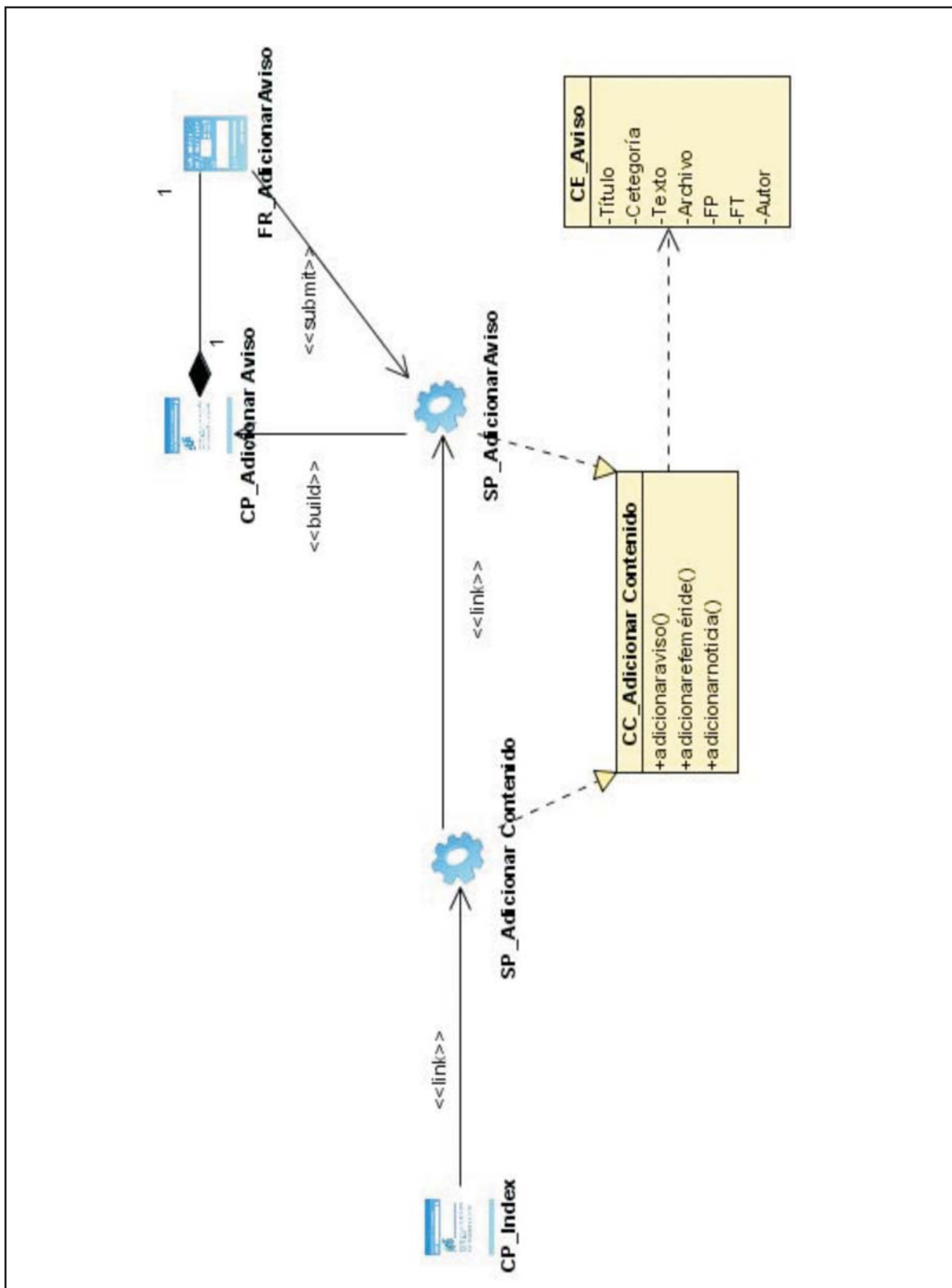


FIG. 38 DIAGRAMA DE CLASES DE DISEÑO WEB CU «ADICIONAR CONTEÚDO».SECCIÓN «ADICIONAR AVISO».

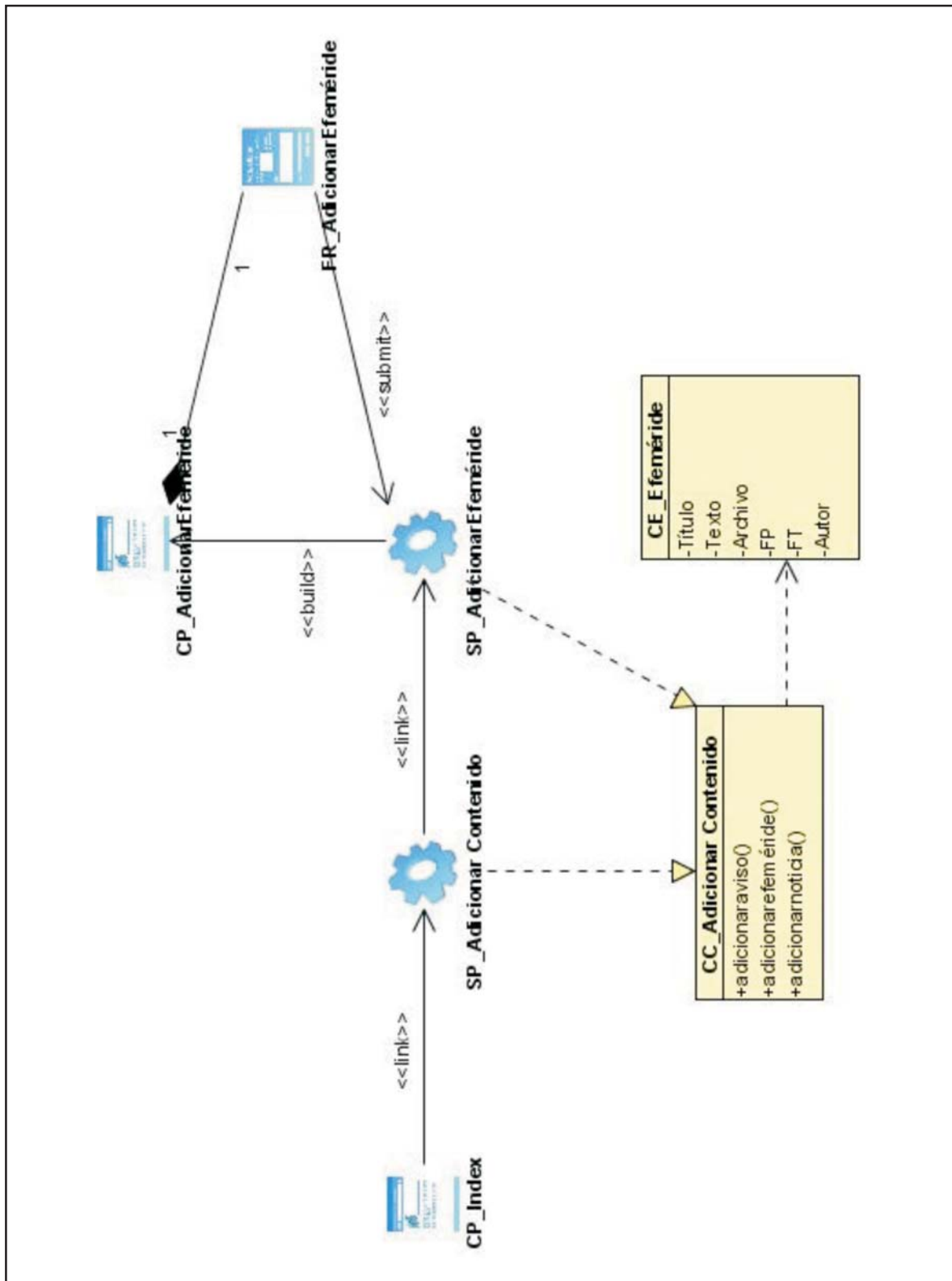


FIG. 39 DIAGRAMA DE CLASES DE DISEÑO WEB CU «ADICIONAR CONTENIDO».SECCIÓN «ADICIONAR EFEMÉRIDE».

