



Universidad Nacional Autónoma de México

Facultad de Ingeniería

**Sistema de Información para Proyectos que
Generan Ingresos Extraordinarios**

TESIS

Que para obtener el título de Ingeniero en Computación

Presenta:

Acosta Soto Ubaldo

Gamiño García Marcela Gerarda

Director de Tesis:

Ing. Germán Santos Jaimes



México, D.F.

2004

Agradecimientos

*A mis padres por todo el amor
y apoyo brindados durante mi vida personal y académica.
A mi madre Alicia García Méndez por su dedicación.
A mi padre Estanislao Gamiño Durán
por su comprensión y paciencia.
A los maestros que han contribuido a mi desarrollo profesional.
A Ubaldo Acosta Soto por su apoyo incondicional,
toda su paciencia y su amor, por enseñarme que siempre
hay que seguir adelante, aún cuando
hubo momentos en los que casi me rendí.
A mis verdaderos amigos y
todas aquellas personas que contribuyeron en mi vida
con sus consejos y enseñanzas.*

MARCELA G. GAMIÑO GARCÍA

*Cuando creemos que hemos
llegado al final del camino,
éste apenas comienza.*

*A mis padres,
Aleja Soto Soto y
Ángel Acosta Vargas,
y a mis hermanos Luis Acosta Soto
y Miguel Ángel Acosta Soto
por todo su apoyo, amor y comprensión
a lo largo de mi vida.
A todas las personas que han participado
en mi desarrollo personal y profesional.
Y por supuesto a Marcela G. Gamiño García
que me ha dado su apoyo,
comprensión y todo su amor,
y es con quien espero estar el resto de mi vida.*

UBALDO ACOSTA SOTO

Tabla de Contenido

Introducción	1
Capítulo I. Unidad de Administración de Proyectos de Ingresos Extraordinarios	3
1.1 Introducción	4
1.2 Datos Generales de la Unidad de Administración de Proyectos de Ingresos Extraordinarios "UAPIEX"	4
1.2.1 Conceptos generales de Ingresos Extraordinarios	6
1.2.2 Instancias proveedoras de información de los proyectos	6
1.2.3 Actividades principales de la UAPIEX.....	8
1.3 Proceso actual del manejo de los proyectos en la UAPIEX	8
1.3.1 Etapas de los proyectos que generan ingresos extraordinarios.....	9
1.3.1.1 Etapa de promoción	9
1.3.1.2 Etapa de operación	12
1.3.1.3 Etapa de memoria tecnológica	14
Capítulo II. Metodologías Orientadas a Objetos.....	15
2.1 Introducción	16
2.2 Conceptos Fundamentales de Metodologías Orientadas a Objetos	16
2.2.1 Objetos	17
2.2.2 Clases.....	17
2.2.3 Comunicación entre objetos	17
2.2.4 Herencia.....	18
2.2.5 Polimorfismo	18
2.2.6 Sobrecarga	18
2.2.7 Abstracción.....	19
2.2.8 Encapsulamiento	19
2.2.9 Modularidad	19
2.2.10 Jerarquía	19
2.3 Metodología orientada a objetos "Booch"	19
2.3.1 Macroproceso.....	20
2.3.2 Microproceso.....	20
2.3.3 Notación Booch	21
2.3.3.1 Objetos	21
2.3.3.2 Clases	21
2.3.3.3 Relaciones	23
2.4 Metodología orientada a objetos "OMT".....	23
2.4.1 Fases de la metodología OMT	23
2.4.1.1 Análisis del sistema	23
2.4.1.2 Diseño del sistema	24
2.4.1.3 Diseño de objetos	24
2.4.2 Notación OMT.....	24
2.4.2.1 Objetos	25
2.4.2.2 Clases	25
2.4.2.3 Asociaciones	25
2.5 Metodología orientada a objetos "UML".....	26
2.5.1 Notación UML.....	26
2.5.1.1 Objetos	26
2.5.1.2 Clases	27
2.5.1.3 Relaciones	27
2.5.1.3.1 Relación de asociación	27

2.5.1.3.2	Relación de generalización	28
2.5.1.3.3	Relación de agregación y composición	29
2.5.1.3.4	Relación de realización	30
2.5.1.3.5	Relación de dependencia de uso	30
2.5.2	Vistas de UML	31
2.5.2.1	Vista estática	32
2.5.2.2	Vista de casos de uso	32
2.5.2.2.1	Relaciones de casos de uso	33
2.5.2.3	Vista de implantación y vista de despliegue	34
2.5.2.3.1	Componente	34
2.5.2.3.2	Nodo	35
2.5.2.4	Vista de máquina de estados.	36
2.5.2.5	Vista de interacción	37
2.5.2.5.1	Diagrama de secuencia	38
2.5.2.5.2	Diagrama de colaboración	39
2.5.2.6	Vista de actividades.....	40
2.5.2.6.1	Diagrama de actividades	40
2.5.2.7	Vista de gestión del modelo.....	41
Capítulo III. Software Libre y Propietario.....		43
3.1	Introducción	44
3.2	Ventajas y desventajas de utilizar software propietario	44
3.2.1	Ventajas del software propietario	44
3.2.2	Desventajas del software propietario	45
3.3	Ventajas y desventajas de utilizar software libre	45
3.3.1	Ventajas del software libre	46
3.3.2	Desventajas del software libre	46
3.4	Desarrollo en México.....	47
3.4.1	Perfil para el país.....	47
Capítulo IV. Redes de Computadoras e Internet.....		49
4.1	Introducción	50
4.2	Conceptos Básicos	50
4.3	Topologías de redes de computadoras	52
4.3.1	Topología de Estrella	52
4.3.2	Topología de Bus	53
4.3.3	Topología de Anillo.....	53
4.4	Tipos de cables	54
4.4.1	Cable coaxial.....	54
4.4.2	UTP (Unshielded Twisted Pair)	54
4.4.3	FTP (Foil Screen Twisted Pair).....	55
4.4.4	STP (Shielded Twisted Pair).....	55
4.4.5	Fibra Óptica	55
4.5	Tecnologías de Interfaces de Red.....	56
4.5.1	Ethernet.....	56
4.5.2	Fast Ethernet	56
4.5.3	Giga Ethernet.....	56
4.5.4	Token-Ring	57
4.5.5	FDDI (Fiber Distributed Data Interface).....	57
4.5.6	Frame Relay	57
4.5.7	ATM (Asynchronous Transfer Mode).....	58
4.6	Elementos de Conectividad	58
4.6.1	Puentes.....	59
4.6.2	Routers	59
4.6.3	Gateways	60

4.6.4	Concentradores (Hub's)	60
4.6.5	Switches	60
4.7	Modelo ISO/OSI	60
4.7.1	Nivel uno: Físico	61
4.7.2	Nivel dos: Enlace	62
4.7.3	Nivel tres: Red	62
4.7.4	Nivel cuatro: Transporte	62
4.7.5	Nivel cinco: Sesión	62
4.7.6	Nivel seis: Presentación	63
4.7.7	Nivel siete: Aplicación	63
4.7.8	Familia de protocolos TCP/IP	63
4.8	Internet	64
4.8.1	Intranet	65
4.8.2	Extranet	65
4.8.3	Servicios en Internet	65
4.8.3.1	Correo Electrónico (e-mail)	65
4.8.3.2	Conexión Remota (Telnet)	65
4.8.3.3	Transferencia de Archivos (FTP)	66
4.8.3.4	Grupos de noticias	66
4.8.3.5	World Wide Web (WWW)	66
Capítulo V. Estudio Inicial del Sistema		67
5.1	Introducción	68
5.2	Planteamiento del problema	68
5.3	Metodología de desarrollo de software	68
5.3.1	Proceso Unificado	69
5.3.1.1	Características principales	69
5.3.1.2	Modelo iterativo	70
5.3.1.3	Modelo iterativo e incremental	71
5.3.1.4	Fases del ciclo de desarrollo	72
5.4	Tecnologías para el desarrollo del software	77
5.4.1	Manejador de base de datos	77
5.4.2	Lenguaje de programación	78
5.4.3	Servidor web	80
5.4.4	Sistema operativo	80
5.5	Propuesta de arquitectura del sistema	81
5.5.1	Arquitectura preliminar del sistema	81
5.5.2	Tipos de clases a utilizar en el sistema	82
5.5.3	Acoplamiento y cohesión	83
5.5.4	Modelo multicapas	84
5.5.5	Modelo MVC	86
Capítulo VI. Análisis, Diseño y Desarrollo del Sistema		89
6.1	Introducción	90
6.2	Análisis del sistema	90
6.2.1	Actores	92
6.2.2	Tabla de eventos	93
6.2.3	Diagrama de casos de uso	94
6.2.4	Trayectorias de los casos de uso	95
6.2.4.1	Trayectoria principal	95
6.2.4.2	Trayectoria alternativa	96
6.2.4.3	Trayectoria de excepción	97
6.2.5	Incrementos	98
6.3	Diseño del sistema	100
6.3.1	Descripción de Clases	100

6.3.1.1	Clases de entidad.....	100
6.3.1.2	Clases de límite	100
6.3.1.3	Clases de control	101
6.3.1.4	Relaciones entre clases.....	101
6.3.1.5	Métodos y atributos de las clases	101
6.3.2	Diagramas de Secuencia.....	105
6.3.3	Modelo entidad-relación.....	145
6.3.3.1	Mapeo entre clases y tablas de la base de datos	145
6.3.3.2	Diagrama entidad relación.....	146
6.3.3.3	Diccionario de datos	146
6.4	Desarrollo del sistema.....	149
6.4.1	Presentación de las pantallas.....	149
6.4.2	Sitio Web de la UAPIEX.....	150
6.4.2.1	Página principal	150
6.4.2.2	Entrar al sistema	151
6.4.2.3	Memoria Tecnológica	152
6.4.2.4	Descargas.....	155
6.4.2.5	Avisos	156
6.4.3	SIPIEX	157
6.4.3.1	Menú principal	157
6.4.3.2	Proyectos en Promoción	158
6.4.3.3	Proyectos en Operación	165
6.4.3.4	Usuarios	170
6.4.3.5	Responsables o Líderes de Proyecto.....	173
6.4.3.6	Contratantes	175
6.4.3.7	Divisiones	178
6.4.3.8	Ayuda.	180
6.4.3.9	Salir.....	180
6.4.3.10	Acerca.....	180
6.5	Pruebas y resultados en el sistema	180
	Conclusiones.....	183
	Anexo A. Formatos utilizados en la UAPIEX	185
	Anexo B. Plan de Actividades.....	191
	Referencias.....	195
	Glosario	197

Introducción

La Universidad Nacional Autónoma de México es un espacio para la pluralidad de ideas y conocimientos. La docencia, la investigación y la difusión de la cultura son sus labores primordiales, las cuales se desarrollan a través de sus distintos recintos, como son facultades, institutos, centros culturales, etc.

La Facultad de Ingeniería de la UNAM es líder en la formación de ingenieros, maestros, doctores y especialistas que demanda el país, además es generadora de avances en la investigación aplicada y el desarrollo tecnológico en México.

Con el propósito de mejorar su participación en la construcción de un país más competitivo, la Facultad de Ingeniería tiene vínculos con el sector productivo del país. Como fruto de estos vínculos se desarrollan investigaciones, estudios, trabajos y/o proyectos en beneficio de las empresas o instituciones que así lo requieran.

Debido al desarrollo de estos proyectos, la Facultad recibe una remuneración económica por sus servicios y ya que estas captaciones no forman parte de su presupuesto asignado, los proyectos se denominan *proyectos que generan ingresos extraordinarios*.

La entidad encargada de la gestión de estos proyectos es la Unidad de Administración de Proyectos que Generan Ingresos Extraordinarios, la cual en su búsqueda de mejorar los procesos, se dio cuenta de la necesidad de un sistema de información debido a problemas tales como: dificultades al registrar los proyectos, falta de comunicación entre las instancias involucradas, retrasos en la obtención de la información, entre otros.

Conscientes de esta necesidad y con el objetivo de cubrirla, ofrecemos la solución desarrollada en la presente tesis, el SIPIEX (Sistema de Información para Proyectos que Generan Ingresos Extraordinarios).

Con este proyecto se espera obtener un sistema que cubra los requerimientos solicitados por la Unidad de Administración de Proyectos que Generan Ingresos Extraordinarios en el menor tiempo posible y aprovechando al máximo los recursos de hardware con los que cuenta.

El sistema permitirá registrar proyectos, conocer información relevante, oportuna y confiable, consultada vía Internet por medio del sitio Web que se desarrollará para la Unidad de Administración de Proyectos que Generan Ingresos Extraordinarios, con el propósito de ser accesible a todo usuario.

También vemos como una meta importante el desarrollo de la Memoria Tecnológica, que será parte del sitio Web y proporcionará información de interés institucional a investigadores, profesores y público interesado sobre los proyectos que se desarrollan en la Universidad, lo que puede significar el impulso de nuevos vínculos con el sector productivo y al mismo tiempo con la comunidad en general.

De esta manera esperamos obtener un control eficiente de los proyectos que generan ingresos extraordinarios. Además, consideramos que el uso del SIPIEX promoverá un incremento de la productividad y competitividad entre las instancias involucradas.

El presente trabajo está dividido en seis capítulos, los cuatro primeros son nuestro marco teórico y los dos últimos pertenecen al desarrollo del sistema.

El primer capítulo es una introducción a la Unidad de Administración de Proyectos de Ingresos Extraordinarios. En él explicamos el proceso de los proyectos que dicha unidad gestiona.

En el segundo capítulo mostramos los conceptos fundamentales relacionados con las metodologías orientadas a objetos, como son: Booch, OMT y UML. Este capítulo nos va a dar las bases para especificar, visualizar, construir y documentar nuestro sistema.

El tercer capítulo es un marco general de las tecnologías en cuanto a software se refiere. Indicamos las ventajas y desventajas de utilizar software libre o propietario, y el desarrollo del software en México.

El cuarto capítulo nos brinda las bases necesarias de las redes de computadoras e Internet. Esto es necesario debido a que el sistema a desarrollar es una aplicación Web y ésta será consultada vía Internet.

En el quinto capítulo abordamos el estudio inicial del sistema. En él mencionamos el problema a resolver, la metodología de desarrollo utilizada, las tecnologías de software manejadas en el sistema y la arquitectura preliminar del mismo.

En el sexto y último capítulo definimos el análisis, diseño y desarrollo del sistema. En el análisis describimos los actores, casos de uso y eventos a los que el sistema deberá responder. En el diseño definimos las clases y objetos necesarios para el manejo del sistema, además mostramos el modelo entidad-relación utilizado. En el desarrollo del sistema presentamos las pantallas utilizadas en el mismo y como último punto exponemos las pruebas y resultados del sistema.

Objetivo

Desarrollar un Sistema de Información para Proyectos que Generan Ingresos Extraordinarios, con el fin de registrar proyectos, conocer información relevante, oportuna y confiable, obteniendo un control eficiente de los mismos. La información será consultada vía Internet por medio del sitio Web de la Unidad, con el propósito de ser accesible a todo usuario del sistema.

CAPÍTULO I

**UNIDAD DE ADMINISTRACIÓN DE
PROYECTOS DE INGRESOS
EXTRAORDINARIOS "UAPIEX"**

1.1 Introducción

La Facultad de Ingeniería de la Universidad Nacional Autónoma de México es la institución educativa de nuestro país de mayor prestigio en la formación de ingenieros. Los egresados han destacado en el campo profesional, industrial, de servicios, asimismo en la docencia y en la investigación.

La Facultad tiene vínculos con distintas entidades dentro de la Universidad, tales como, institutos y facultades, y al mismo tiempo con instituciones gubernamentales, empresas privadas y públicas de México; de estos lazos de unión se concretan proyectos en beneficio del país.

Para aquellas empresas que requieren el desarrollo de tecnologías propias, la Facultad es un centro adecuado para ello, ya que cuenta con los recursos humanos idóneos. En tal sentido, los desarrollos e innovaciones tecnológicas redundan en la elevación de la calidad de sus productos y a su vez les permite tener una mayor presencia en los mercados.

Para establecer el vínculo con las empresas, la Facultad de Ingeniería "FI" cuenta con el Centro de Innovación y Servicios de Ingeniería "CISI", y como subdivisión a la Unidad de Administración de Proyectos de Ingresos Extraordinarios "UAPIEX".

El CISI es un órgano coordinador entre el sector productivo y el sector social. Su objetivo es impulsar el desarrollo de innovaciones y ofrecer servicios de ingeniería aplicada, basados en los proyectos y en la investigación, fruto de la capacidad y creatividad de sus académicos y alumnos.

La UAPIEX, por otra parte, es la encargada de gestionar los procesos de captación, ejercicio y control de los proyectos que generan ingresos extraordinarios. Ésta tiene la finalidad de simplificar los procesos administrativos, proporcionando soluciones eficaces y con absoluta transparencia.

Este primer capítulo tiene como objetivo explicar las actividades de la UAPIEX, además de describir el manejo y las tareas requeridas para cumplir con la administración de los proyectos.

1.2 Datos Generales de la Unidad de Administración de Proyectos de Ingresos Extraordinarios "UAPIEX"

La UAPIEX es la encargada de gestionar la información relacionada con los proyectos que generan ingresos extraordinarios. La mejora en la operación de los proyectos es una tarea primordial de la Unidad, evitando la intervención directa de otras áreas operativas.

La entrega de información oportuna y confiable es de vital importancia para la operación de los proyectos. De esta manera, cualquier instancia involucrada se ve beneficiada; además, se evitan deficiencias en el manejo de los proyectos, tales como problemas al registrar los mismos, falta de comunicación entre las instancias proveedoras de información, desconocimiento del avance y beneficios obtenidos de los proyectos, entre otros.