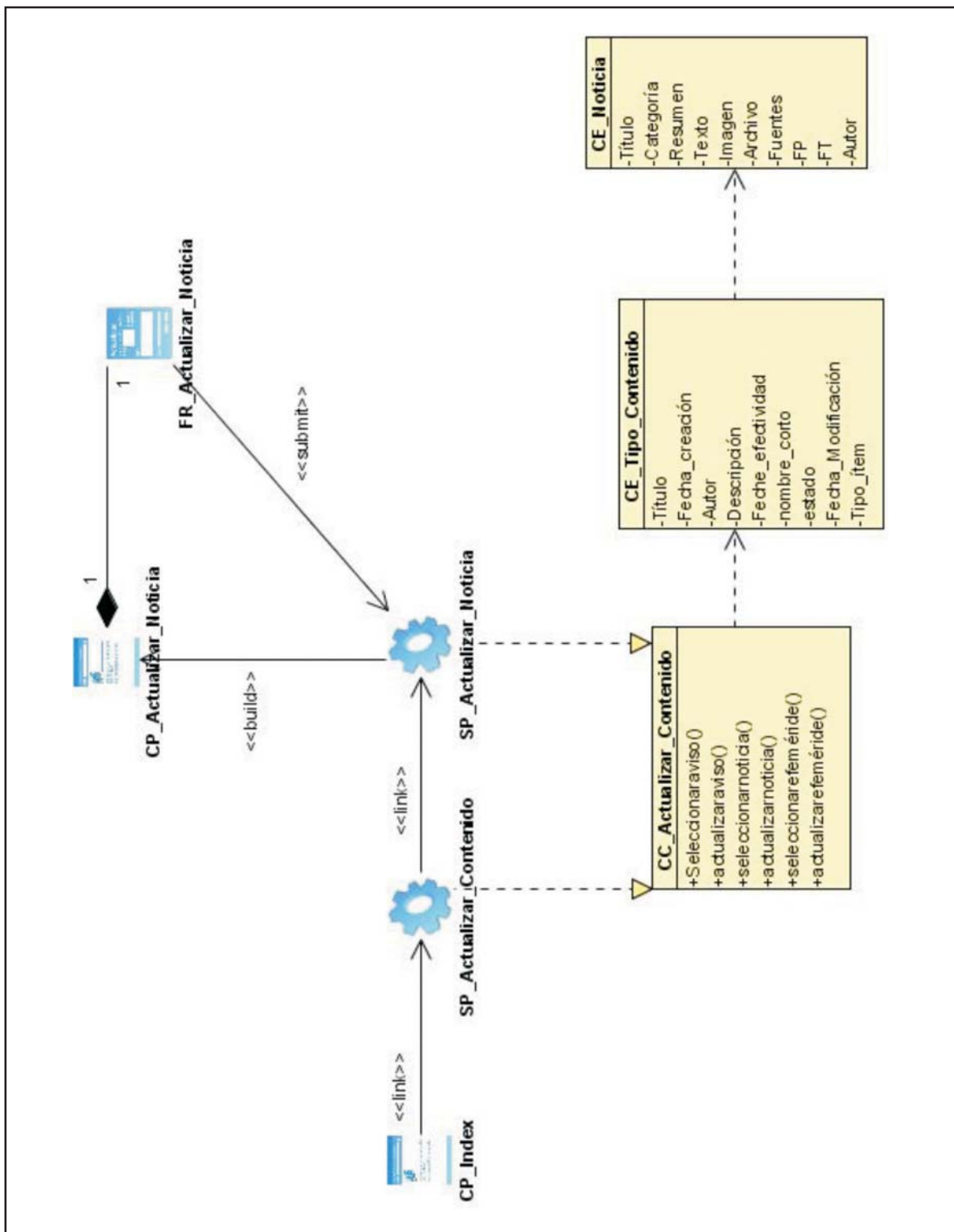


FIG. 40 DIAGRAMA DE CLASES DE DISEÑO WEB CU «ACTUALIZAR CONTENIDO».SECCIÓN «ACTUALIZAR NOTICIA».

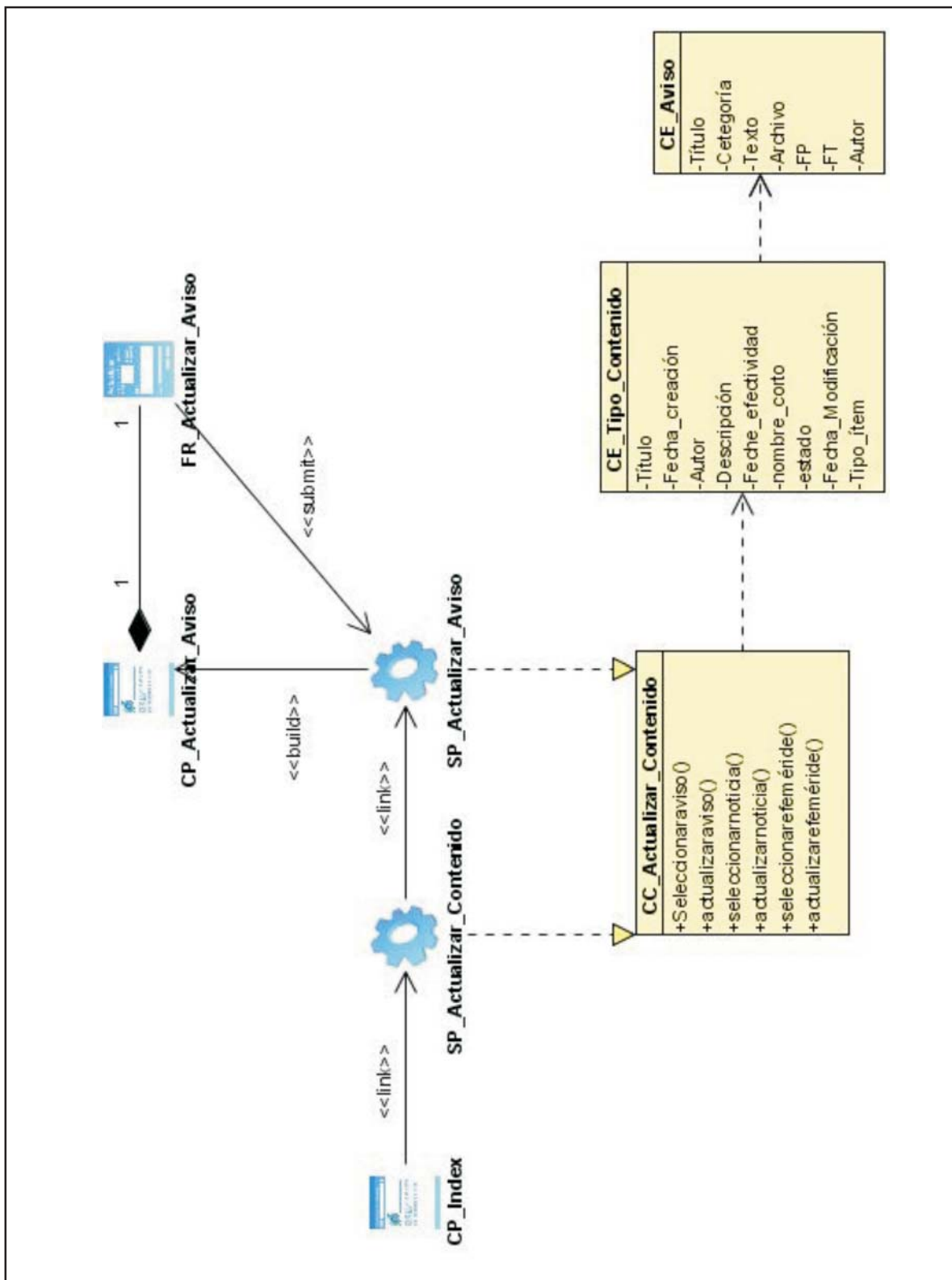


FIG. 41 DIAGRAMA DE CLASES DE DISEÑO WEB CU «ACTUALIZAR CONTENIDO».SECCIÓN «ACTUALIZARAVISO».

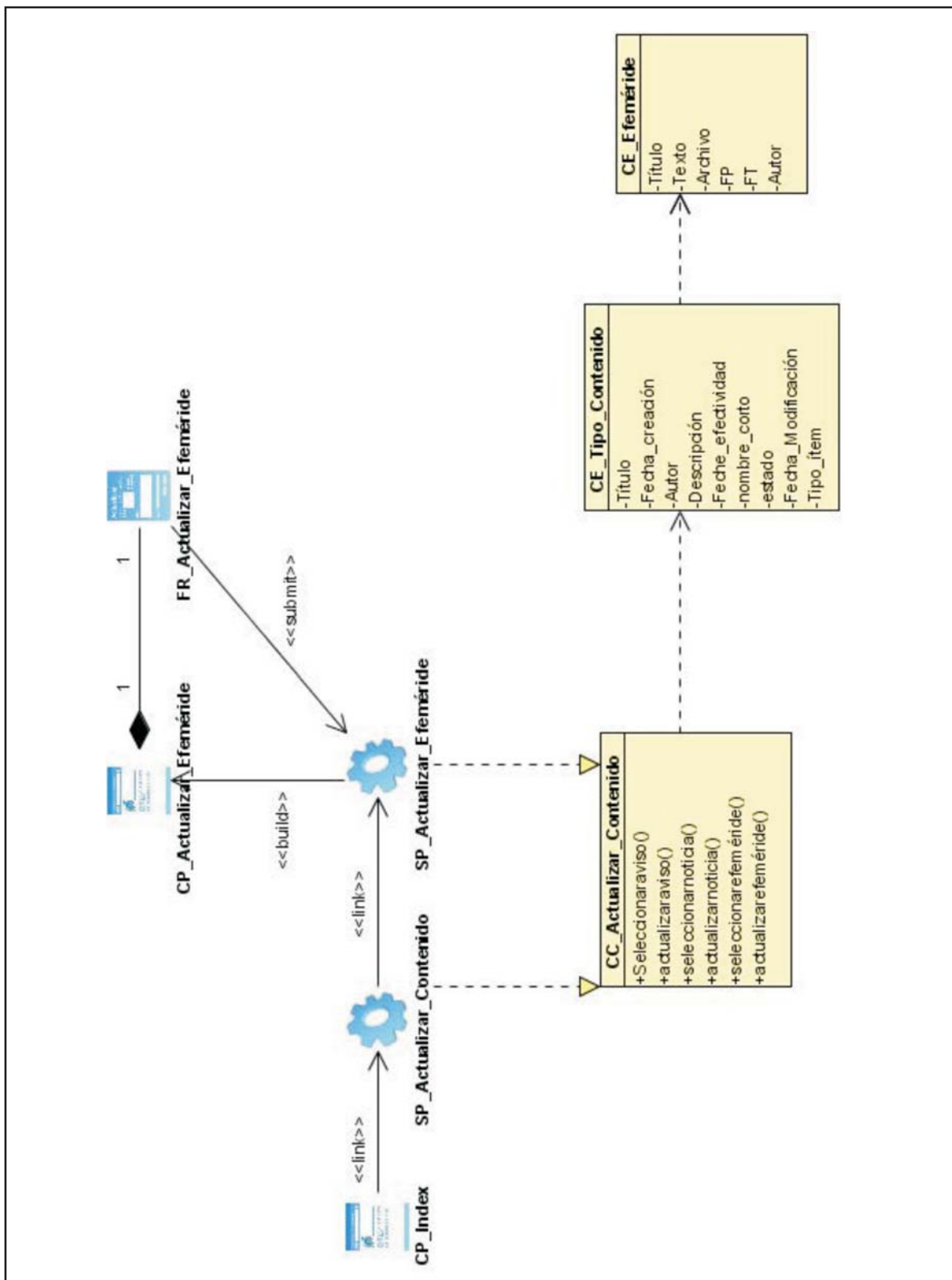


FIG. 42 DIAGRAMA DE CLASES DE DISEÑO WEB CU «ACTUALIZAR CONTENIDO».SECCIÓN «ACTUALIZAR EFEMÉRIDE».