En la Figura 1.1 se muestra el organigrama de la Facultad de Ingeniería de la UNAM, ubicando a la UAPIEX dentro del mismo:



Figura 1.1 Organigrama de la Facultad de Ingeniería de la UNAM

1.2.1 Conceptos generales de Ingresos Extraordinarios

La UAPIEX tiene su base en los conceptos de *Ingresos Extraordinarios* y de *Proyectos que Generan Ingresos Extraordinarios*, los cuales se definen a continuación:

- ✓ **Ingresos Extraordinarios** son los no incluidos en el presupuesto programático anual de la Facultad, aprobado por el Consejo Universitario, y que sean generados por:
 - Prestación de servicios: como los de carácter profesional (médicos, odontólogos, veterinarios, asesorías, consultorías, etc.); los de carácter técnico (mantenimiento y construcción de equipo, uso de laboratorios, etc.); los relacionados con aspectos educativos (cursos, incluyendo los de educación continua, conferencias, seminarios o congresos); y los que se deriven de contratos, convenios o acuerdos. Esto último es de especial importancia, ya que es la base para los proyectos que generan ingresos extraordinarios.
 - Enajenación y arrendamiento: incluye la venta de materiales (publicaciones, libros, apuntes, fotocopias, programas o equipos); y la renta de los bienes.
 - Donativos y aportaciones: como los efectuados por sociedades de alumnos y exalumnos, organizaciones profesionales, empresas, fundaciones, entre otros.
 - Licenciamiento de tecnología.
 - Uso de patentes y cualquier otro diferente a estos: como el uso de recintos culturales, deportivos e instalaciones universitarias, entre otros.
- ✓ **Proyecto que Genera Ingresos Extraordinarios** es una actividad propuesta por el personal académico a la Institución, que a través de convenios, contratos, acuerdos, bases de colaboración y demás instrumentos, suscritos con personas físicas o morales, del sector público o privado, permite la realización de modelos, estudios, trabajos, productos, entre otros.

Además, debe haber una captación de recursos financieros adicional a los incluidos en el presupuesto programático de la Facultad, cuya recepción está sujeta a una aplicación determinada o a la realización concreta de una actividad de asesoría, consultoría, investigación, desarrollo tecnológico u otra similar.

1.2.2 Instancias proveedoras de información de los proyectos

Es claro que para que exista una mejora en la gestión de los proyectos, es necesaria una excelente fluidez de información entre las distintas instancias participantes. A continuación se describen éstas, omitiendo al CISI y a la UAPIEX explicadas con anterioridad.

Dirección de la Facultad de Ingeniería. Instancia que autoriza los proyectos que generan ingresos extraordinarios.

Secretaría Administrativa de la Facultad de Ingeniería. Instancia responsable de proporcionar apoyo en materia administrativa y financiera para la integración de un proyecto, además de dar el visto bueno a los mismos.

Unidad Jurídica de la Facultad de Ingeniería. Instancia responsable de elaborar los instrumentos jurídicos que generen ingresos extraordinarios. La Unidad vigila el cumplimiento, las modificaciones y la conclusión de dichos instrumentos. Éstos pueden ser de tres tipos: convenios generales, de colaboración o específicos.

Responsable de Proyecto. Instancia que capta y promueve proyectos, además de llevar a cabo una supervisión constante de los mismos. Esta instancia es el jefe de la división en la cual está registrado el proyecto.

Líder de Proyecto. Instancia experta en el tema a desarrollar. El líder elabora el programa de trabajo y lo ejecuta. El Líder de Proyecto, a no ser que se asigne a otra persona, es la instancia que está en contacto permanente con la UAPIEX, pues proporciona en todo momento información relacionada con los proyectos que están a su cargo.

Oficina de la Abogada General de la UNAM. Instancia que valida el convenio del proyecto mediante un número de registro y validación que plasma en el instrumento jurídico.

Patrocinador o Entidad Contratante. Instancia responsable de autorizar los proyectos que se llevarán a cabo en su beneficio por la Facultad de Ingeniería de la UNAM. La Comisión Federal de Electricidad (CFE) o Petróleos Mexicanos (PEMEX) son solo algunos de los patrocinadores más importantes para la UNAM, y en particular para la Facultad de Ingeniería.

Unidad Jurídica de la Entidad Contratante. Instancia encargada de gestionar lo relacionado con los instrumentos jurídicos, en común acuerdo con la Unidad Jurídica de la Facultad de Ingeniería.

Responsable de la Entidad Contratante. Instancia responsable de verificar el cumplimiento de las actividades incluidas en el programa de trabajo, el cual debe estar en permanente contacto con el Líder de Proyecto para cumplir dicha tarea.

La Figura 1.2 muestra a las instancias que intervienen en la operación de los proyectos:

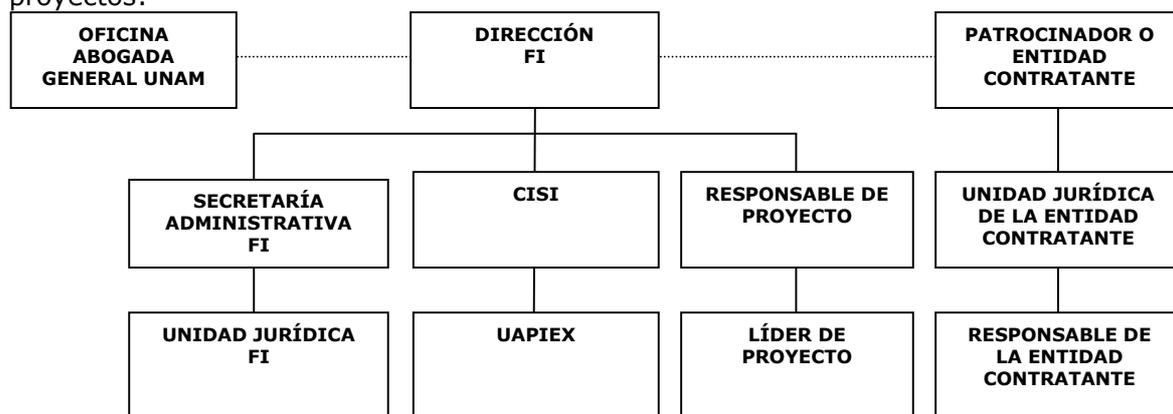


Figura 1.2 Instancias participantes en la operación de los proyectos

1.2.3 Actividades principales de la UAPIEX

Entre sus primordiales tareas se encuentran:

- Identificar, mejorar y dar seguimiento a los diferentes procesos técnico-administrativos que se presentan en la operación de los proyectos.
- Registrar los proyectos en promoción y dar seguimiento hasta la firma del convenio.
- Elaborar y registrar la apertura de proyecto, además de tener un archivo digitalizado del convenio de cada proyecto.
- Colaborar con la Secretaría Administrativa en la gestión de la información financiera, y así, obtener un seguimiento confiable de cada proyecto.
- Verificar los datos de cada proyecto registrado, incluyendo información referente a convenios. Para realizar esta tarea, la UAPIEX se apoya con la Unidad Jurídica de la Facultad de Ingeniería.
- Proporcionar asesoría a Responsables y Líderes de Proyecto en cualquier ámbito referente a los proyectos y, con las mismas instancias mencionadas, recopilar información acerca de nuevos proyectos.
- Proporcionar información preliminar, actual e histórica de los proyectos que generan ingresos extraordinarios a las distintas instancias participantes en los proyectos según lo requieran.
- Administrar el sitio Web de la UAPIEX, que contendrá: Sistema de Información para Proyectos que Generan Ingresos Extraordinarios (SIPIEX), convenios digitalizados de cada proyecto, área de avisos, área de descarga de aplicaciones de la UAPIEX, entre otros.

1.3 Proceso actual del manejo de los proyectos en la UAPIEX

Para que la UAPIEX pueda realizar sus tareas con la mayor efectividad posible, es necesario identificar claramente el proceso para la integración de un proyecto que genera ingresos extraordinarios, además de mejorarlo y darle un seguimiento apropiado.

Al inicio de nuestro análisis, el proceso solo cubría los siguientes puntos:

- El Líder de Proyecto realizaba la apertura y el borrador del convenio, los cuales son documentos que pueden modificarse varias veces, debido al ajuste en las negociaciones con la Entidad Contratante.
- El responsable y líder de proyecto presentaban la apertura de proyecto y quedaba concretado el convenio, que se enviaba a la Unidad Jurídica de la Facultad de Ingeniería y se le informaba a la Secretaría Administrativa.
- Una vez que se firmaba la apertura de proyecto y el convenio, se consideraba al proyecto en el estado de operación.

- Durante el avance del proyecto, la Secretaría Administrativa se encargaba de administrar la información respecto al ingreso y egreso del proyecto.
- Se daba por concluido el proyecto con una carta finiquito expedida por la Unidad Jurídica de la Facultad de Ingeniería.

Inconvenientes:

- La UAPIEX no tenía un papel importante en la gestión de proyectos de ingresos extraordinarios, por lo que no había una instancia que concentrara la información y otorgara los beneficios obtenidos por ello.
- No existía ningún apartado que considerara a los proyectos en promoción, los cuales son de gran importancia para la Facultad de Ingeniería.
- No existía un seguimiento adecuado de las variables cualitativas y presupuestales de cada proyecto.
- No se notifica por ningún medio si un proyecto había sido aceptado, en que estado se encontraba, o si había terminado, pues cada división manejaba sus proyectos e información por separado, únicamente teniendo vínculos con las instancias necesarias en la realización de dicho proyecto.
- El manejo de ingresos y egresos era complicado, pues no se tenía una instancia que proporcionara resúmenes de la información financiera.
- No se guardaban resultados obtenidos de los proyectos, ni ningún tipo de información de relevancia para su futura consulta.

1.3.1 Etapas de los proyectos que generan ingresos extraordinarios

Tomando en cuenta los puntos anteriores, además de sus inconvenientes, detectamos tres etapas en la operación de los proyectos: etapa de promoción, etapa de operación y etapa de memoria tecnológica. Al mismo tiempo, integramos las mejoras identificadas en los procesos, eliminando la participación de instancias innecesarias.

1.3.1.1 Etapa de promoción

La Facultad de Ingeniería ha puesto especial interés en esta etapa, ya que de ésta se pueden obtener datos importantes, tales como: qué nichos de mercado se están tratando, cifras aproximadas de cuantos proyectos están por realizarse en la Facultad y monto aproximado del ingreso a futuro. Esto último trasciende en una mejor planeación financiera para la Facultad, así como poder conocer con mayor exactitud los beneficios que se obtendrán, ya sea a corto o largo plazo.

Esta etapa contempla *información preliminar* de los proyectos, esto es, nombre del proyecto, responsable, líder, división o campo disciplinario al que pertenece, patrocinador o entidad contratante, importe probable, avance, estatus del convenio y objetivo del mismo. Los datos mencionados son recopilados por medio de un formato llamado "Alta Preliminar" (ver Anexo A).

Existen algunos puntos a considerar para que el proyecto se registre en esta etapa, por ejemplo: pláticas formales entre el Líder de Proyecto y la Entidad Contratante, la realización de un borrador de convenio, entre otros. No obstante, la información proporcionada por el Líder de Proyecto a la UAPIEX es la más importante, ya que de él depende que un proyecto paulatinamente se concrete y sea tomado en cuenta como proyecto en promoción. Los diagramas de flujo los podemos apreciar en las Figuras 1.3, 1.4 y 1.5.

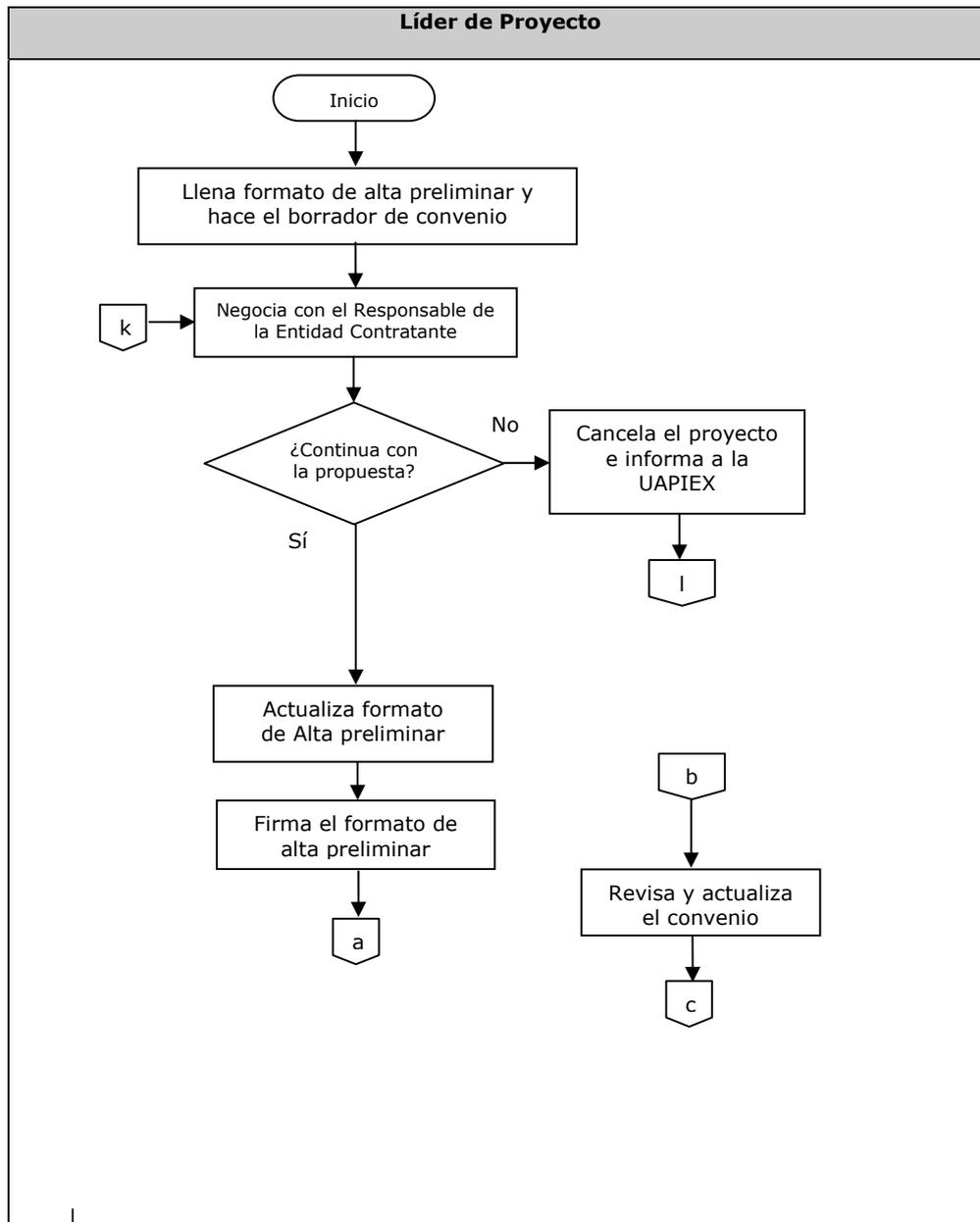


Figura 1.3 Diagrama de Flujo en la Etapa de Promoción

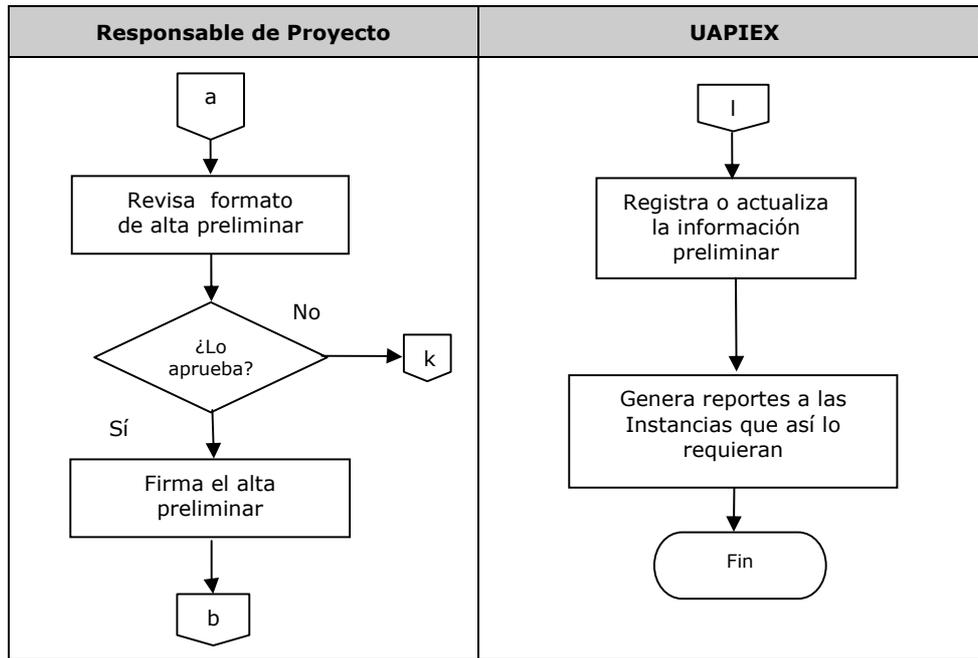


Figura 1.4 Diagrama de Flujo de Alta preliminar en la Etapa de Promoción

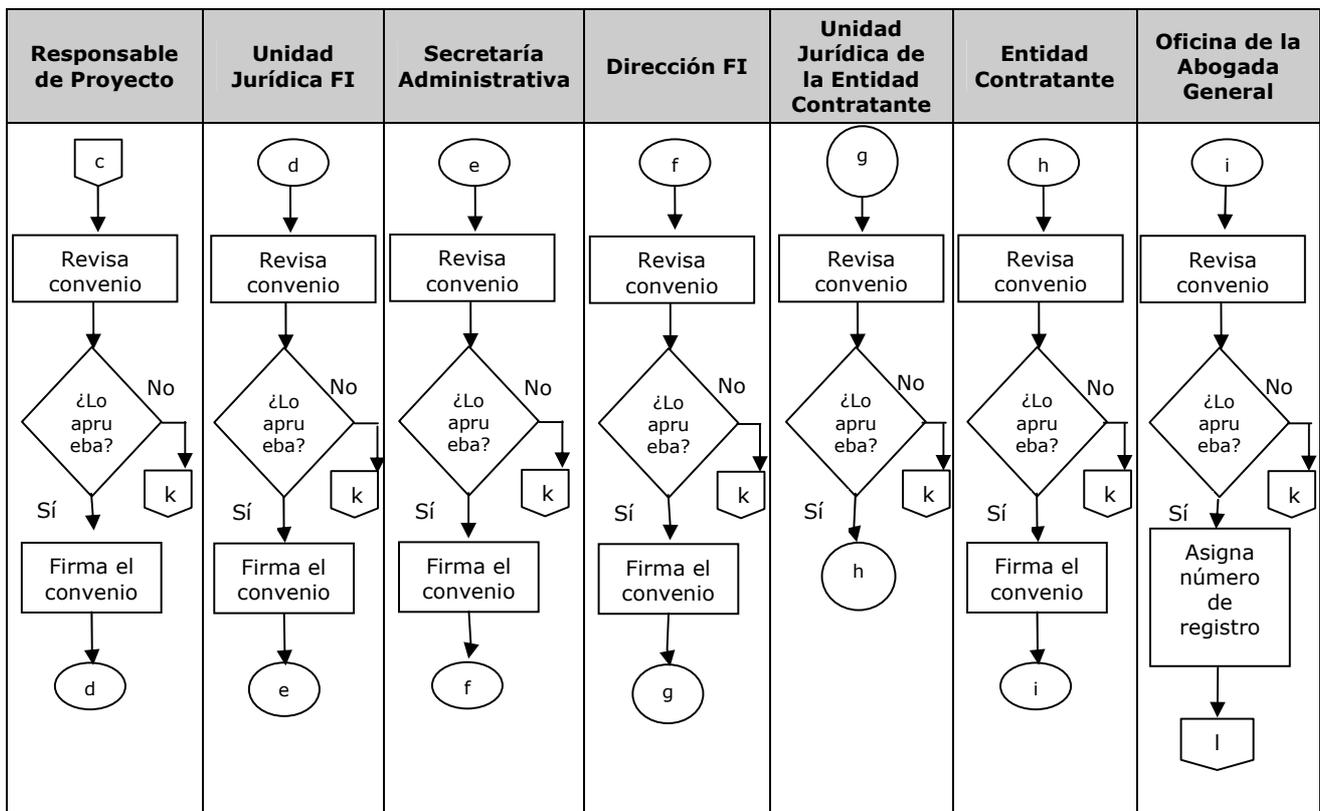


Figura 1.5 Diagrama de Flujo del Convenio en la Etapa de Promoción

1.3.1.2 Etapa de operación

Una vez que se firmó el convenio, se coloca al proyecto en la etapa de operación. Por otro lado la Apertura de Proyecto es un formato oficial manejado en la Facultad de Ingeniería con el objetivo de registrar al proyecto. Este formato es generado por el Líder de Proyecto (ver Anexo A).