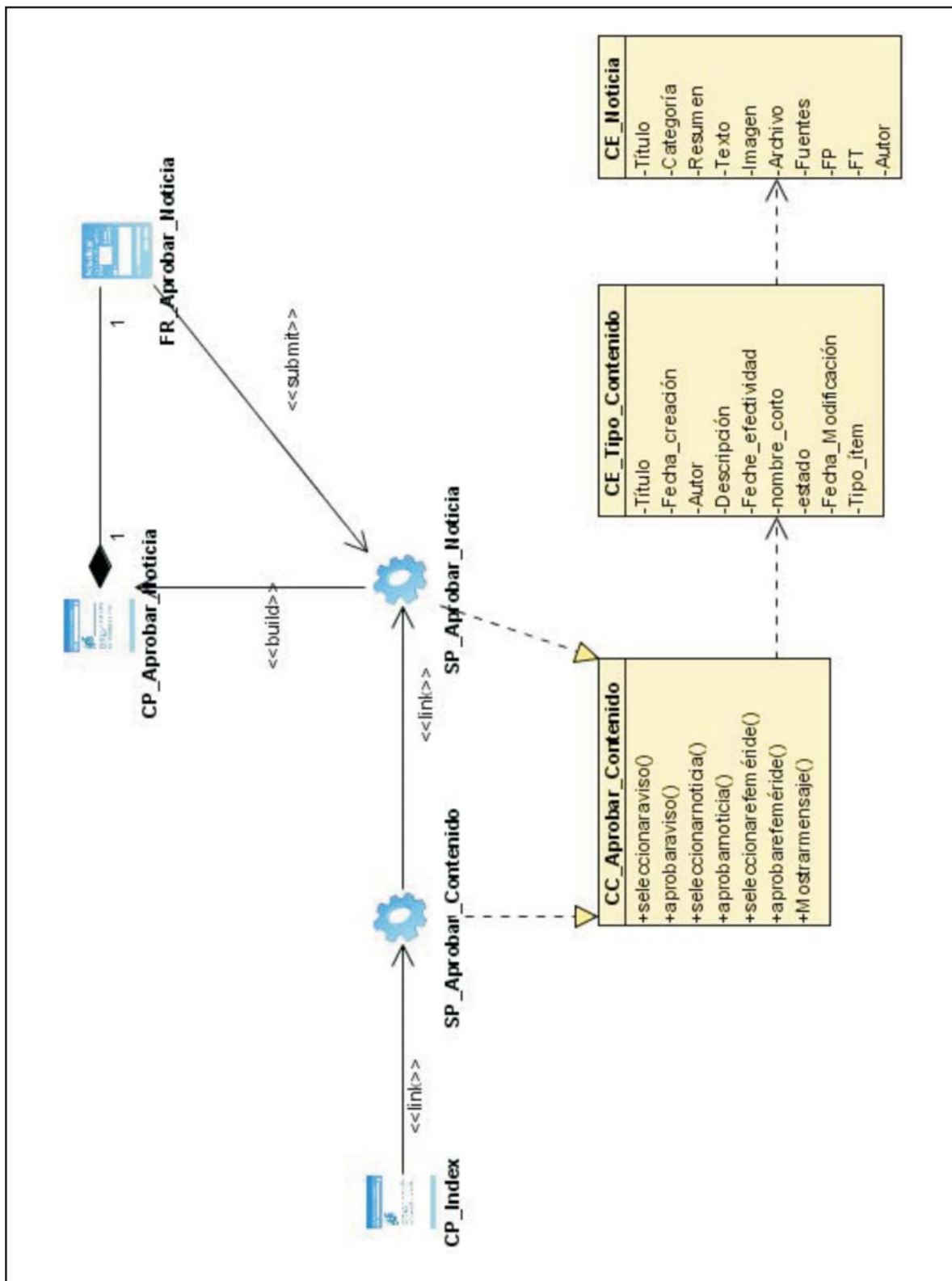


FIG. 43 DIAGRAMA DE CLASES DE DISEÑO WEB CU «APROBAR CONTENIDO».SECCIÓN «APROBAR NOTICIA».

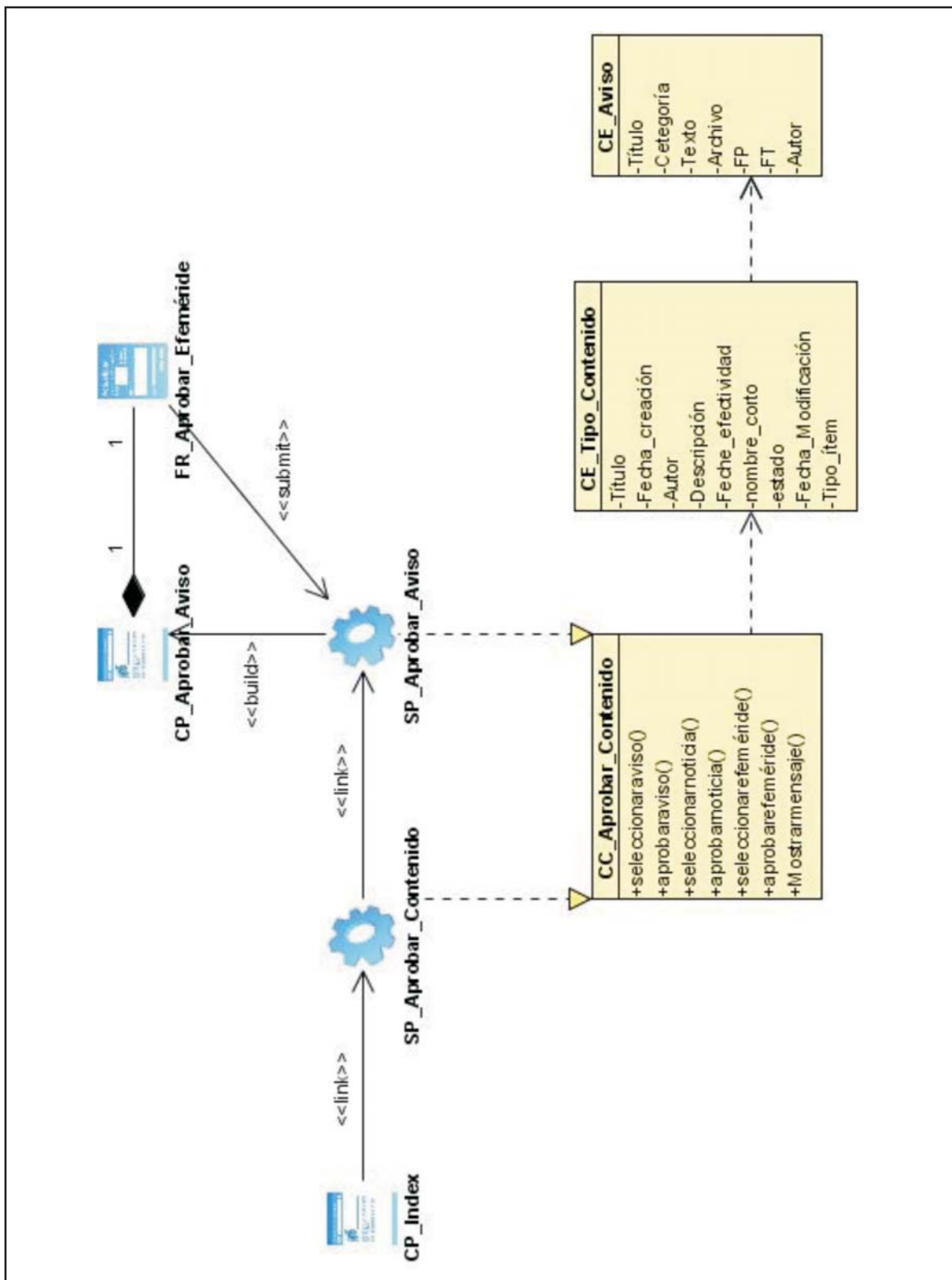


FIG. 44 DIAGRAMA DE CLASES DE DISEÑO WEB CU «APROBAR CONTENIDO».SECCIÓN «APROBAR AVISO».

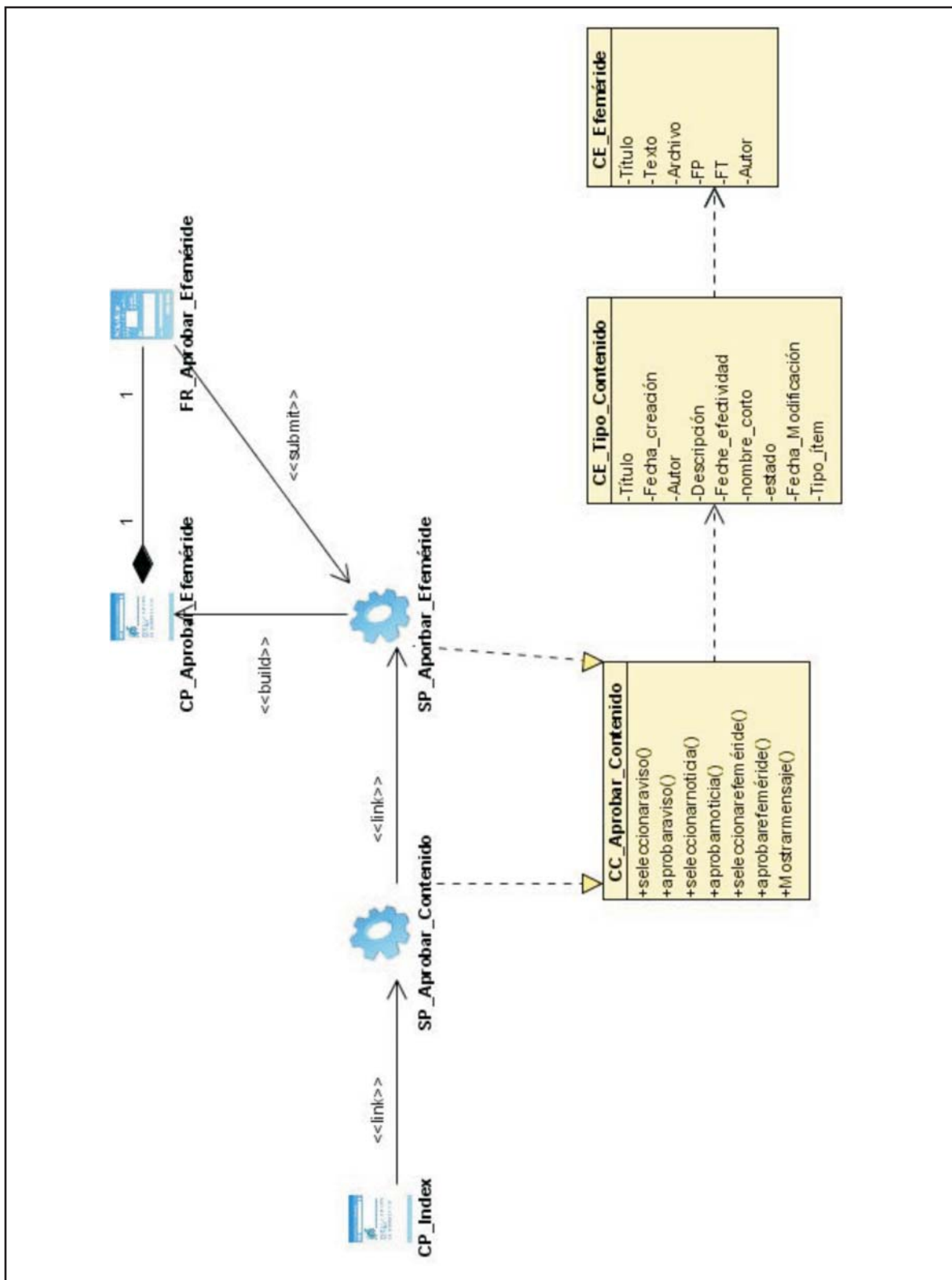


FIG. 45 DIAGRAMA DE CLASES DE DISEÑO WEB CU «APROBAR CONTENIDO».SECCIÓN «APROBAR EFEMÉRIDE».

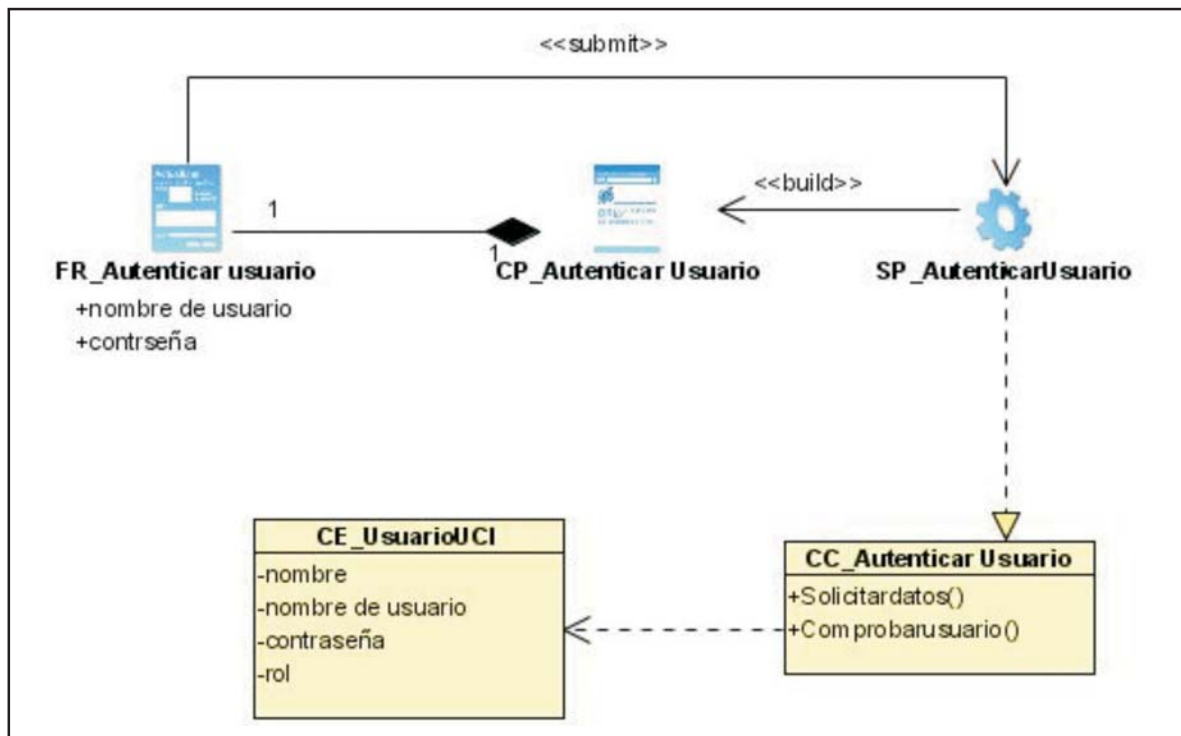
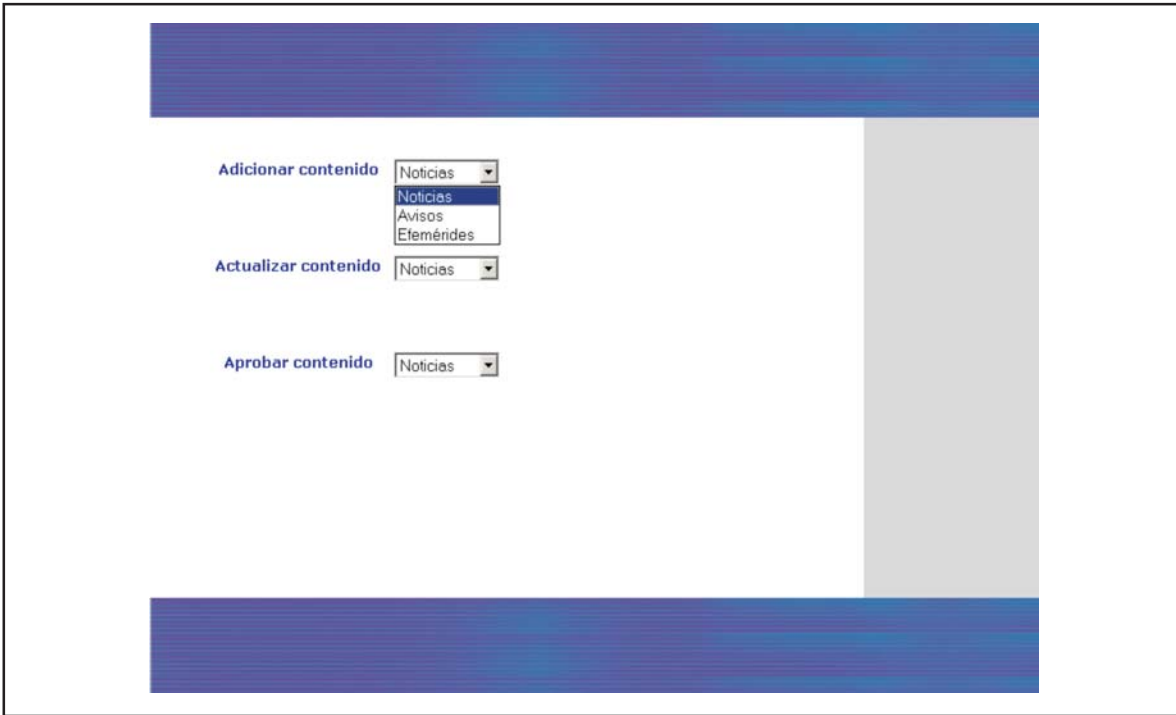


FIG. 46 DIAGRAMA DE CLASES DE DISEÑO WEB CU «AUTENTICAR USUARIO».