Una vez generada la Apertura de Proyecto es necesario que las instancias involucradas en esta etapa proporcionen información a la UAPIEX. Se identificaron tres escenarios, los cuales describiremos a continuación:

- **Escenario Técnico:** La información proporcionada en este punto incluye distintas variables, ya registradas pero deben seguirse actualizando, tales como: nombre, importe, objetivo, entre otras. La importancia de este tipo de información radica en su futura consulta, reincidiendo en la última etapa. El Líder de Proyecto es quien proporciona esta información a la UAPIEX.
- **Escenario Financiero:** En esta sección se provee información respecto al ingreso y egreso del proyecto, haciendo énfasis por mostrar solo resúmenes, pues el objetivo no es llevar la contabilidad de los proyectos, sino solo tener un estatus financiero. La Secretaría Administrativa y el Líder de Proyecto son los responsables de proporcionar esta información a la UAPIEX.
- **Escenario Jurídico:** La UAPIEX tiene en su poder una copia del convenio digitalizado de cada proyecto, el cual se podrá consultar con el fin de verificar que todo marche en orden con las condiciones mencionadas en el mismo, y asimismo tener un mayor control de cada proyecto. Además, la UAPIEX se apoya de la Unidad Jurídica de la FI para resolver cualquier inconveniente generado en este rubro. Se da por concluido el proyecto con una carta finiquito expedida por la Unidad Jurídica de la Facultad de Ingeniería.

Los diagramas de flujo de esta etapa los observamos en las Figuras 1.6, 1.7 y 1.8.

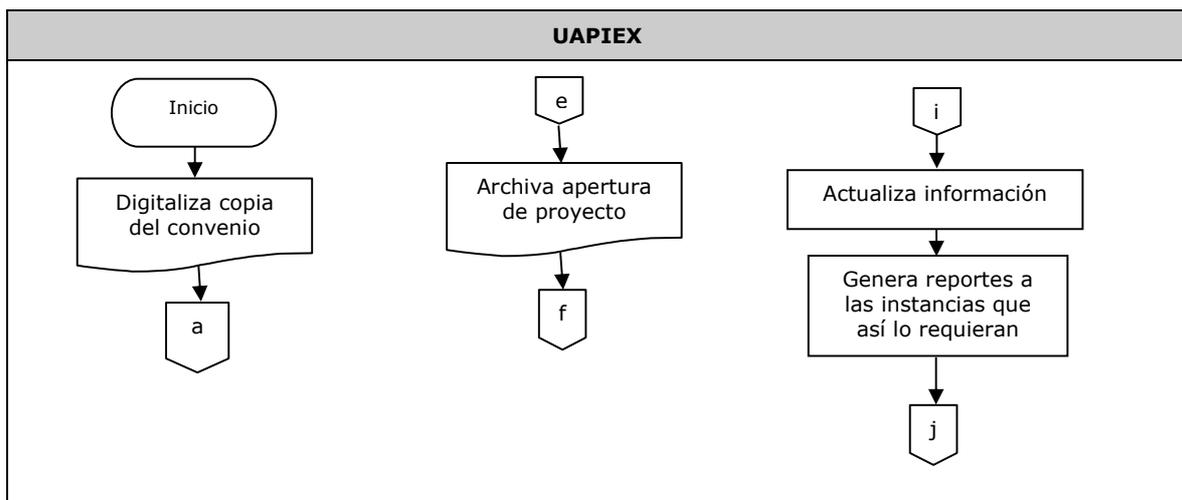


Figura 1.6 Flujo de Información en la Etapa de Operación

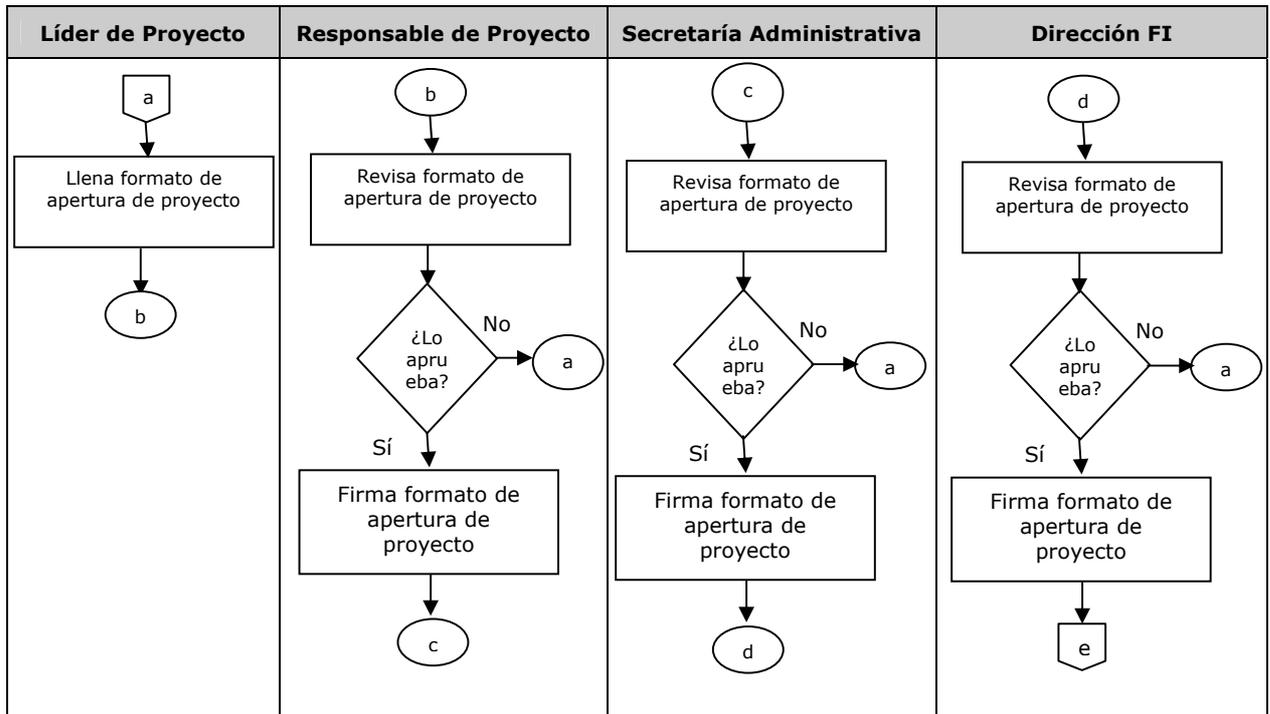


Figura 1.7 Flujo de Información de la Apertura de Proyecto en la Etapa de Operación

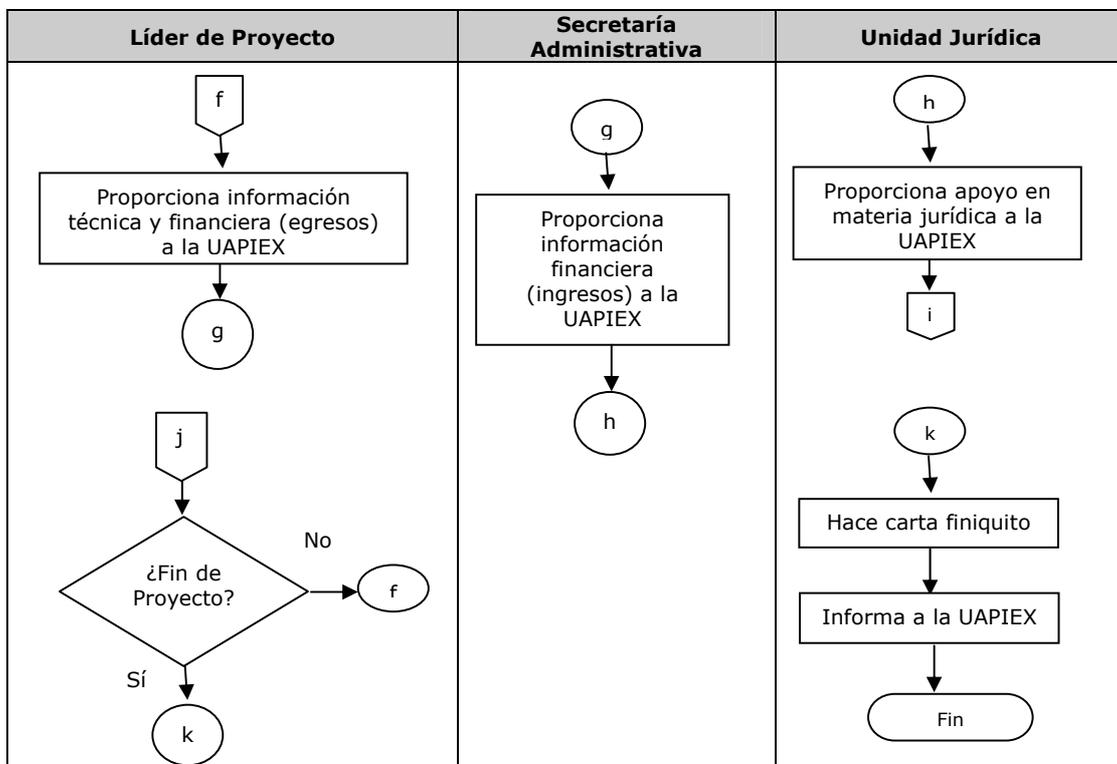


Figura 1.8 Flujo de Información de los Escenarios Técnico, Financiero y Jurídico en la Etapa de Operación

1.3.1.3 Etapa de memoria tecnológica

Esta última etapa integra información técnica de los proyectos en cuanto a su metodología, objetivos, personas que participaron en su realización y resultados del mismo, con el propósito de conformar una "memoria tecnológica" o información histórica, la cual podrá ser consultada por investigadores, profesores, alumnos, entidades contratantes, directivos, coordinadores internos de la FI y público en general interesado.

Además la memoria tecnológica puede contener referencias a publicaciones que son hechas por el líder y el grupo de trabajo del proyecto, y en este sentido, las personas que así lo requieran, pueden profundizar en el tema.

La mayoría de la información contenida en la memoria tecnológica es tomada de la etapa anterior, aunque puede existir información adicional que el Líder de Proyecto debe proporcionar, en especial, los resultados y publicaciones del proyecto.

Un beneficio inmediato de la memoria tecnológica es la promoción de proyectos exitosos y de esta manera se cierra el ciclo de vida del mismo, es decir, un proyecto se promueve con la entidad contratante, se desarrolla, se culmina exitosamente, y este mismo sirve para fortalecer y crear nuevos lazos de unión entre la UNAM y las empresas.

El diagrama de flujo lo observamos en la Figura 1.9.

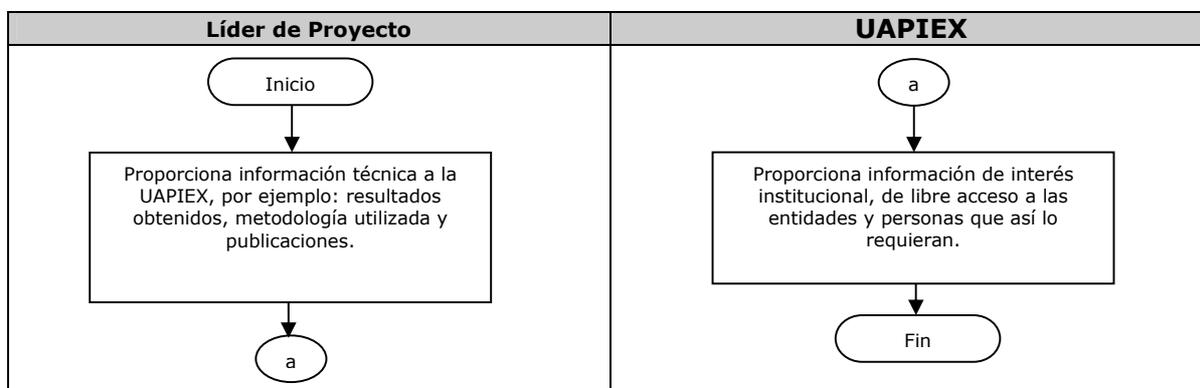


Figura 1.9 Flujo de Información en la Etapa de Memoria Tecnológica

Con lo anterior, se ha expuesto el proceso en la gestión de los proyectos que generan ingresos extraordinarios.

Con ayuda del sistema de información, la UAPIEX podrá proporcionar información en tiempo y forma a cada una de las instancias involucradas en los proyectos que generan ingresos extraordinarios en la FI de la UNAM.

CAPÍTULO II

METODOLOGÍAS ORIENTADAS A OBJETOS

2.1 Introducción

Las tecnologías orientadas a objetos se han convertido en uno de los motores clave de la industria del software y a la par se han hecho innovaciones en las metodologías utilizadas para el desarrollo de software. Esta evolución de metodologías tiene como propósito la creación de sistemas cada vez más complejos, acordes a las necesidades presentes.

Las tecnologías más comunes para el modelado de software son la perspectiva del análisis estructurado y la perspectiva orientada a objetos. En el análisis estructurado, la construcción principal del sistema es la función y el proceso, y el modelado se realiza en descomposición de funciones. En la perspectiva de modelado orientada a objetos el bloque principal de todo el sistema es la clase y el objeto, y el problema se analiza mediante el estudio de los objetos dentro del ámbito en que se presentan.

El modelado contemporáneo se ha inclinado por la metodología orientada a objetos, pues ha logrado derribar problemas que se resisten al análisis estructurado. Si bien las metodologías estructuradas marcaron la pauta para la construcción de software, la metodología orientada a objetos es un paradigma que ha demostrado ser de gran utilidad para la solución de problemas clásicos que se presentan al elaborar software, brindando, entre otros beneficios: software reutilizable, costos y tiempo de desarrollo menores, sistemas de alta calidad, fáciles de modificar y de mejorar.

Una metodología orientada a objetos abarca métodos de análisis, diseño e implantación, permitiendo especificar la conducta de la estructura del sistema, documentar el problema desde su construcción, además de proporcionar un anteproyecto del sistema final.

El auge de las tecnologías orientadas a objetos se debe fundamentalmente a la existencia y popularidad de varios lenguajes de programación orientados a objetos, tales como C++ o Java, por lo que es de vital importancia tener un buen entendimiento del tema.

Este capítulo es una introducción a las metodologías orientadas a objetos, en el cual analizaremos los conceptos básicos y la terminología sobre las metodologías más utilizadas, esto es, Booch, OMT y UML, siendo esta última la estandarización de las anteriores.

Debido a las ventajas que UML nos brinda al momento de especificar, visualizar, construir y documentar un sistema de software, es la metodología utilizada en el presente trabajo y, por consiguiente, será especialmente tratada.

2.2 Conceptos Fundamentales de Metodologías Orientadas a Objetos

Para comenzar con este capítulo, consideramos primordial mencionar los conceptos relacionados con las tecnologías orientadas a objetos, para un mejor entendimiento de éstas. *La orientación a objetos es un conjunto de disciplinas (ingeniería) que*

desarrollan y modelan software que facilitan la construcción de sistemas complejos a partir de componentes¹.

El atractivo de la orientación a objetos es que proporciona conceptos y herramientas con las cuales se modela y representa el mundo real tan fielmente como sea posible. La orientación a objetos trata de cubrir las necesidades de los usuarios finales, así como las propias de los desarrolladores de productos de software.

A continuación mencionaremos las propiedades más importantes de la orientación a objetos.

2.2.1 Objetos

Definimos un objeto como un concepto, una abstracción o cosa con límites definidos y significativos para el problema en cuestión. Los objetos sirven para dos propósitos: promueven el entendimiento del mundo real y proporcionan una base práctica para la implantación computacional. La descomposición de un problema en objetos depende de la naturaleza del problema.

Un objeto es una instancia de una clase. Una idea fundamental en la orientación a objetos es la comunicación de los objetos a través del paso de mensajes.