ANEXO 5

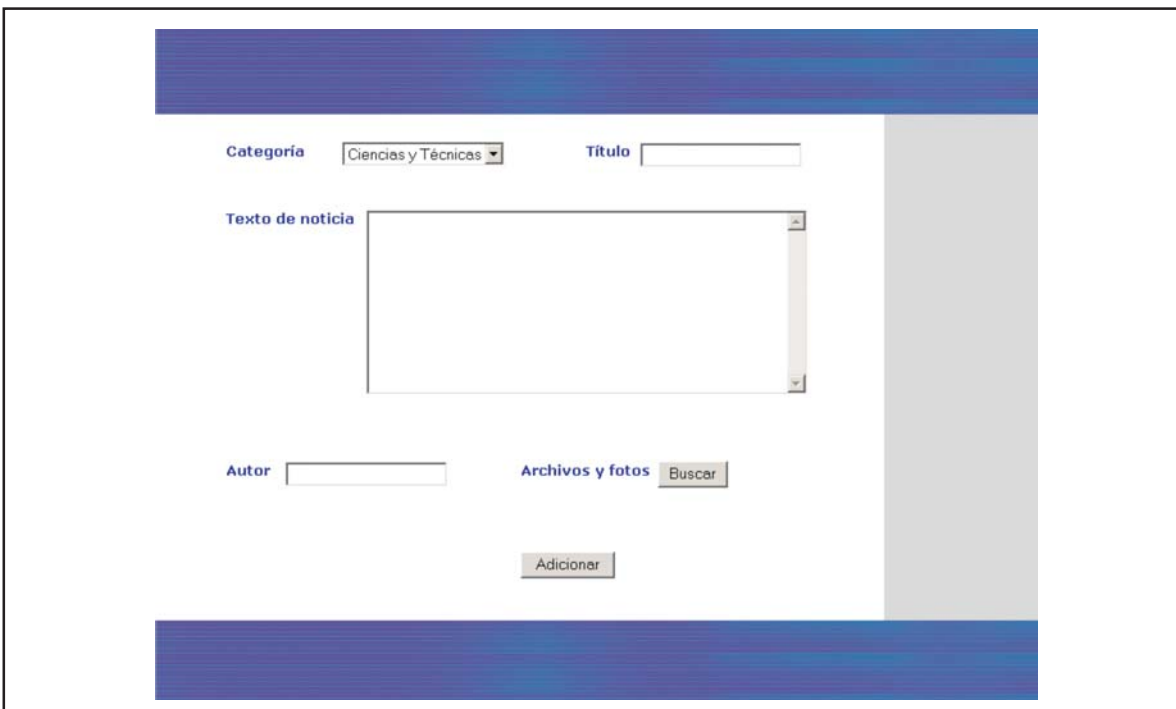


The screenshot displays a web interface with a blue header and footer. The main content area is white. On the left, there are three sections for content management:

- Adicionar contenido**: A dropdown menu with 'Noticias' selected, and a list showing 'Noticias', 'Avisos', and 'Efemérides'.
- Actualizar contenido**: A dropdown menu with 'Noticias' selected.
- Aprobar contenido**: A dropdown menu with 'Noticias' selected.

A vertical grey bar is visible on the right side of the interface.

FIG. 47 INTERFAZ COMPLETA



The screenshot shows a form titled 'ADICIONAR NOTICIA' within a blue header and footer. The form fields include:

- Categoría**: A dropdown menu with 'Ciencias y Técnicas' selected.
- Título**: A text input field.
- Texto de noticia**: A large text area for the news content.
- Autor**: A text input field.
- Archivos y fotos**: A section with a 'Buscar' button.
- Adicionar**: A button at the bottom of the form.

A vertical grey bar is visible on the right side of the interface.

FIG. 48 ADICIONAR NOTICIA

This screenshot shows a web form for updating a news item. The form is set against a white background with blue horizontal bars at the top and bottom. It includes the following fields and controls:

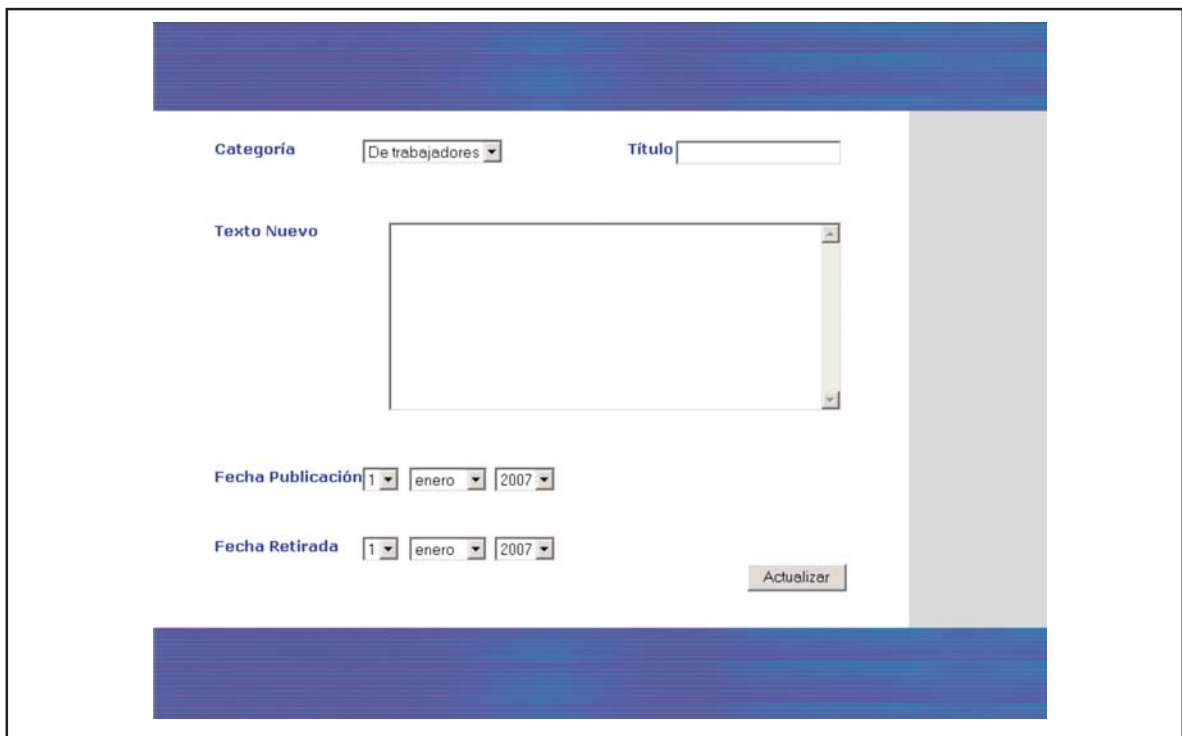
- Categoría:** A dropdown menu currently showing 'Ciencias y Técnicas'.
- Título:** A text input field.
- Texto nuevo:** A large text area for the main content.
- Autor:** A text input field.
- Archivos y fotos:** A section with a 'Buscar' button.
- Fecha Publicación:** Three dropdown menus for day (1), month (enero), and year (2007).
- Fecha Retirada:** Three dropdown menus for day (1), month (enero), and year (2007).
- Actualizar:** A button to save the changes.

FIG. 49 ACTUALIZAR NOTICIA

This screenshot shows a web form for adding a new notice. The form is set against a white background with blue horizontal bars at the top and bottom. It includes the following fields and controls:

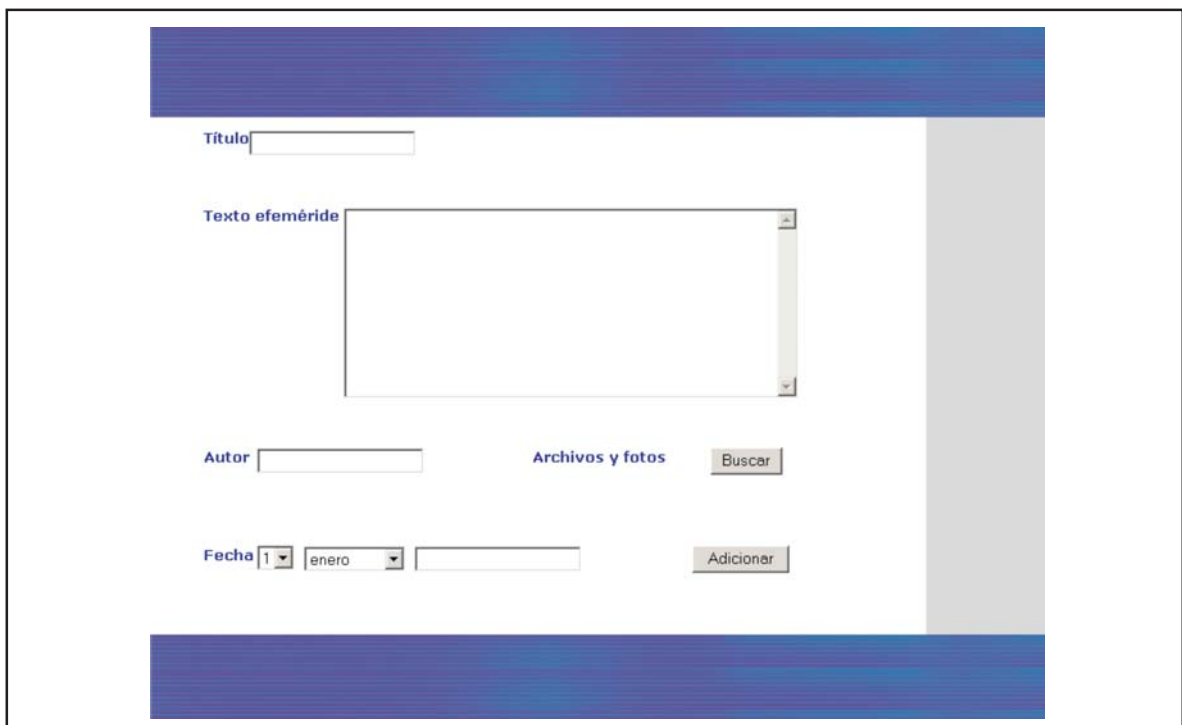
- Categoría:** A dropdown menu with a list of options: 'De trabajadores', 'De trabajadores', 'De estudiantes', and 'Relevantes'. The first 'De trabajadores' option is highlighted.
- Título:** A text input field.
- Texto de Aviso:** A large text area for the notice content.
- Autor:** A text input field.
- Área solicitante:** A text input field.
- Adicionar:** A button to submit the new notice.

FIG. 50 ADICIONAR AVISO



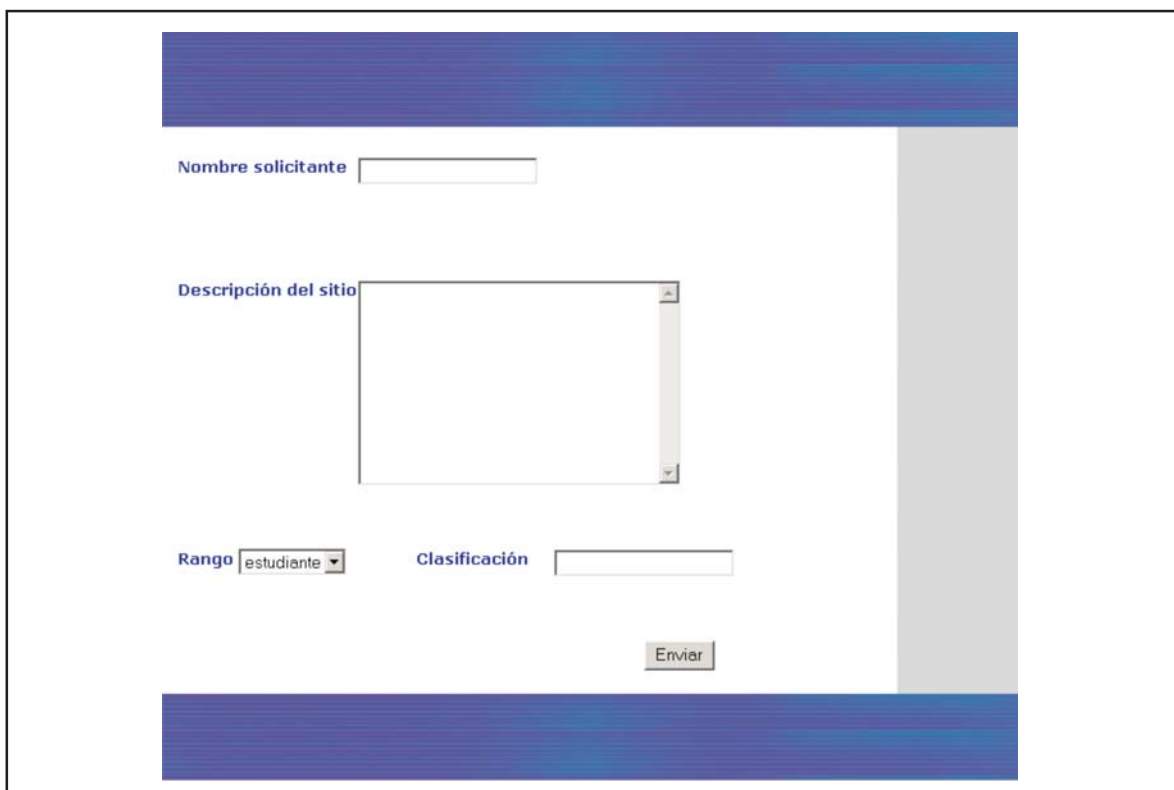
The screenshot shows a web form titled "ACTUALIZAR AVISO". It features a blue header bar at the top. Below the header, there are two main sections. The first section contains a "Categoría" dropdown menu with "De trabajadores" selected, and a "Título" text input field. The second section contains a "Texto Nuevo" label next to a large text area. Below these, there are two date pickers: "Fecha Publicación" and "Fecha Retirada", both set to "1" for the day, "enero" for the month, and "2007" for the year. A "Actualizar" button is located at the bottom right of the form.