2.2.2 Clases

Una clase es un concepto discreto dentro de la aplicación que se está modelando: una cosa física, de negocios, lógica o de una aplicación. Las clases son los focos alrededor de los cuales se organizan los sistemas orientados a objetos.

Una clase se puede definir como una descripción abstracta de un grupo de objetos con las mismas características, esto es, siguen una misma estructura, comportamiento y relaciones similares, cada una de las cuales se diferencia por su estado específico (atributos) y por la posibilidad de realizar una serie de operaciones (métodos).

Un atributo es una característica o propiedades de una clase y aplica a todos los objetos de dicha clase. Un método es la realización de una función, servicio o acción que se puede aplicar a todos los objetos de una clase para comportarse de cierta manera. El nombre de un método es generalmente un verbo que representa claramente el comportamiento de la clase.

2.2.3 Comunicación entre objetos

Un sistema orientado a objetos consiste en un conjunto de objetos que se comunican unos con otros llamando a sus métodos. Estos últimos residen en el objeto y determinan cómo actúan en el mismo cuando reciben un mensaje.

Un mensaje es la acción que hace un objeto. Un método es el procedimiento que se invoca para actuar sobre un objeto, esto es, especifica cómo se ejecuta un mensaje. El conjunto de mensajes a los cuales puede responder un objeto se denomina protocolo del objeto. Estos mensajes son los únicos conductos que conectan al objeto con el mundo externo.

¹ Joyanes Aguilar Luis, Programación Orientada a Objetos, 2ª. Ed., Osborne McGraw-Hill, 1998.

2.2.4 Herencia

La herencia es una propiedad que permite a los objetos ser construidos a partir de otros objetos. El objetivo final es la reutilización de código, es decir, utilizar las clases anteriormente desarrolladas.

La herencia supone una *clase base* y una *jerarquía de clases* que contiene las clases derivadas de la clase base. Las clases derivadas pueden heredar características de su clase base, pero añaden otras nuevas características propias. Las clases que heredan propiedades de una clase base pueden a su vez servir como definiciones base de otras clases.

Existen dos mecanismos utilizados normalmente: herencia simple y herencia múltiple. En la herencia simple, una clase puede tener solo un ascendente, o dicho de otro modo, una subclase puede heredar características de una única clase. La herencia múltiple es la propiedad de una clase de poder heredar características de más de una clase. Básicamente, todo lo que se puede hacer con herencia múltiple se puede hacer con herencia simple, aunque a veces resulta más difícil utilizar herencia múltiple debido a que se llegan a combinar diferentes tipos de objetos, cada uno de los cuales define métodos o atributos iguales y con esto pueden surgir problemas de ambigüedad.

2.2.5 Polimorfismo

Esta propiedad supone que un mismo mensaje puede producir acciones (resultados) totalmente diferentes cuando se reciben por objetos diferentes.

Con esta propiedad un usuario puede enviar un mensaje genérico y dejar los detalles de la implantación exacta para el objeto que recibe el mensaje (ya sea un objeto padre o hijo), por esto último, podemos deducir que el polimorfismo se fortalece con el mecanismo de herencia.

2.2.6 Sobrecarga

La sobrecarga es una clase de polimorfismo. La sobrecarga es una propiedad en la cual se utiliza el mismo nombre de una operación para representar operaciones similares que se comportan de modo diferente cuando se aplican a clases diferentes o de la misma. Por consiguiente, los nombres de las operaciones se pueden sobrecargar, esto es, las operaciones se definen en clases diferentes o en la misma, pudiendo tener nombre idénticos, aunque su código programado puede diferir.

Si el nombre de una operación se utiliza para nuevas definiciones en clases de una jerarquía, la operación del nivel inferior anula la operación del nivel más alto.

La sobrecarga elimina la necesidad de definir métodos diferentes que en esencia hacen lo mismo, también hace posible que un método se comporte de una u otra forma según el número de argumentos con el que sea invocado.

2.2.7 Abstracción

La abstracción es la propiedad que permite representar las características esenciales de un objeto, sin preocuparse de las características restantes (no esenciales).

Una abstracción se centra en la vista externa de un objeto, de modo que sirva para separar su comportamiento esencial de la implantación. Definir una abstracción significa describir una entidad del mundo real, no importa lo compleja que pueda ser.

Es importante aclarar que una clase abstracta sirve como base común para otras clases, pero no tendrá instancias.

2.2.8 Encapsulamiento

El encapsulamiento o encapsulación es la propiedad que permite asegurar que el contenido de la información de un objeto está oculto al mundo exterior. La encapsulación es ver al objeto como una caja negra, la cual, sólo sabemos utilizarla pero no conocemos cómo fue hecha ni su contenido.

La encapsulación permite la división de un sistema en módulos. Éstos se implantan mediante clases, de forma que una clase representa la encapsulación de la abstracción. Cada clase debe constar de dos partes: una interfaz y una implantación. La interfaz de una clase captura sólo su vista externa y la implantación contiene la representación de la abstracción, así como los mecanismos que realizan el comportamiento deseado.

2.2.9 Modularidad

La modularidad es la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), en las que cada una debe ser tan independiente como sea posible de la aplicación en sí y de las partes restantes.

2.2.10 Jerarquía

La jerarquía es una propiedad que permite una ordenación de las abstracciones. Las dos jerarquías más importantes de un sistema complejo son: estructura de clases (generalización/especialización) y estructura de objetos (agregación), las cuales serán detalladas más adelante.

2.3 Metodología orientada a objetos "Booch"

Este método marcó la conducta en el análisis del diseño orientado a objetos. Booch lo modela desde un punto de vista lógico para representar gráficamente clases, objetos y las relaciones entre ellos.

La orientación a objetos que Booch propone es iterativa, creciente y evolutiva, utilizando la combinación de un concepto llamado *microproceso* y *macroproceso* de desarrollo.

2.3.1 Macroproceso

El *Macroproceso* se inclina a observar el problema desde el punto de vista global, esto es, la visión que podría tener un administrador o el líder de proyecto. Sirve como marco de referencia para controlar al microproceso y dicta una serie de procesos y actividades, que permiten al equipo de desarrollo evaluar el riesgo de forma que se centren mejor las actividades de análisis y diseño de equipo.

Las fases que constituyen al macroproceso son:

- La conceptualización. Establece los requerimientos básicos.
- Análisis. Desarrollo de un modelo de acuerdo al comportamiento del problema.
- Diseño. Creación de la arquitectura a detalle.
- Evolución o Puesta en Marcha. Se realiza la prueba, corrección y entrega, tantas veces como sea necesario y evoluciona hacia la versión para el usuario final.
- Mantenimiento. Se realiza después de la entrega. Cabe mencionar que si esta etapa provoca nuevas ampliaciones del sistema que en la primera fase no fueron previstas, entonces se comenzará nuevamente el macroproceso.

2.3.2 Microproceso.

El *Microproceso* está orientado a una enfoque más estrecho; es la visión que un programador podría tener del sistema. El *microproceso* consta de 4 etapas:

- Identificación de las clases y objetos a un nivel de abstracción.
- Identificación de semánticas de estas clases y objetos.
- Identificación de las relaciones entre esas clases y objetos.
- Implantación de las clases y objetos.

Además, ofrece cuatro técnicas para la documentación de:

- *Diagrama de clases*, los cuales indican la existencia de clases y sus relaciones.
- *Diagrama de objetos*, los cuales son usados para mostrar la existencia de objetos y su comportamiento.
- *Diagrama de estado-transición*, los cuales muestran los estado posibles de cada objeto.
- *Diagramas de tiempo*, los cuales indican la secuencia de operaciones de los objetos.

2.3.3 Notación Booch

Booch propone su notación pre-UML para representar gráficamente clases, objetos y relaciones entre ellos, siendo sus nubes lo más sobresaliente. Con ellas denota clases (nubes con líneas punteadas) y objetos (nubes con líneas continuas) ambas conteniendo su nombre, atributos y métodos. También proporcionó la notación para las diferentes relaciones entre clases (de asociación, agregación y herencia).

2.3.3.1 Objetos

Los objetos se dibujan usando una nube con el borde continuo (Figura 2.1). Cada objeto contiene su nombre, atributos y operaciones.



Figura 2.1 Notación de un Objeto en Booch

2.3.3.2 Clases

Una clase está representada por una nube punteada (Figura 2.2). Dentro de ella contiene su nombre, sus atributos y operaciones. Los atributos son las características que representan a la clase y las operaciones son las funciones o acciones que se pueden aplicar a los objetos de una clase.

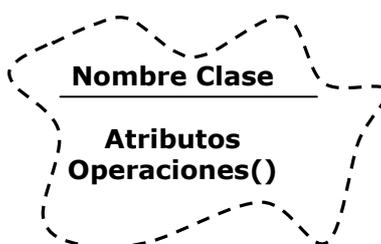


Figura 2.2 Notación de una Clase en Booch

Para proporcionar información adicional de una clase, esta notación permite utilizar símbolos que a continuación se describen:

A – Abstracta. Es una clase que puede usarse solamente como base de alguna otra clase. Es usada para definir el comportamiento común de otras clases llamadas comúnmente subclasses. Además, no puede ser instanciada porque representa una gran variedad de objetos. Su representación gráfica se muestra en la Figura 2.3.

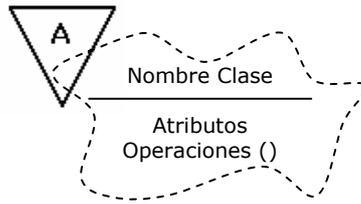


Figura 2.3 Notación de una Clase Abstracta en Booch

F – Friend (Amiga). Las clases de este tipo admiten funciones no públicas y otras clases (Figura 2.4).

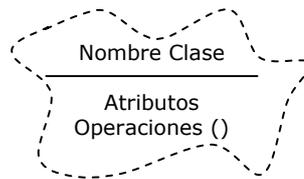


Figura 2.4 Notación de una Clase Amiga en Booch

S – Static. Una clase estática (static) provee datos Su representación se muestra en la Figura 2.5.

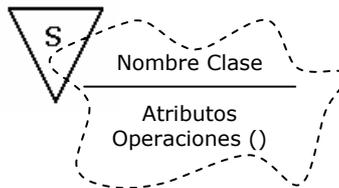


Figura 2.5 Notación de una Clase Estática en Booch

V – Virtual. Una clase Virtual comparte una clase base, es la clase más general en el sistema (Figura 2.6).

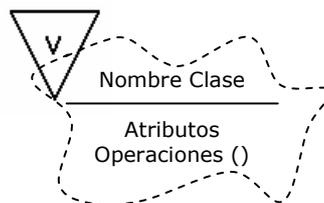


Figura 2.6 Notación de una Clase Virtual en Booch

Esta metodología también incorporó un símbolo para englobar clases, que representan un cluster de clases similares. Su notación es un rectángulo con compartimentos (Figura 2.7).

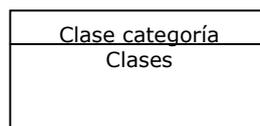


Figura 2.7 Notación de un Cluster en Booch

2.3.3.3 Relaciones

Para denotar relaciones entre objetos, Booch utilizó líneas y flechas (Figura 2.8).



Figura 2.8 Notación de Relaciones en Booch

La metodología Booch, sin embargo, no alcanzó a cubrir totalmente la necesidad de crear sistemas cada vez más novedosos y complejos, por lo que surgieron otras metodologías.

2.4 Metodología orientada a objetos "OMT"

Aunque en la evolución de las diferentes metodologías se han utilizado modelos orientados a objetos, muchas de éstas no han tenido soportes suficientes para desarrollar aplicaciones de bases de datos o no ponen énfasis en la importancia de la estructura de datos.

OMT (Object Modeling Technique), sin embargo, pone atención en la importancia del modelo y uso del mismo para lograr una abstracción, en la cual el análisis está enfocado al mundo real para un nivel de diseño, además, agrega especial interés para el modelado de los recursos computacionales.

El objetivo principal de OMT es la conceptualización de una entidad que permita manejar atributos, asociaciones y sus entidades, así como obtener una mejor interacción con la base de datos. Por todo ello, OMT llegó a ser líder en el avance de las metodologías orientadas a objetos.

La metodología de modelado de objetos OMT descrito por Rumbaugh, tiene como objetivo desarrollar un modelo de sistemas, mediante objetos y las relaciones entre ellos.

2.4.1 Fases de la metodología OMT

Las fases utilizadas en la metodología OMT son: análisis del sistema, diseño del sistema y diseño de objetos.

2.4.1.1 Análisis del sistema

Su objetivo es desarrollar un modelo del sistema. El modelo se expresa en función de objetos, relaciones entre ellos, flujo dinámico de control y las transformaciones funcionales. Los pasos a seguir en el análisis son:

- Se obtiene una descripción inicial del problema.
- Se construye el *modelo de objetos* y sus relaciones. Su objetivo es describir la estructura estática del software, esto es: se abstraen los conceptos de los datos que son más importantes para la aplicación y se describen gráficamente por los diagramas de objetos que definen las clases y sus relaciones.

- Se desarrolla un *modelo dinámico*. Su objetivo es describir los aspectos del sistema que cambian conforme pasa el tiempo.
- Se construye un *modelo funcional*. Su objetivo es describir las transformaciones de los datos del sistema.
- Se verifican, iteran y depuran los tres modelos.

2.4.1.2 Diseño del sistema

En esta etapa se define la arquitectura del sistema. Los pasos para el diseño del sistema son:

- Organizar el sistema en subsistemas y ordenarlos en capas y divisiones.
- Identificar la concurrencia inherente en el problema.
- Asignar subsistemas a procesos.
- Definir la estrategia de implantación del administrador de datos.
- Identificar las fuentes globales y definir el mecanismo para controlar el acceso a ellos.
- Elegir un enfoque para la implantación de control de software.
- Considerar las condiciones de los límites.
- Establecer cambios fuera de las prioridades.

2.4.1.3 Diseño de objetos

Su objetivo es depurar el modelo del análisis y proporcionar una base detallada para la implantación, tomando en cuenta el ambiente en que se construye. Los pasos que se realizan en el diseño de objetos son los siguientes:

- Se depuran las operaciones para el modelado de objetos.
- Se diseñan algoritmos para las operaciones y estructuras de datos.
- Se optimizan las vías de acceso a los datos.
- Se construye un sistema controlado por procedimientos.
- Se empaquetan las clases y las asociaciones en módulos.

2.4.2 Notación OMT

La metodología OMT, igual que las metodologías predecesoras, representa gráficamente conceptos claves en la estructura del sistema tales como: clases, objetos y asociaciones. Esta notación si bien, no tiene la gama de opciones que las metodologías actuales manejan, sí mantiene características afines con las actuales.

2.4.2.1 Objetos

Los objetos en OMT se representan con un rectángulo redondeado con su respectivo nombre, atributos y valores (Figura 2.9).

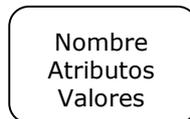


Figura 2.9 Notación de un Objeto en OMT

2.4.2.2 Clases

Las clases se representan por rectángulos con tres compartimentos. En el primero se pone el nombre de la clase, en el segundo los atributos y en el tercero las operaciones o métodos (Figura 2.10).

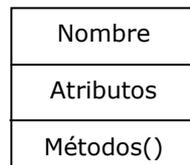


Figura 2.10 Notación de una Clase en OMT

2.4.2.3 Asociaciones

La asociación es una dependencia entre dos o más clases. Las asociaciones se representan por líneas que unen a las clases (Figura 2.11).

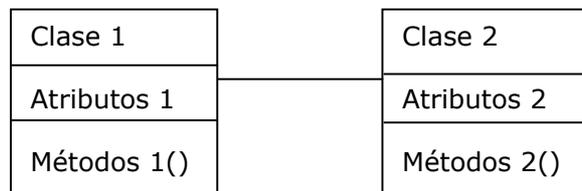


Figura 2.11 Notación de una Asociación en OMT

La multiplicidad de asociaciones especifica cuantas instancias de una clase pueden relacionarse a cada instancia de otra clase. En el caso más general, la multiplicidad puede ser especificada con un número o un conjunto de intervalos, tales como "1" (exactamente uno), "1+" (uno o más), "3-5" (tres a cinco) y "2,4,18" (dos, cuatro o dieciocho).

Existen terminadores de línea especiales para indicar ciertos valores de multiplicidad comunes. Un círculo relleno es un símbolo de OMT para "muchos" y significa cero o más. Un círculo vacío indica "opcional", indica cero o uno. Las líneas sin símbolos de multiplicidad indican una asociación uno a uno. En el caso general, el grado de multiplicidad se escribe a un lado del extremo de la línea, por ejemplo, "1+" para indicar uno o más (Figura 2.12).

Esta metodología, a pesar de que cubrió puntos que Booch u otras metodologías no habían satisfecho, quedó muy lejos de ser el estándar para las nuevas tecnologías en el desarrollo de sistemas.

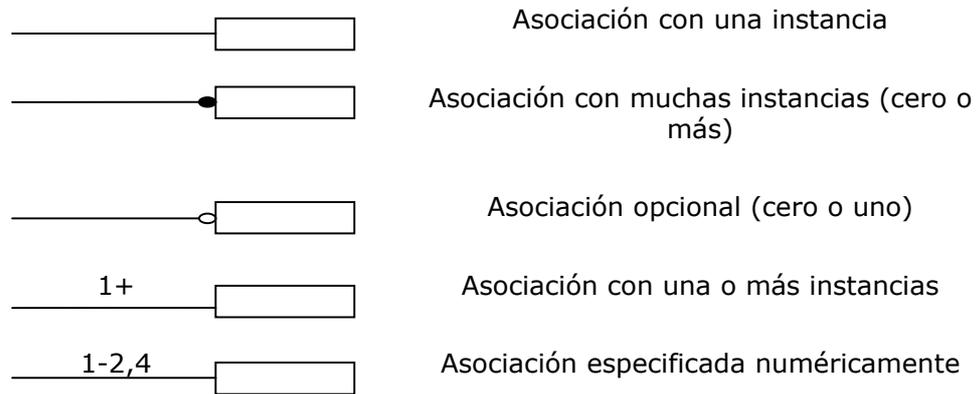


Figura 2.12 Notación de Multiplicidad en OMT

2.5 Metodología orientada a objetos "UML"

UML (Unified Modeling Language), es probablemente, una de las mayores innovaciones conceptuales en el mundo tecnológico del desarrollo del software. Es actualmente el estándar de la industria que requiera la construcción de modelos como condición previa al diseño y después a la construcción de prototipos.

UML se diseñó combinando una gran cantidad de estándares, si embargo se rige a través de tres metodologías precedentes (Booch, OMT y Objectory) y de la colaboración de J. Rumbaugh, G. Booch e I. Jacobson, además del análisis y estudio de alrededor de 20 métodos estándares.

Este lenguaje de modelado es de propósito general, el cual puede ser usado por todos los modeladores, no tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática. Está pensado en reemplazar al menos los modelos de Booch, OMT, entre otros, y es utilizado para entender, diseñar, configurar, mantener y controlar la información de sistemas computacionales.

2.5.1 Notación UML

La notación que esta metodología maneja comprende el comportamiento de todo el sistema, permitiendo la realización de diagramas que lo describen. Las metas claves de UML se encuentran en integrar la terminología más aceptada comúnmente y, con ello, permitir contar con software avanzado que proporcione diagramas mejor definidos para el desarrollo del sistema.

2.5.1.1 Objetos

La notación del objeto es un rectángulo con dos compartimentos. El compartimiento superior contiene el nombre del objeto y el nombre de la clase a la que pertenece, ambos subrayados para distinguirlo como individuo. El compartimiento inferior

contiene la lista de nombres y valores de atributos. En ellos no existe necesidad de mostrar sus operaciones porque son las mismas para todos los objetos de la clase (Figura 2.13).



Figura 2.13 Notación de un Objeto en UML

2.5.1.2 Clases

Las clases en UML se dibujan como rectángulos. Las listas de atributos y operaciones se dibujan en compartimentos separados (Figura 2.14). Los compartimentos pueden ser suprimidos cuando no es necesario el detalle completo. Además una clase puede aparecer en varios diagramas.

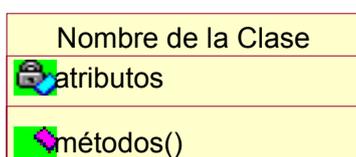


Figura 2. 14 Notación de una Clase en UML

2.5.1.3 Relaciones

Las relaciones entre clases son asociaciones, generalizaciones, agregaciones y varias clases de dependencia, que incluyen la de realización y de uso.

Las relaciones entre clases se dibujan como líneas que conectan rectángulos de clases. Los distintos tipos de relaciones se diferencian por la textura de la línea y por los adornos de las mismas en sus extremos.

2.5.1.3.1 Relación de asociación

La relación de asociación proporciona las conexiones con las cuales los objetos de diversas clases pueden interactuar. Una asociación describe conexiones discretas entre objetos u otras instancias de un sistema.

Las asociaciones llevan la información sobre relaciones entre objetos en un sistema. Éstas mantienen unido un sistema. Sin ellas no hay más que clases aisladas que no trabajan juntas.

Cada conexión de una asociación a una clase se llama extremo. Los extremos de una asociación pueden tener nombres llamados *roles* y éstos son nombrados de acuerdo al tipo de asociación que exista entre las clases. La propiedad más importante que tienen las asociaciones es la multiplicidad, ésta indica cuántas instancias de una clase se pueden relacionar con una instancia de otra clase.

La direccionalidad es otra característica importante de las asociaciones y se determina mediante una triángulo indicando la dirección de la asociación.

La notación UML para una asociación es una línea continua que conecta a dos clases. El nombre de la asociación se pone a lo largo de la línea con el nombre del rol y la multiplicidad en cada extremo. Las multiplicidades más comúnmente utilizadas en UML son:

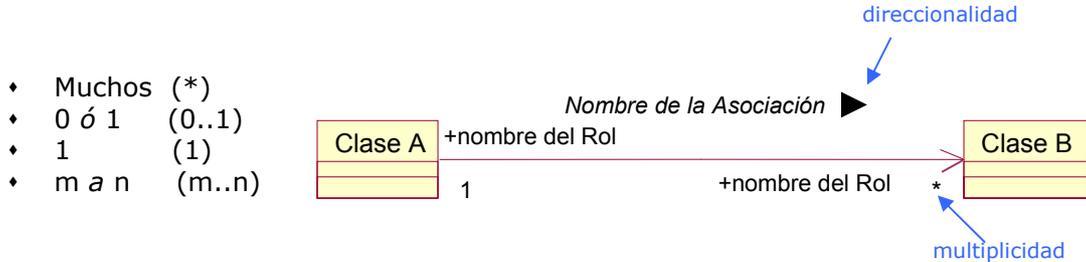


Figura 2.15 Notación de Asociación en UML

La Figura 2.15 muestra una multiplicidad de 1 a muchos (*), esto es, que una instancia de la *Clase A* puede relacionarse con muchas instancias de la *Clase B*. De la misma manera podemos hacer uso de las multiplicidades anteriormente mencionadas.

Una asociación puede también tener atributos por sí misma, en cuyo caso es una asociación y una clase, conocida como una clase asociación (Figura 2.16).

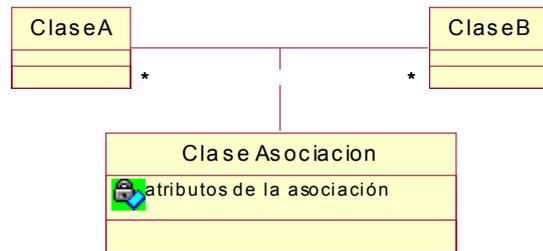


Figura 2.16 Notación de Clase Asociación en UML

Si un atributo de la asociación es único dentro de un conjunto de objetos relacionados, entonces es un calificador. El calificador es un valor que selecciona un objeto único del conjunto de objetos relacionados a través de la asociación (Figura 2.17).



Figura 2.17 Notación de Asociación con calificador en UML

2.5.1.3.2 Relación de generalización

La relación de generalización relaciona descripciones generales de la clase padre (superclase) con clases hijos (subclases). La generalización permite compartir atributos, operaciones y relaciones en común, de diferentes clases sin tener que repetirlos.

En la generalización la clase padre define los métodos y atributos de manera más general, mientras que la clase hija se extiende de la clase padre definiendo más específicamente los métodos de la superclase.

Una generalización se dibuja como una flecha desde el hijo al padre, con un triángulo hueco en el extremo conectado con el padre (Figura 2.18).

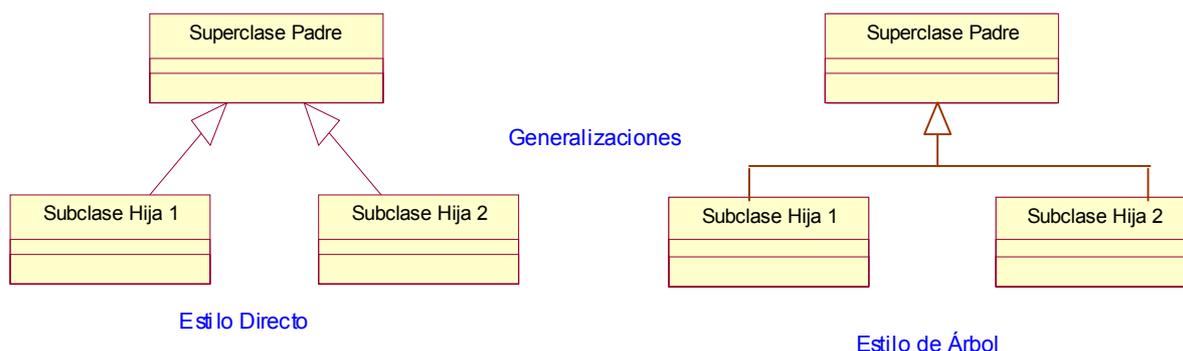


Figura 2.18 Notación de Generalización en UML

La generalización permite operaciones polimórficas, es decir, operaciones cuya implantación (método) es determinada por la clase objeto a la que se aplican, en vez de ser indicada explícitamente. Esto funciona porque una clase padre puede tener muchos hijos posibles, cada uno de los cuales implanta su propia variación de una operación que se define a través de todo el conjunto de clases.

El propósito de la generalización es permitir la descripción incremental de un elemento que comparte las descripciones de sus antecesores (herencia). Al compartir, se reduce el tamaño del modelo y más importante aún, se reduce el número de los cambios y la posibilidad de inconsistencia accidental al momento de realizar una actualización al modelo.

2.5.1.3.3 Relación de agregación y composición

Estas relaciones representan una relación todo – partes, donde *todo* se refiere a la clase principal y las *partes* son piezas de la clase principal.

La agregación es aquella en que las partes pueden ser partes en cualquier todo. Además la multiplicidad, en el *lado todo*, debe ser distinta de uno.

En la composición existe un alto grado de pertenencia, esto es, las partes existen dentro del todo y se destruyen junto con él. Además, la clase *todo* es la responsable de gestionar sus partes.

La notación de agregación es un diamante hueco en el extremo de la trayectoria unida a la clase agregada (Figura 2.19 (a)). Por otra parte, en la composición la *clase todo* se muestra con un diamante relleno (Figura 2.19 (b)).



Figura 2.19(a) Notación para la agregación en UML

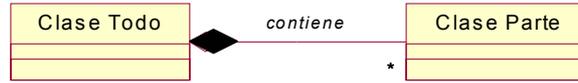


Figura 2.19(b)Notación para composición en UML

2.5.1.3.4 Relación de realización

La relación de realización relaciona una especificación con una implantación. Una interfaz es una especificación del comportamiento sin la implantación; una clase incluye la estructura de implantación. Una o más clases pueden realizar una interfaz, y cada clase implementa las operaciones de la interfaz.

La realización se indica con una flecha de línea discontinua con una punta de flecha hueca cerrada (Figura 2.20).

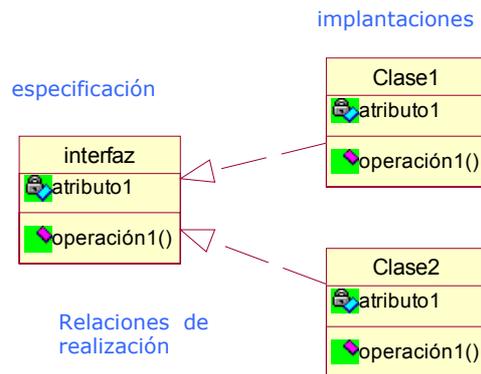


Figura 2.20 Notación de Realización en UML

Existe una notación reducida para mostrar interfaces sin su contenido y las clases o componentes que las realizan. Se muestra la interfaz con un círculo pequeño, unido al rectángulo de la clase por una línea continua (Figura 2.21).

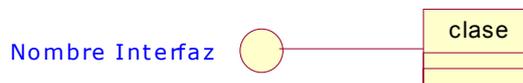


Figura 2.21 Notación reducida de Realización en UML

2.5.1.3.5 Relación de dependencia de uso

La dependencia de uso es una declaración en la cual el comportamiento de un elemento (el proveedor) afecta el comportamiento de otro elemento (cliente). En otras

palabras, establece que un elemento requiere la presencia de otro elemento para su correcto funcionamiento. Los estereotipos de uso incluyen la llamada y la instanciación, sin embargo, está abierto a otros tipos. La dependencia de llamada indica que un método, en una clase, llama a una operación en otra clase; la instanciación indica que un método, en una clase, crea una instancia de otra clase.

Una dependencia de uso se dibuja como una flecha discontinua desde la clase cliente hasta la clase proveedor, con la palabra clave para distinguirla (Figura 2.22).

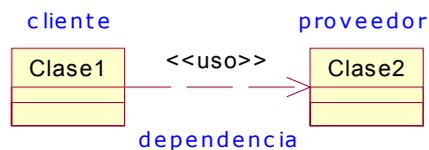


Figura 2.22 Notación de Dependencia de Uso en UML

2.5.2 Vistas de UML

En UML no hay ninguna línea entre los diferentes conceptos y construcciones, pero por conveniencia se puede dividir en varias vistas, donde una vista es un subconjunto de UML que modela construcciones que representan un aspecto del sistema.

En el nivel superior, las vistas se pueden dividir en tres áreas: clasificación estructural, comportamiento dinámico y gestión del modelo.

La clasificación estructural describe los elementos del sistema y sus relaciones con otros elementos. La clasificación de las vistas en este punto incluye la vista estática, la vista de casos de uso, la vista de implantación y la vista de despliegue.

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

La gestión del modelo describe la organización de los propios modelos en unidades jerárquicas. La gestión del modelo cruza las otras vistas y las organiza para el trabajo de desarrollo y control de configuración.

Resumiendo: dependiendo del área y la vista en que se encuentre, se desarrollan diferentes diagramas:

- ✓ Área Estructural.
 - Vista Estática: Diagrama de clases.
 - Vista de Casos de Uso: Diagrama de casos de uso.
 - Vista de Implantación: Diagrama de componentes.
 - Vista de Despliegue: Diagrama de despliegue.

- ✓ Área dinámica.
 - Vista de Máquina de Estados: Diagrama de estados.
 - Vista de Actividad: Diagrama de actividad.
 - Vista de Interacción: Diagrama de secuencia y diagrama de colaboración.
- ✓ Gestión del modelo.
 - Vista de Gestión del Modelo: Paquetes.

2.5.2.1 Vista estática

La vista estática modela conceptos principales de la aplicación. Esta visión no describe el comportamiento del sistema dependiente del tiempo. Los componentes principales de la vista estática son las clases y sus relaciones.

La visión estática se exhibe en los diagramas de clases (Figura 2.23), llamados así por que su objetivo principal es la definición de las características de cada una de las clases, las interfaces, las colaboraciones, relaciones de asociación, relaciones de generalización, de realización, de dependencia, de herencia y de agregación.

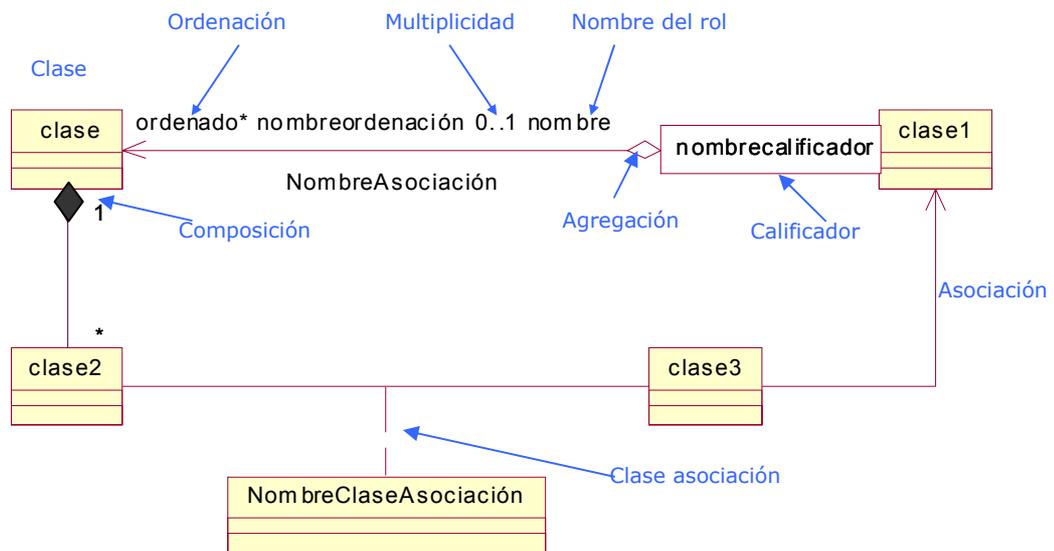


Figura 2.23 Diagrama de Clases en UML

2.5.2.2 Vista de casos de uso

La vista de casos de uso modela la funcionalidad del sistema según lo perciben los usuarios exteriores, llamados actores. El diagrama de caso de uso es muy útil para especificar que va hacer el sistema, su comportamiento, y no como va lo va a realizar.

El diagrama de casos de uso consiste en actores y casos de uso. Los actores son usuarios externos que interactúan con el sistema, aunque también se pueden ser otros sistemas. Los actores se dibujan como "muñecos" de palo (Figura 2.24).

Los casos de uso representan los escenarios que el sistema atraviesa en respuesta a un estímulo desde un actor. Éstos se dibujan como elipses.



Figura 2.24 Notación de Actor y Caso de Uso en UML

2.5.2.2.1 Relaciones de casos de uso

Existen varias relaciones para unir a los actores con los casos de uso y así describir el comportamiento del sistema (Figura 2.29).

La relación de asociación es una línea continua, denota que existe comunicación entre el actor y el caso de uso en el que participa (Figura 2.25).



Figura 2.25 Notación de Asociación en los Casos de Uso.

La relación de extensión se puede definir como la inserción de comportamiento adicional en un caso base que no tiene conocimiento sobre él. Puede haber varias extensiones del mismo caso de uso base, y pueden ser aplicadas conjuntamente. Esta relación se especifica con una flecha continua, donde la flecha está con dirección hacia el caso de uso base (Figura 2.26) además, debe llevar la leyenda de *extend* sobre la línea flecha.

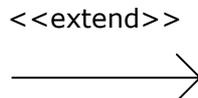


Figura 2.26 Notación de Extensión en los Casos de Uso

Por otra parte, aunque cada instancia de un caso de uso es independiente, la descripción de un caso de uso se puede descomponer en factores de otros casos de uso más simples. Un caso de uso puede incorporar el comportamiento de otros casos de uso como fragmentos de su propio comportamiento. Esto se llama relación de inclusión, se describe con una flecha continua con la leyenda *include* sobre la línea y está direccionada al caso de uso a ser incluido (Figura 2.27).

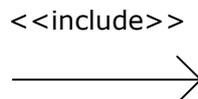


Figura 2.27 Notación de Inclusión en los Casos de Uso

Un tipo de relación muy utilizada también dentro de los diagramas de casos de uso es la generalización. Ésta es una relación entre un caso de uso general y un caso de uso más específico que hereda y añade propiedades. Se dibuja como una flecha continua, apuntando hacia el caso de uso padre o base (Figura 2.28).



Figura 2.28 Notación de Generalización en los Casos de Uso

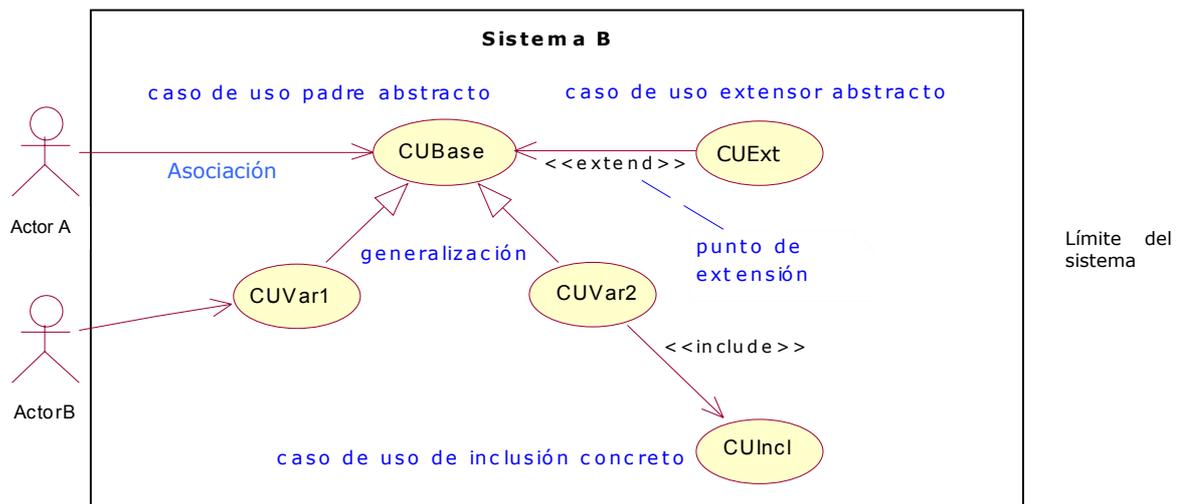


Figura 2.29 Diagramas de Caso de Uso en UML

2.5.2.3 Vista de implantación y vista de despliegue

La vista de implantación muestra el empaquetado físico de las partes reutilizables del sistema en unidades sustituibles, llamadas componentes. Esta vista muestra los elementos del diseño, tales como las clases, mediante componentes, así como sus interfaces y dependencias entre componentes. Los componentes son las piezas reutilizables a partir de las cuales se pueden construir los sistemas.

La vista de despliegue muestra la disposición física de los recursos de ejecución computacional, tales como las computadoras y sus interconexiones, llamados nodos.

2.5.2.3.1 Componente

Un componente es una unidad física de implantación con interfaces bien definidas pensada para ser utilizada como parte reemplazable de un sistema. Cada componente incorpora la implantación de ciertas clases del diseño del sistema.

Los componentes soportan interfaces y en ocasiones éstas se requieren de otros componentes. Una interfaz es una lista de las operaciones que una pieza de software o de hardware ofrece y puede realizar. El uso de las interfaces permite evitar dependencias directas entre componentes, facilitando una sustitución más fácil de nuevos componentes.

Un componente se dibuja como un rectángulo, con dos rectángulos pequeños a un lado. Puede ser unido por medio de líneas sólidas a los círculos que representan sus interfaces (Figura 2.30).

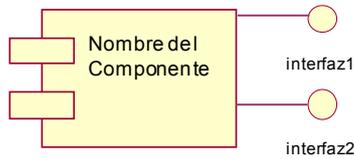


Figura 2.30 Notación de Componente e Interfaces en UML

2.5.2.3.2 Nodo

Un nodo es un objeto físico de ejecución que representa un recurso computacional, teniendo generalmente memoria y capacidad de proceso. Los nodos tienen estereotipos diferentes para reconocer recursos, tales como CPU, dispositivos, memorias etc. Además, los nodos pueden contener objetos e instancias.

Un nodo se representa mediante un cubo con el nombre del nodo y opcionalmente su clasificación. Las asociaciones entre nodos representan líneas de comunicación. Los nodos pueden tener relaciones de generalización para relacionar una descripción general de un nodo, con una variación más específica (Figura 2.31).

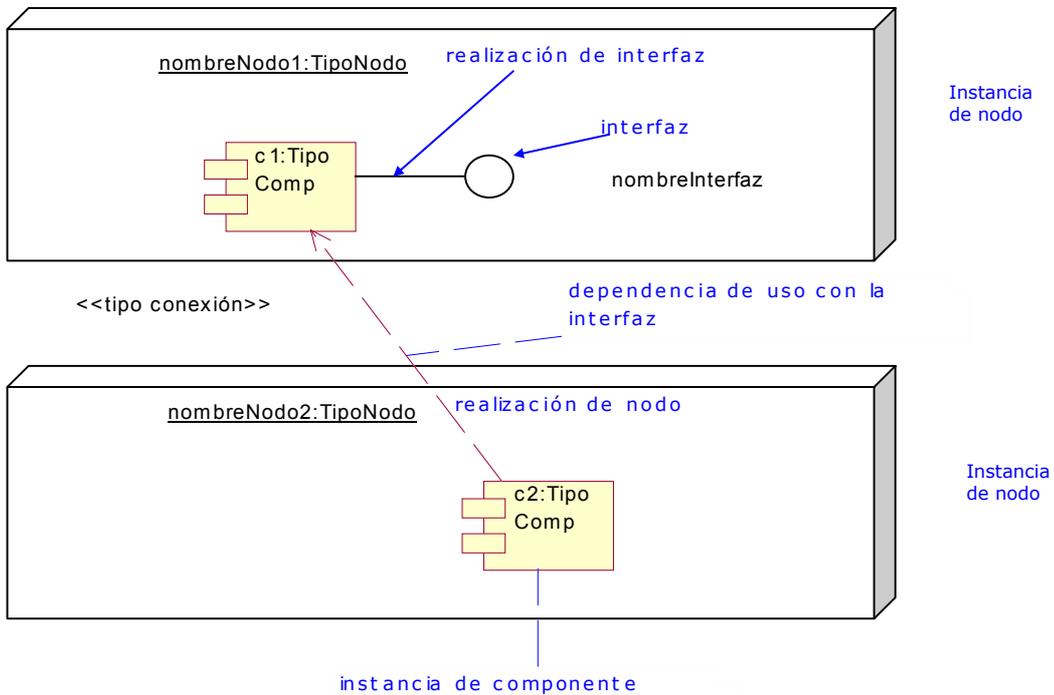


Figura 2.31 Diagramas de Componentes y Nodos en UML

2.5.2.4 Vista de máquina de estados.

Una máquina de estados es un modelo de todas las historias posibles de la vista de un objeto de una clase (Figura 2.34). El objeto se examina aisladamente. Cualquier influencia externa del resto del mundo se resume como evento. Cuando ocurre un evento, se puede desencadenar una transición que lleve el objeto a un nuevo estado. Cuando se dispara una transición, se puede ejecutar una acción a la transición.

Un evento es una ocurrencia significativa que tiene una localización en el tiempo. Ocurre en un punto en el tiempo y no tiene duración. Los eventos se pueden dividir en varios tipos, explícitos e implícitos: eventos de señal, eventos de llamada, eventos de cambio y eventos de tiempo.

En los eventos de señal, la señal es una entidad con nombre, que es como el vehículo de comunicación entre dos objetos. Las señales incorporan la comunicación unidireccional asíncrona, esto es, el remitente no espera que el receptor se ocupe de la señal, sino continúa con su propio trabajo independiente. Para modelar la bidireccionalidad se pueden utilizar varias señales, por lo menos una en cada dirección.

Un evento de llamada es la recepción de una llamada por un objeto que elige poner una operación en ejecución. Por otro lado el evento de cambio es un cambio en el valor de una expresión booleana. Por último, el evento de tiempo representa el paso del tiempo y se puede especificar de modo absoluto (hora) o de modo relativo (tiempo que transcurrió desde un evento dado).

Además, los estados describen un período durante la vida de un objeto de una clase. Un estado puede tener un nombre, aunque a menudo es anónimo y viene descrito simplemente por sus acciones.

En la máquina de estados, un conjunto de estados está conectado mediante transiciones. Las transiciones son procesadas por el estado del que salen. Los estados se muestran como rectángulos con las esquinas redondeadas (Figura 2.32).

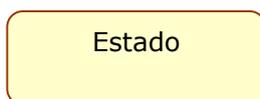


Figura 2.32 Notación de Estado en UML

Una transición es una acción que se ejecuta cuando pasa de un estado a otro. Existen dos tipos de transiciones: externa e interna. La transición externa es una respuesta a un evento que causa un cambio de estado o una transición a sí mismo, junto con la acción especificada. La transición interna es una respuesta a un evento que causa la ejecución de una acción, pero no causa cambio de estado o ejecución de acciones de salida o entrada.

Una acción de entrada, es aquella que se ejecuta cuando se entra en el estado y una acción de salida, es la que se ejecuta cuando sale del estado. La transición se representa por una flecha continua (Figura 2.33).

Figura 2.33 Notación de transición en UML

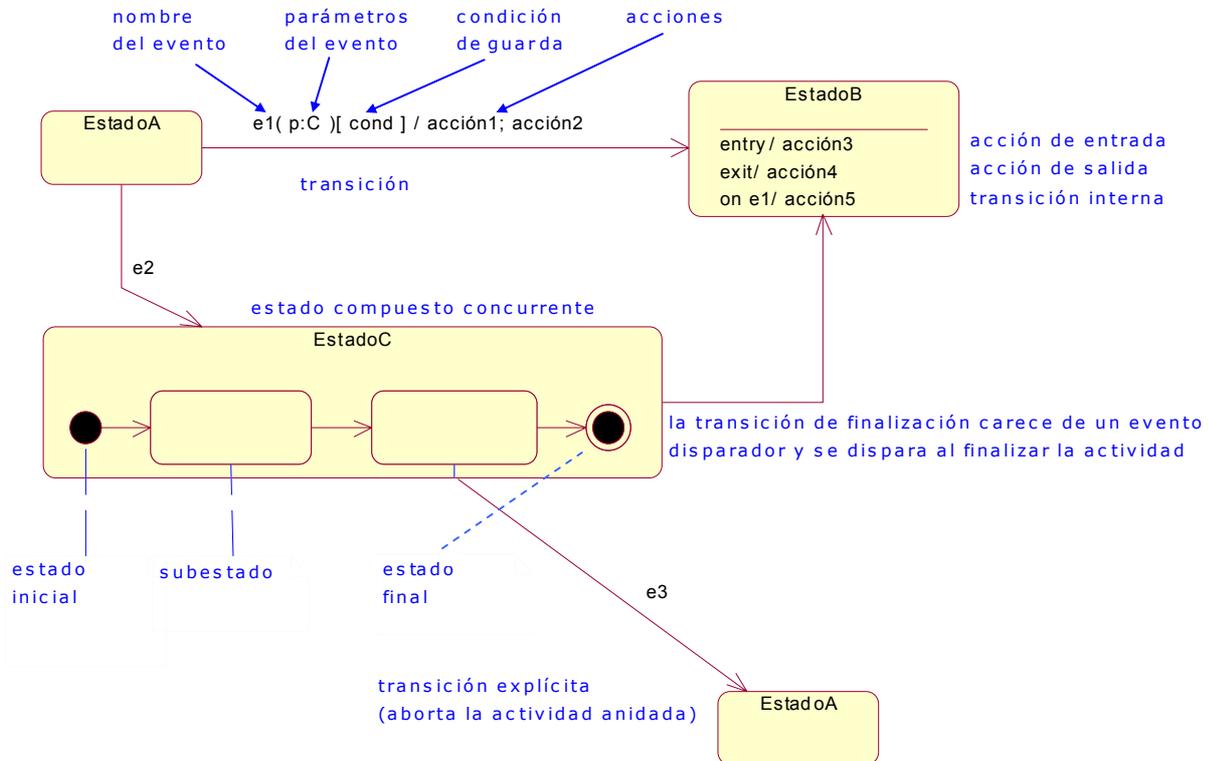


Figura 2.34 Diagrama de Estados en UML

2.5.2.5 Vista de interacción

La vista de interacción describe secuencias de intercambios de mensajes entre objetos para representar el comportamiento de un sistema. Esta visión proporciona una vista integral de su comportamiento.

Una interacción es un conjunto de mensajes dentro de una colaboración. Un mensaje puede ser una señal (una comunicación explícita entre objetos, con nombre y asíncrona) o una llamada (la invocación síncrona de una operación con un mecanismo para el control, que retorna posteriormente al remitente).

Una colaboración es una descripción de una colección de objetos que interactúan para llevar a cabo un cierto comportamiento dentro de un contexto. Detalla una sociedad de objetos cooperantes unidos para realizar un cierto propósito.

Existen dos tipos de diagramas en una vista de interacción: un diagrama de secuencia, que se centra en las secuencias de tiempo de los mensajes y un diagrama de colaboración, que se centra en las relaciones de los objetos que intercambian los mensajes.

2.5.2.5.1 Diagrama de secuencia

Un diagrama de secuencia representa una interacción como un gráfico bidimensional (Figura 2.35). La dimensión vertical es el eje del tiempo, que avanza hacia abajo. La dimensión horizontal muestra los objetos individuales en la colaboración. Durante el tiempo que existe un objeto se muestra por medio de una línea discontinua.

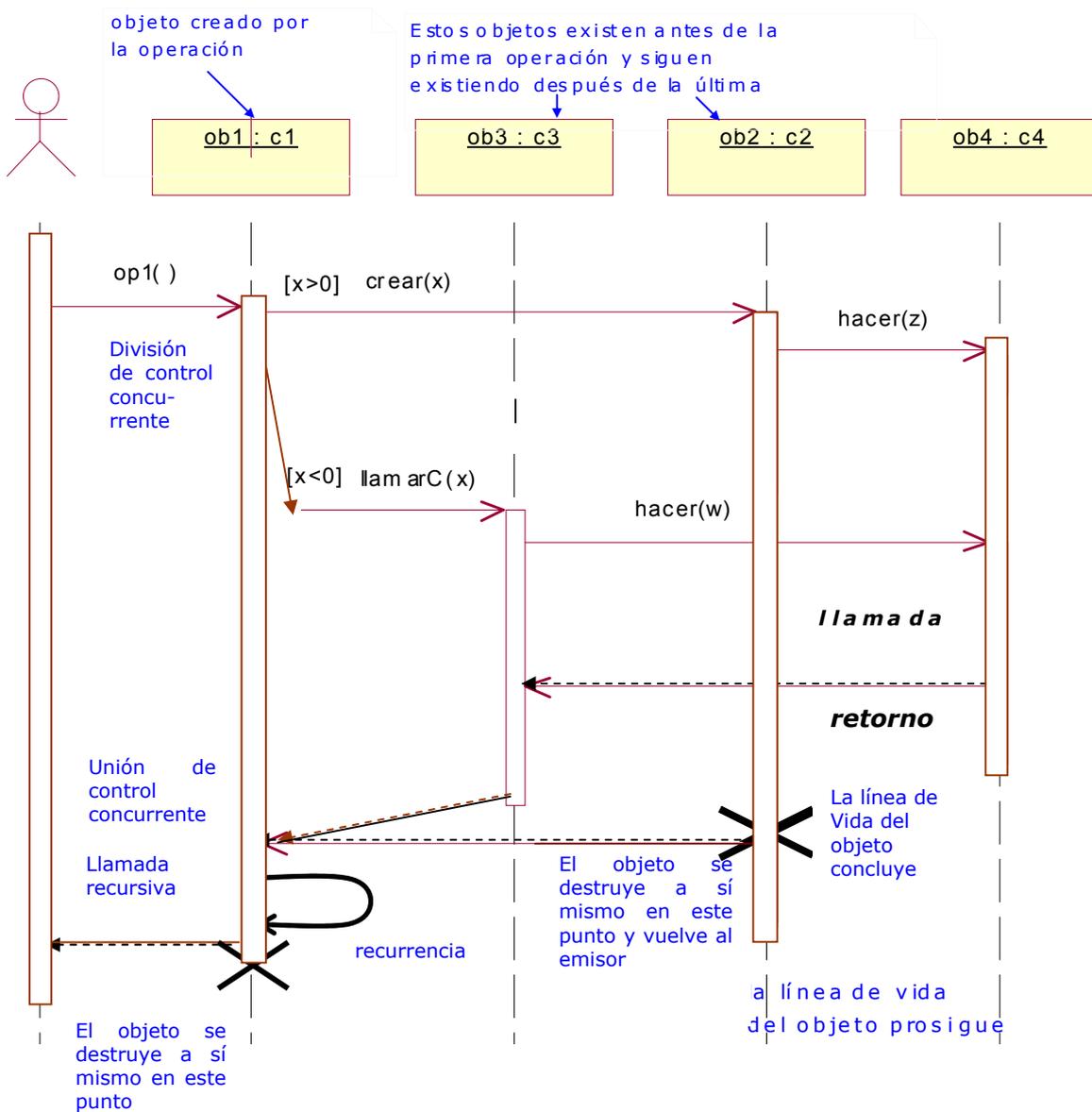


Figura 2.35 Diagrama de Secuencia en UML

Un mensaje se muestra con una flecha, desde la línea de vida de un objeto a la de otro. Las flechas se organizan en el diagrama en orden cronológico hacia abajo.

2.5.2.5.2 Diagrama de colaboración

Un diagrama de colaboración modela los objetos y los enlaces implicados en la interacción (Figura 2.36 y 2.37). Este diagrama muestra eventos (mensajes entre los objetos) y enumera el orden relativo de esos eventos.

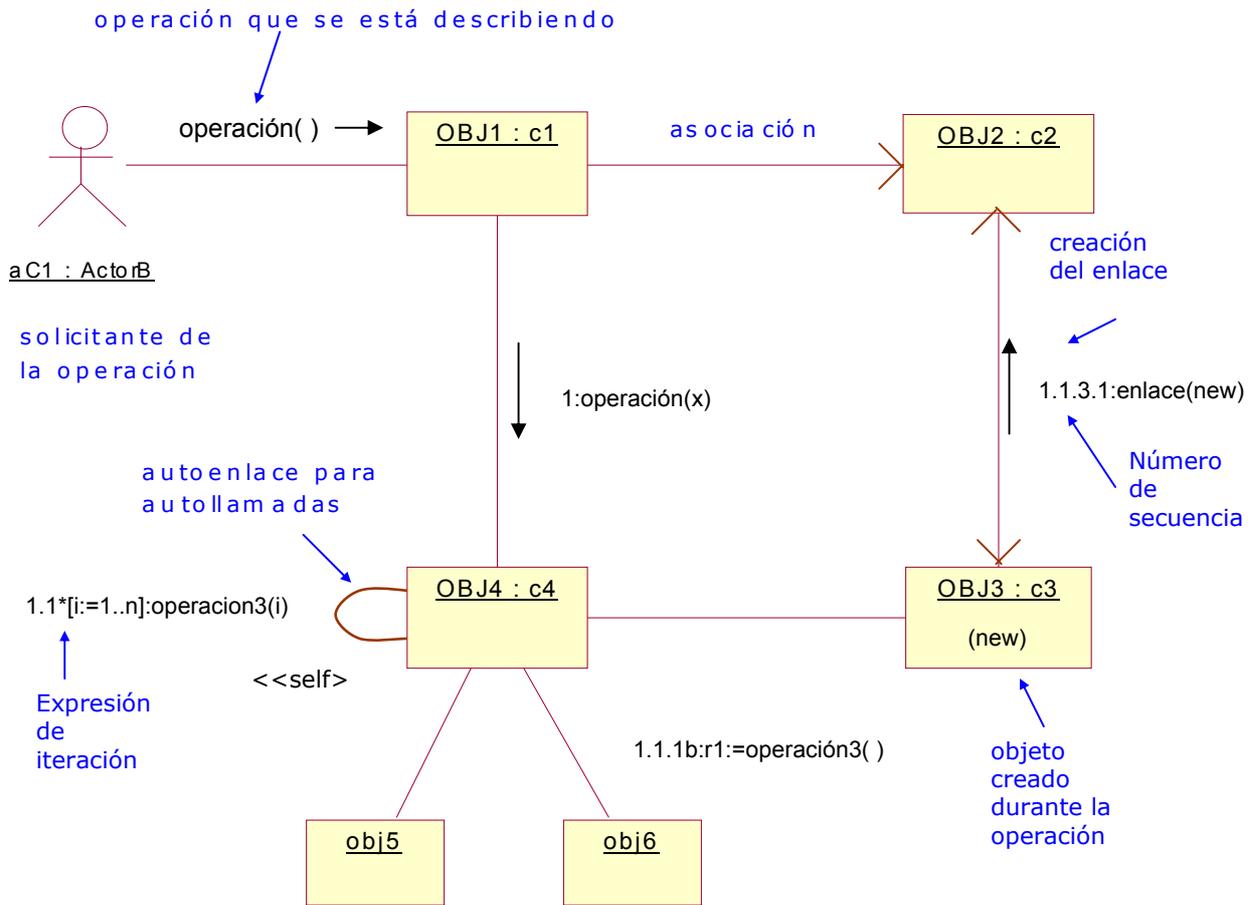


Figura 2.36 Diagrama de Colaboración en UML

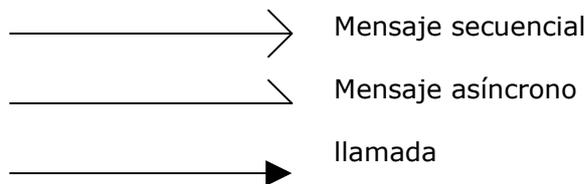


Figura 2.37 Notación de Mensajes en UML

2.5.2.6 Vista de actividades

Una vista de actividades es una variante de una máquina de estados. En esta vista se muestran las actividades implicadas en la ejecución de un cálculo. Su objetivo es mostrar relaciones temporales entre actividades, además de mostrar el flujo de trabajo entre los casos de uso. La vista de actividades se refleja en los diagramas de actividades.

2.5.2.6.1 Diagrama de actividades

Un diagrama de actividades describe grupos de secuencias. Los diagramas de actividades contienen estados de actividades. Éstos se representan como un rectángulo con los extremos redondeados que contiene una descripción de actividad.

Las transiciones simples de terminación se muestran como flechas. Las ramas se muestran como diamantes con múltiples flechas de salida etiquetadas. Una división o unión de control se muestra con múltiples flechas que entran o salen de una barra gruesa de sincronización (Figura 2.38).

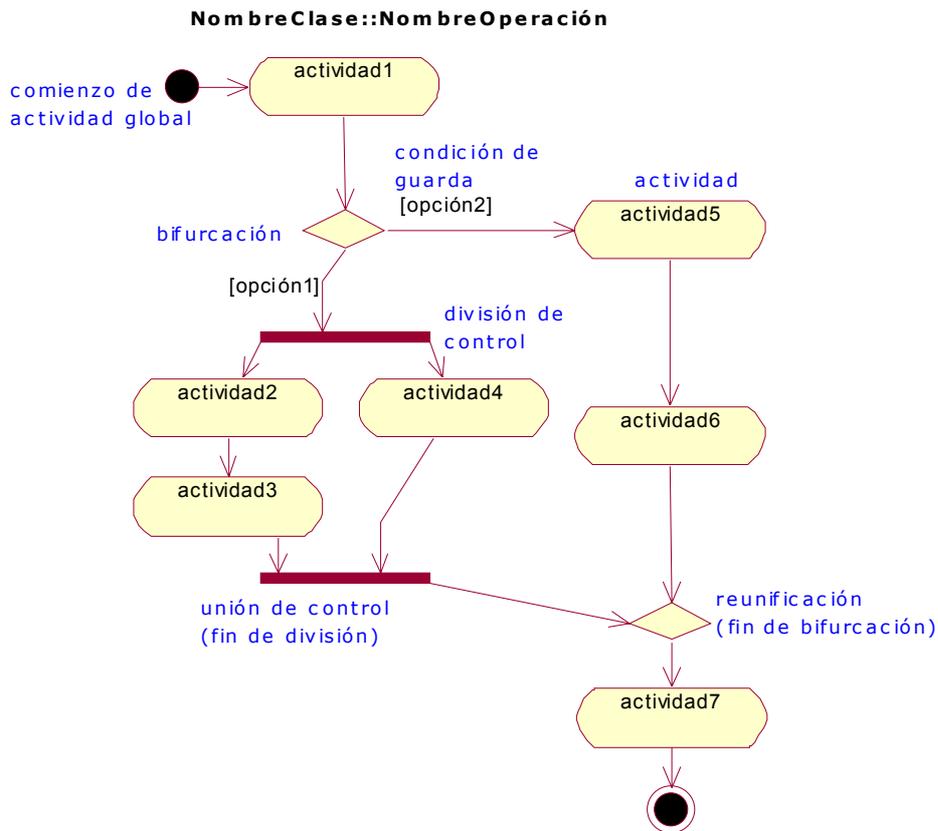


Figura 2.38 Diagrama de Actividades en UML

2.5.2.7 Vista de gestión del modelo

La vista de gestión de modelo, constituye la organización del modelo. Esta vista consiste en mostrar paquetes y relaciones de dependencia entre paquetes. Un conjunto de paquetes contiene elementos del modelo, tales como: clases, máquinas de estados y casos de uso.

Un paquete es una parte del modelo (Figura 2.39). Éstos favorecen al control del contenido de un modelo, así como las unidades para el control de acceso y el control de configuración. Además, cabe señalar que los paquetes pueden contener otros paquetes.

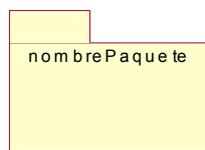


Figura 2.39 Diagrama de Actividades en UML

Las dependencias entre paquetes resumen dependencias entre los elementos internos, esto es, se derivan de las dependencias de los elementos individuales. Éstas se muestran como flechas con líneas discontinuas.

Existe también la relación de generalización y se describe con una flecha hueca del paquete hijo al paquete padre. Por otra parte, un modelo es un paquete que abarca una descripción completa de una vista particular de un sistema, mientras que un subsistema es un paquete que tiene piezas separadas, el cual representa generalmente la partición del sistema (Figura 2.40).

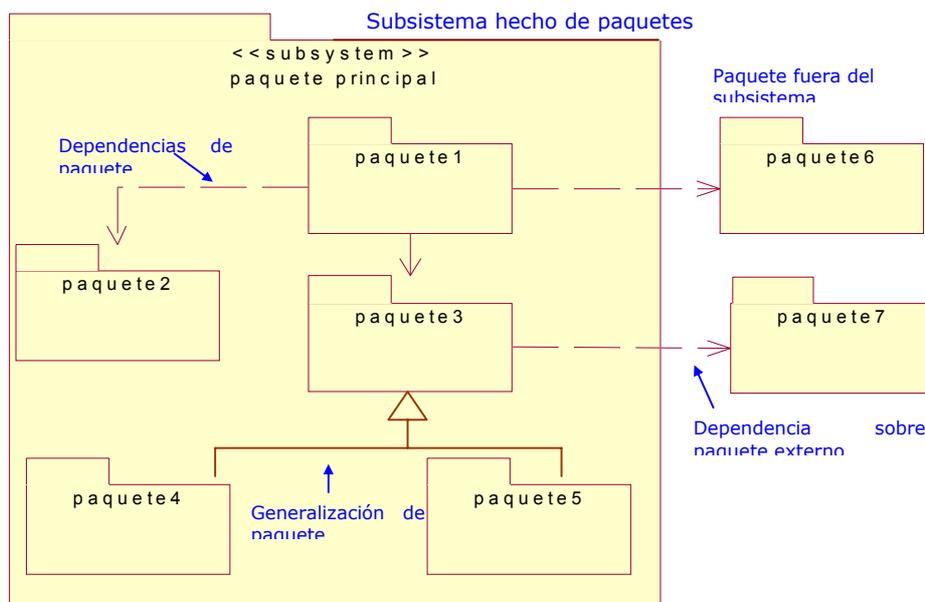


Figura 2.40 Diagrama de Actividades en UML

Las vistas y diagramas explicados para esta última metodología, hacen que sea la más completa y apropiada para proporcionar mejores técnicas y paradigmas en el desarrollo de sistemas, además de hacer más sencillo el modelado del sistema.

CAPÍTULO III

SOFTWARE LIBRE Y PROPIETARIO

3.1 Introducción

El mundo del software está creciendo vertiginosamente; cada vez hay más empresas dedicadas a este ramo. Actualmente tenemos la libertad de escoger entre dos opciones: software propietario o software libre.

El software propietario es el modelo clásico donde éste es provisto por una empresa de renombre, de alto costo y el funcionamiento del sistema es celosamente guardado por la compañía que lo produce. Además, es ilegal extender una pieza de software propietario para adaptarla a las necesidades particulares de un problema específico. En tal caso es necesario pagar una elevada suma de dinero a la compañía fabricante, para que sea ésta quien lleve a cabo la modificación.

Por otra parte el modelo de software libre se basa en la colaboración y la ayuda, esto es, un grupo de personas desarrolla una aplicación para resolver una necesidad específica, comparten el proyecto con la comunidad y ésta colabora en su desarrollo e incluso en su corrección, puesto que también los beneficia. En el proceso, el proyecto se transforma en un trabajo de todos y para todos, de esta manera, se generan líderes con autoridad moral sobre el proyecto, pero nunca dueños.

Ciertamente el software libre da la oportunidad de trascender tecnológicamente, ya que compartir el trabajo con la sociedad permite a pequeños empresarios tener herramientas sin restricciones de uso y con posibilidad de adaptar el sistema a problemas específicos. Además, devolviendo el sistema a la comunidad, el ciclo de este trabajo continúa, dando paso a un mayor desarrollo.

Tomando en cuenta las bondades que ofrece la utilización de software libre y dado que cada vez es más usado, ésta será la opción utilizada en el presente proyecto. Por tal motivo en este capítulo indicaremos las ventajas y desventajas de emplear tanto software libre como propietario, asimismo mencionaremos el perfil de México respecto al software libre.

3.2 Ventajas y desventajas de utilizar software propietario

El software propietario es aquel que se adquiere con una empresa específica, con lo que se crea una clase de dependencia tecnológica ya que es la única que se puede encargar del desarrollo del software; por otro lado, dicha compañía puede ofrecer varios tipos de servicios y de soporte para garantizar su producto.

3.2.1 Ventajas del software propietario

El software propietario generalmente es conocido por un gran número de personas, asegurando usuarios diestros en su utilización, de igual manera sus herramientas son soportadas por una gran cantidad de plataformas y de compañías, de todos tamaños, a niveles nacionales e internacionales. También existen certificaciones que proveen "credibilidad" a las soluciones creadas por cada compañía.

Las grandes empresas dedicadas a este ramo, por lo general cuentan con departamentos de control de calidad que llevan a cabo muchas pruebas sobre el software que producen y se destina una parte importante de sus recursos financieros a

la investigación para sus productos. Igualmente, los programadores encargados de desarrollar el software son muy capaces y con mucha experiencia.

Una ventaja más es que la documentación que se puede obtener sobre el producto es amplia, existiendo gran cantidad de publicaciones que facilitan el uso de estas tecnologías.

3.2.2 Desventajas del software propietario

Existe software propietario diseñado para aplicaciones muy específicas que no existen en ningún otro lado, más que con la compañía que lo produce.

En ocasiones es difícil aprender a utilizar eficientemente el software sin haber asistido a cursos de capacitación, o haber leído manuales de considerable costo. Por otra parte el código fuente del software propietario es conocido únicamente por la compañía que lo produce.

En cuanto al soporte técnico que ofrecen estas compañías, en la mayoría de los casos es costoso, insuficiente o tarda demasiado tiempo en ofrecer una respuesta satisfactoria.

Es ilegal extender un componente de software propietario para adaptarlo a las necesidades particulares. En caso de que sea vitalmente necesaria dicha adaptación, es necesario pagar una elevada suma de dinero a la compañía fabricante, para que sea ésta quien lleve a cabo la modificación a su propio ritmo de trabajo y sujeto a su calendario de proyectos.

Es ilegal hacer copias del software propietario. Si una dependencia contrata este tipo de software no es posible compartirlo con otras a menos que cada una de éstas contrate las licencias necesarias.

La mayoría de las grandes empresas se hacen dependientes de un solo proveedor, que generalmente es extranjero y esto finalmente le resta desarrollo tecnológico al país.

En el caso de que la compañía fabricante del software propietario deje de existir, desaparecen las posibilidades de darle mantenimiento al sistema y las ventajas que ésta ofrecía.

3.3 Ventajas y desventajas de utilizar software libre

La gran mayoría del software libre tiene como postulado la determinación de que el usuario no es un ignorante computacional. Parte del hecho de que confía en los conocimientos y habilidad del usuario para lograr su objetivo. Este punto de partida otorga gran poder y flexibilidad para los usuarios capaces y con ánimo de aprender, pero limita al resto de las personas que solo quieren facilitarse una tarea de manera inmediata.

3.3.1 Ventajas del software libre

El software libre tiene como característica primordial que cualquier persona tiene derecho de usarlo sin costo alguno, es de libre distribución y todos tienen la facultad de acceder a su diseño, aprender de él e inclusive de modificarlo. Lo anterior significa que es posible adaptarlo a necesidades específicas y redistribuirlo libremente en un tiempo de desarrollo menor en comparación del software propietario.

Usualmente en el software propietario hay que pagar una "licencia de uso" a la compañía productora (como el pago de derechos por el uso de una patente) y se está sujeto a las condiciones del fabricante. Normalmente estas condiciones no otorgan ningún derecho al usuario final, sin embargo, la utilización del software libre ahorraría el gasto por la adquisición de dichas licencias y finalmente, de esta forma, se combatiría efectivamente la copia ilícita de software.

El software libre tiende a ser muy eficiente y robusto ya que lo desarrolla mucha gente, siendo de interés general mejorar esta base común. Las aplicaciones son auditadas antes de ser usadas en procesos críticos, además del hecho de que las más populares se encuentran muy depuradas.

Anteriormente en este tipo de software la configuración del hardware y software no era tan intuitiva, aunque existía documentación para realizar esta tarea. Actualmente la configuración es tan sencilla como la del software propietario.

El éxito del software libre se debe en su mayor parte a Internet, el cual ha permitido que las personas interesadas en dicho software se pongan fácilmente en contacto con otras. De esta manera Internet actúa como un puente de comunicación que acelera su desarrollo.

Lo mencionado anteriormente, ha dado cabida a que se desarrollen sistemas de cómputo que compiten en casi todos los niveles con los sistemas propietarios.

3.3.2 Desventajas del software libre

En el software libre la curva de aprendizaje es mayor, esto es, es más difícil comenzar con un sistema totalmente nuevo y desconocido que con las aplicaciones conocidas y más utilizadas (por ejemplo, aplicaciones de Microsoft®). De igual manera, el software libre no tiene garantía proveniente del autor.

El costo del software libre podría ser únicamente lo que se necesita para capacitar a los recursos humanos y para la corrección de errores, aunque de no haber seleccionado la herramienta correcta, a la larga podría ser más costoso que haberse decidido por el software propietario.

Como hemos podido observar dentro de este campo en los últimos años, únicamente los proyectos importantes y de trayectoria tienen buen soporte, tanto de los desarrolladores como de los usuarios; sin embargo, existen muchos proyectos más pequeños y recientes que carecen del compromiso necesario por parte de los usuarios y/o desarrolladores para que sean implantados de manera confiable.

Una desventaja más: el usuario debe tener nociones de programación para la corrección de errores en caso de que quiera contribuir con el círculo del software libre.

3.4 Desarrollo en México

El desarrollo tecnológico en México, en cuanto al software libre se refiere, no ha tenido mucho impulso. En la mayoría de las empresas se utiliza el software propietario sin considerar al software libre como alternativa, esto significa que gran parte de las divisas que generan las empresas mexicanas salen del país, pues el proveedor de software es generalmente extranjero.

El papel del gobierno es muy importante en este punto. Éste debería ser el encargado de hacer consciente a la sociedad mexicana para impulsar el desarrollo tecnológico del país, siendo allí donde el software libre puede lograr un gran auge en México.

3.4.1 Perfil para el país

Los gobiernos actuales han evolucionado, del estado paternalista hacia una administración dinámica y preocupada por no aumentar el déficit público, además de eso, es necesario resaltar la importancia social del gobierno como rector y responsable de garantizar que el desarrollo integral del país sea sustentable.

Entre las actividades necesarias para garantizar el desarrollo sustentable está la educación. En las universidades el uso del software libre ayudaría muchísimo, no sólo en el costo de las licencias, sino también, al ser las universidades las principales promotoras de este tipo de tecnología, enseñándola, difundiéndola y desarrollándola, cabría la posibilidad de que con el conocimiento generado forje mejores profesionistas, así como mayores posibilidades de desarrollo para la sociedad mexicana, entre otros beneficios.

De igual manera las PyMES (Pequeñas y Medianas Empresas) se verían beneficiadas con esta tecnología, ya que al decidirse por esta opción los costos respecto a la administración de su negocio se verían notablemente reducidos. Más aún: las grandes empresas pueden utilizar software libre, reafirmando en el mercado como una opción factible inclinando la balanza a su favor.

Es conveniente que haya un mayor desarrollo en México, puesto que crear tecnología siempre ha sido sinónimo de crear riqueza. Fomentar el desarrollo tecnológico con software comercial propietario, bien puede dar lugar a la generación de riqueza, pero esto no garantiza que los recursos se queden en nuestro país.

Impulsar el desarrollo del software libre para México, no solamente significa crear riqueza económica e intelectual, sino también significa asegurar una posición como líderes de tecnología, y sobre todo, contribuir y fomentar el crecimiento de México.

CAPÍTULO IV

REDES DE COMPUTADORAS E INTERNET

4.1 Introducción

En esta sociedad cada vez más globalizada, las empresas modernas, instituciones y organismos se desenvuelven con la necesidad de establecer mecanismos de comunicación entre sus sistemas de información. Es aquí donde las redes de computadoras representan un papel preponderante.

Las empresas necesitan definir sistemas de comunicación entre sucursales, con sus clientes y proveedores, con prestadores de servicios, con sistemas gubernamentales, financieros y a través de Internet.

Por otro lado, las necesidades de comunicación deben considerarse con un enfoque integral, ya que actualmente requerimos transmitir voz, datos y video por los mismos medios y simultáneamente. Además, las soluciones que actualmente se desarrollan siempre deben mirar hacia un posible uso de Internet para establecer comunicación con los usuarios del sistema.

Conscientes de la importancia de este campo en el área de Ingeniería en Computación, examinaremos los conceptos básicos de las redes de computadoras, las topologías, el cableado estructurado, las tecnologías de interfaces de red, los elementos de conectividad, el modelo de referencia ISO/OSI, además de los servicios más demandados hoy en día en Internet.

4.2 Conceptos Básicos

A partir del surgimiento del microprocesador y a través de su constante evolución, las computadoras se han convertido en una herramienta básica en la vida moderna, y al mismo tiempo, las redes de computadoras son una necesidad incuestionable para resolver las distintas problemáticas en el quehacer cotidiano del hombre.

Con esta herramienta podemos compartir y explotar recursos tanto físicos como lógicos y además nos permiten la interacción y la comunicación entre usuarios, departamentos y corporaciones. Es claro que las redes de computadoras bien diseñadas e implantadas, garantizan la productividad del ser humano en todos los campos de las ciencias y tecnologías.

Comenzando con las definiciones que conlleva el presente capítulo, una **Computadora** es un dispositivo capaz de realizar cálculos y tomar decisiones lógicas a velocidades hasta miles de millones de veces más altas que los seres humanos.

El **Hardware** implica todo aquello que sea físico (circuitos integrados de la computadora, discos de almacenamiento, periféricos, cableado, etc.), y el **Software** implica todo aquello que sean programas (sistemas operativos, aplicaciones, paquetería, etc.).

En el campo de la computación, una **Red** es un conjunto de computadoras enlazadas entre sí y/o con otros equipos, cuya configuración permite que ésta sea un medio para transmitir, recibir, compartir y manejar información de manera confiable y segura. Una red tiene como objetivo principal compartir recursos físicos (equipos y sus periféricos) y recursos lógicos (archivos de datos y programas), actualizándolos, organizándolos y explotándolos.

La computadora que cuenta con los recursos a compartir recibe el nombre de **Servidor de la Red**, que auxiliado por el sistema operativo de la red viene a ser el "cerebro", dedicado a administrar los recursos y las comunicaciones entre las demás computadoras, mismas que reciben el nombre de **Estaciones de Trabajo**.

El servidor puede ser de dos tipos:

- Dedicado: Exclusivamente administra los recursos de la red.
- No dedicado: Además de administrar los recursos de la red, funciona como estación de trabajo.

Las características y configuración de la computadora que se define como servidor están en función de los requerimientos particulares de cada caso, generalmente es un equipo robusto tanto en hardware como en software. El servidor y las estaciones de trabajo se conectan al canal de comunicación por medio de una interfaz de red.

Las **Interfaces de Red** son dispositivos que permiten la interconexión de los nodos de la red. Según su especificación y normas, cada interfaz de red determina los protocolos de comunicación y la forma de interconexión (topología) de cada red.

El **Canal de Comunicación** es el medio físico por el cual se comunican los nodos de una red. Puede ser un cable dedicado a las comunicaciones (coaxial, par trenzado, fibra óptica, etc.), o medios inalámbricos (tecnologías de radio frecuencia, microondas, rayo láser, etc.). Este canal de comunicación determina la velocidad máxima de transferencia de información dependiendo del tipo de cable que se utilice.

El **Sistema Operativo** de una red, es el conjunto de programas que regulan su funcionamiento. Éste proporciona los elementos para establecer una interfaz con el usuario, para controlar y definir los niveles de seguridad, así como precisar la manera en que se comparten los recursos.

El **Software de Aplicación** son programas disponibles en la red para los usuarios. Las más importantes son: suites de productividad personal (procesadores de texto, hojas de cálculo, manejadores de archivos, presentaciones, etc.), manejadores de bases de datos y correo electrónico.

Otro concepto inherente a las redes es el término **Nodo**, el cual, desde el punto de vista de las redes, es cualquier dispositivo que se encuentre conectado a la red, por ejemplo: servidores, estaciones de trabajo, impresoras, routers, etc.

A continuación clasificaremos las redes dependiendo de su cobertura geográfica. Solo mencionaremos las más importantes para los fines de este capítulo, aunque es saludable aclarar que existen otras.

Una **LAN** (Local Area Network, Red de Área Local) es una red de computadoras interconectadas, distribuida en los límites de una oficina, un grupo de oficinas, un edificio o un grupo cercano de edificios. También son llamadas redes privadas de datos. Su principal característica es la velocidad de conexión, ya que el canal de comunicación está totalmente dedicado a la red.

Reconociendo la necesidad de contar con redes de mayor alcance que las redes de área local, se establecieron las **MAN** (Metropolitana Area Network, Red de Área Metropolitana) y las **WAN** (Wide Area Network, Red de Área Ampla).

Una MAN es una red que resulta de varias redes locales (LANs) interconectadas por un enlace de mayor capacidad o backbone (por ejemplo de fibra óptica) en varias zonas. Es el tipo de estructura de red que se utiliza, por ejemplo, en un campus universitario, donde se conectan los diversos edificios, casas de estudiantes, bibliotecas y centros de investigación. Una MAN ocupa un área geográfica más extensa que una LAN, pero más limitada que una WAN.

Una WAN es el resultado de la interconexión de varias redes locales localizadas en diferentes sitios (distintas ciudades o países), comunicadas a través de conexiones públicas (líneas dedicadas). La conexión puede ser física (un cable) o a través de un satélite.

Los demás conceptos relacionados con las redes de computadoras e Internet los iremos definiendo conforme avancemos en los apartados del presente capítulo.

4.3 Topologías de redes de computadoras

En el ambiente de las redes, **Topología** es la forma en la que se conecta el grupo de elementos que conforman una red. Mencionaremos únicamente tres tipos básicos de topologías: estrella, bus y anillo; aunque se puede sumar a estos tipos básicos, la topología de árbol, que es una conexión compuesta.

Para el estudio de la topología vamos considerar dos tipos: física y lógica. La topología física está determinada por la disposición de los elementos conectados en la red, mientras que la topología lógica la determina el protocolo de comunicación que está operando en la red, no importando la disposición física de los elementos, en otros términos se puede implantar un anillo lógico en un bus físico.

4.3.1 Topología de Estrella

En este tipo de conexión el elemento central es el servidor con sus periféricos (Figura 4.1). Se mantiene "preguntando" constantemente a cada estación de trabajo mediante una comunicación exclusiva y por turno si desea transmitir información, en caso afirmativo, la atiende y al terminar prosigue su interrogatorio permanente con otra estación. A esta regla de comunicación se le conoce como protocolo **POLLING** (poleo).

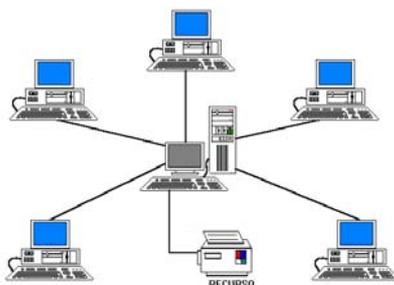
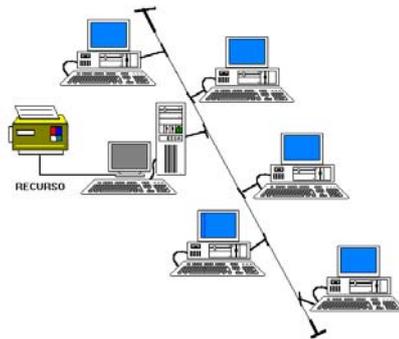


Figura 4.1 Topología de Estrella

4.3.2 Topología de Bus

Esta conexión se considera la más sencilla de todas, donde las estaciones de trabajo y el servidor están enlazadas por un solo cable, y la información viaja en ambos sentidos. Uno de los protocolos que se utiliza en esta topología es el **CSMA/CD** (Carrier Sense Multiple Access / Collision Detection). Con este protocolo, cualquier nodo transmite y espera que se le confirme que la información fue recibida correctamente, de otra forma, se detecta la posible colisión, espera un tiempo a que el canal esté desocupado y la información se transmite nuevamente (Figura 4.2).

Figura 4.2 Topología de Bus



4.3.3 Topología de Anillo

En esta conexión la información viaja ordenadamente en un solo sentido a través de un solo cable, describiendo un anillo imaginario. Las estaciones de trabajo y el servidor están conectadas en serie (Figura 4.3).

Una señal llamada "Token" (contenedor, a modo de estafeta) va circulando por la red y pasando por cada estación; si la primera resultó ser la solicitante, previa identificación, entrega la información, de lo contrario la deposita en "sobre cerrado" para que ésta a su vez la envíe a la siguiente, llevando la consigna de ser entregada hasta identificar a la estación solicitante. Cada estación de paso colecta información adicional enviándola a la siguiente, y así se pasa la señal cerrando ciclos circulares, por ello el protocolo apropiado para este caso se conoce como **Token Passing**.

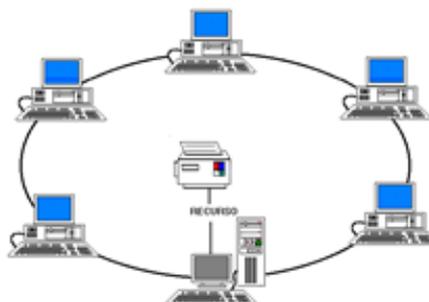


Figura 4.3 Topología de Anillo

Los factores de análisis que se deben considerar para la elección de la topología son:

- Protocolo de comunicación física.
- La flexibilidad de la red para añadir o eliminar nuevas estaciones de trabajo.
- La repercusión en el comportamiento de la red, considerando que se pueda tener una falla en una de las estaciones o nodos.
- El flujo de información que pueda transitar sobre la red sin que existan problemas asociados a retardos en la comunicación debido a una carga excesiva de transporte de información.
- Versatilidad en el diseño de cableado.

4.4 Tipos de cables

A continuación mostraremos los distintos tipos de cables que pueden ser utilizados en las redes de computadoras.

4.4.1 Cable coaxial

El cable coaxial consta de un alambre de cobre duro en su parte central que se encuentra rodeado por un material aislante. Este material está rodeado por un conductor cilíndrico que frecuentemente se presenta como una malla de tejido trenzado. El conductor externo está cubierto por una capa de plástico protector (Figura 4.4).

El cable coaxial produce una buena combinación de un gran ancho de banda con una excelente inmunidad al ruido. El ancho de banda que se puede obtener está en función de la distancia. Para cables de 1 km se puede llegar a velocidades de datos de hasta 10 Mbps, y en cables de longitudes menores, la velocidad es incluso mayor.



Figura 4.4 Aspecto del Cable Coaxial

4.4.2 UTP (Unshielded Twisted Pair)

El cable par trenzado sin blindar (UTP) es un cable compuesto de 8 hilos de cobre forrados individualmente por una cubierta plástica que es de un color diferente para cada cable; estos pares se encuentran forrados por una cubierta plástica que forma en sí al cable. Cada par está trenzado para evitar el ruido y, además, cada señal de cada par no se interfiere con la de otros pares dentro del mismo cable.

4.4.3 FTP (Foil Screen Twisted Pair)

El cable par trenzado, con doble forro metálico para una eficiencia de protección electromagnética mejorada (forro externo de PVC), es adecuado para aplicaciones de multimedia (simultáneamente video y datos) y aplicaciones para datos superiores a 100 Mbps o ambientes industriales ruidosos.

4.4.4 STP (Shielded Twisted Pair)

El cable par trenzado, con forro metálico por cada par separado (forro externo de PVC), es adecuado para aplicaciones de multimedia (simultáneamente video y datos) y aplicaciones superiores a 100 Mbps o ambientes industriales ruidosos (Figura 4.5).

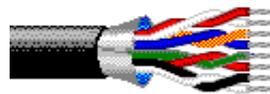


Figura 4.5 Aspecto del Par Trenzado

4.4.5 Fibra Óptica

La fibra óptica es un filamento de vidrio de alta pureza y diámetro microscópico. Tiene un forro de plástico que le da proyección física. Las fibras no se arman en cables por separado, se constituyen en cables de 6, 12, 24 y 48 fibras. La fuente de luz en una fibra óptica suele ser un LED o un diodo láser, pues ambos generan pulsos de luz cuando se les aplica una corriente eléctrica. El detector es un fotodiodo, que ante la recepción de un pulso de luz genera un pulso eléctrico (Figura 4.6).

El uso de fibra óptica para transmitir señales de comunicación tiene muchas ventajas importantes sobre los medios de comunicación convencionales, tales como:

- Baja pérdida en la energía de la señal.
- Baja tasa de distorsión en los pulsos de la señal transmitida.
- Un mayor ancho de banda que en UTP o coaxial.
- Las velocidades de transmisión son muy altas.
- No es susceptible de ruido o interferencia eléctrica o electromagnética.
- Soporta ambientes hostiles como: contaminación, salinidad, humedad o radiación.

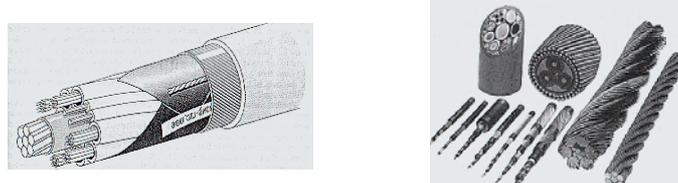


Figura 4.6 Aspecto de la Fibra Óptica

Un gran porcentaje de las fallas en las redes se debe a la mala selección del sistema de cableado estructurado, es por ello que la importancia de diseñar, instalar y administrar un adecuado sistema de cableado se ha hecho cada vez más crítica, conforme las tecnologías, las necesidades y las aplicaciones siguen evolucionando día con día.

4.5 Tecnologías de Interfaces de Red

La tecnología de interfaces de red avanza a pasos agigantados; en breve tiempo se tienen grandes avances. Por lo anterior, analizaremos brevemente las tecnologías de interfaces de red más utilizadas en el campo de las redes, las cuales son:

- Ethernet, Fast Ethernet, Giga Ethernet.
- Token Ring
- FDDI
- Frame Relay
- ATM

4.5.1 Ethernet

Comúnmente las estaciones de trabajo se conectan a un Hub o Switch. Enseguida mencionaremos las características generales de esta interfaz de red:

- Velocidad: 10 Mbps.
- Método de acceso: CSMA/CD.
- Número de nodos: 1 a 1023.
- Topología lógica: bus.
- Topología física: bus y estrella (utilizando HUB's).
- Medio físico:
 - Coaxial de 300 m a 500 m de alcance.
 - Par trenzado (con UTP 100 m, con STP 150 m y con FTP 150 m de alcance).
 - Fibra óptica.

4.5.2 Fast Ethernet

Es el estándar más utilizado actualmente, y las ventajas de utilizar Fast Ethernet son:

- Alto rendimiento.
- Velocidades de 100 Mbps.
- Tecnología basada en estándares.
- Migración a costo aceptable con máximo aprovechamiento del equipo ya existente (infraestructura de cableado, sistemas de administración de red, etc.)
- Soporte de los principales vendedores en todas las áreas de productos de red.
- Costo óptimo.

4.5.3 Giga Ethernet

Algunas de las características de Giga Ethernet son:

- Velocidad de 1000 Mbps.
- Basada en especificación de canal de fibra óptica.
- Conserva compatibilidad con Ethernet y Fast Ethernet.

4.5.4 Token-Ring

Este estándar surgió en 1985 aproximadamente, su creador fue IBM. Como su nombre lo indica, emplea una topología de anillo y el método de acceso con transmisión de señales.

Comúnmente, las estaciones de trabajo se conectan con par trenzado blindado o no blindado, hacia un concentrador de conexiones llamado unidad de acceso a multi-estaciones o MAU, con el fin de no tener que depender de la confiabilidad del cableado para el correcto funcionamiento de la red. Los MAU son aparatos confiables que además facilitan la instalación de la red, así como su mantenimiento.

Las características generales de token ring son:

- Velocidad: 4 ó 16 Mbps
- Método de acceso: token passing
- Nodos: 260 (1023 en su última versión)
- Topología lógica: anillo
- Topología física: estrella distribuida
- Medio físico: cable de par trenzado (UTP o STP) e IBM Tipo 2

4.5.5 FDDI (Fiber Distributed Data Interface)

La Interfaz de Datos Distribuida por Fibra utiliza token ring a 100 Mbps por medio de cable de