FIG. 51 ACTUALIZAR AVISO



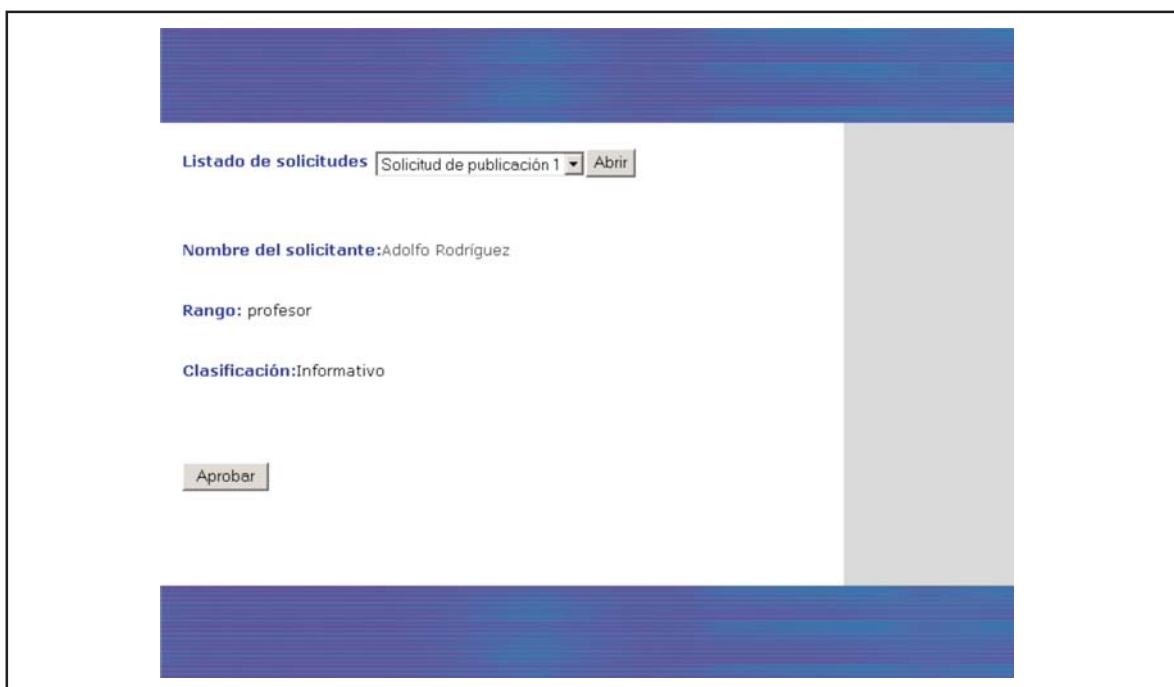
The screenshot shows a web form titled "ADICIONAR EFEMÉRIDE". It features a blue header bar at the top. Below the header, there are two main sections. The first section contains a "Título" text input field. The second section contains a "Texto efeméride" label next to a large text area. Below these, there are two input fields: "Autor" and "Archivos y fotos". A "Buscar" button is located to the right of the "Archivos y fotos" field. At the bottom, there is a "Fecha" date picker set to "1" for the day, "enero" for the month, and an empty field for the year. An "Adicionar" button is located to the right of the "Fecha" field.

FIG. 52 ADICIONAR EFEMÉRIDE



The screenshot shows a web form titled "SOLICITUD DE PUBLICACIÓN". It features a blue header bar at the top and a grey vertical sidebar on the right. The form fields include: "Nombre solicitante" with a text input box; "Descripción del sitio" with a larger text area; "Rango" with a dropdown menu showing "estudiante"; and "Clasificación" with a text input box. An "Enviar" button is located at the bottom right of the form area.

FIG. 53 SOLICITUD DE PUBLICACIÓN



The screenshot shows a web form titled "APROBAR SOLICITUD". It features a blue header bar at the top and a grey vertical sidebar on the right. The form displays the following information: "Listado de solicitudes" with a dropdown menu showing "Solicitud de publicación 1" and an "Abrir" button; "Nombre del solicitante: Adolfo Rodríguez"; "Rango: profesor"; and "Clasificación: Informativo". An "Aprobar" button is located at the bottom left of the form area.

FIG. 54 APROBAR SOLICITUD

Listado de contenido

Noticias por categorías

Avisos por categorías

Efemérides por fechas

FIG. 55 APROBAR CONTENIDO

GLOSARIO.

.....

WWW: La World Wide Web, la Web o WWW, es un sistema de navegador web para extraer elementos de información llamados «documentos» o «páginas web». Puede referirse a «una web» como una página, sitio o conjunto de sitios que proveen información por los medios descritos, o a «la Web», que es la enorme e interconectada red disponible prácticamente en todos los sitios de Internet.

VDSL: Very high bit-rate Digital Subscriber Line (DSL de muy alta tasa de transferencia).

Es una tecnología xDSL que proporciona una transmisión de datos hasta un límite teórico de 52 Mbit/s de bajada y 12 Mbit/s de subida sobre una simple línea de par trenzado. Se puede comparar con la HDSL (High bit-rate Digital Subscriber Line). Actualmente, el estándar VDSL utiliza hasta cuatro bandas de frecuencia diferentes, dos para la subida (del cliente hacia el proveedor) y dos para la bajada. La técnica estándar de modulación puede ser QAM/CAP (carrierless amplitude/phase) o DMT (Discrete multitone modulation), las cuales no son compatibles, pero tienen un rendimiento similar. Actualmente la más usada es DMT.

VDSL es capaz de soportar aplicaciones que requieren un alto ancho de banda como HDTV (televisión de alta definición).

Widget: Un widget es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o Widget Engine. Entre sus objetivos están los de dar fácil acceso a funciones frecuentemente usadas y proveer de información visual. Sin embargo los widgets pueden hacer todo lo que la imaginación desee e interactuar con servicios e información distribuida en Internet; pueden ser vistosos relojes en pantalla, notas, calculadoras, calendarios, agendas, juegos, ventanas con información del clima en su ciudad, etcétera.

Estándar: El término estándar, de origen inglés, tiene varios significados, originalmente, en inglés, significaba bandera; color; pancarta; especialmente nacional u otra enseña; así porta estándar (te). El significado primario moderno que le siguió fue «lo que es establecido por la autoridad, la costumbre o el consentimiento general». En este sentido se utiliza como sinónimo de norma. En tecnología y otros campos, un estándar es una especificación que regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad.

Plataformas de Servidor (Server Platforms): Un término usado a menudo como sinónimo de sistema operativo, la plataforma es el hardware o software subyacentes para un sistema, es decir, el motor que dirige el servidor.

Servidores de Aplicaciones (Application Servers): Designados a veces como un tipo de middleware (software que conecta dos aplicaciones), los servidores de aplicaciones ocupan una gran parte del territorio entre los servidores de bases de datos y el usuario, y a menudo los conectan.

Servidores de Audio/Video (Audio/Video Servers): Los servidores de Audio/Video añaden capacidades multimedia a los sitios web permitiéndoles mostrar contenido multimedia en forma de flujo continuo (streaming) desde el servidor.

Servidores de Chat (Chat Servers): Los servidores de chat permiten intercambiar información a una gran cantidad de usuarios ofreciendo la posibilidad de llevar a cabo discusiones en tiempo real.

Servidores de Fax (Fax Servers): Un servidor de fax es una solución ideal para organizaciones que tratan de reducir el uso del teléfono pero necesitan enviar documentos por fax.

Servidores FTP (FTP Servers): Uno de los servicios más antiguos de Internet, File Transfer Protocol permite mover uno o más archivos.

Servidores Groupware (Groupware Servers): Un servidor groupware es un software diseñado para permitir colaborar a los usuarios, sin importar la localización, vía Internet o vía Intranet corporativo y trabajar juntos en una atmósfera virtual.

Servidores IRC (IRC Servers): Otra opción para usuarios que buscan la discusión en tiempo real, Internet Relay Chat consiste en varias redes de servidores separadas que permiten que los usuarios conecten el uno al otro vía una red IRC.

Servidores de Listas (List Servers): Los servidores de listas ofrecen una manera mejor de manejar listas de correo electrónico, bien sean discusiones interactivas abiertas al público o listas unidireccionales de anuncios, boletines de noticias o publicidad.

Servidores de Correo (Mail Servers): Casi tan ubicuos y cruciales como los servidores web, los servidores de correo mueven y almacenan el correo electrónico a través de las redes corporativas (vía LANs y WANs) y a través de Internet.

Servidores de Noticias (News Servers): Los servidores de noticias actúan como fuente de distribución y entrega para los millares de grupos de noticias públicos actualmente accesibles a través de la red de noticias USENET.

Servidores Proxy (Proxy Servers): Los servidores proxy se sitúan entre un programa del cliente (típicamente un navegador) y un servidor externo (típicamente otro servidor web) para filtrar peticiones, mejorar el funcionamiento y compartir conexiones.

Servidores Telnet (Telnet Servers): Un servidor telnet permite a los usuarios entrar en un ordenador huésped y realizar tareas como si estuviera trabajando directamente en ese ordenador.

Servidores Web (Web Servers): Básicamente, un servidor web sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario.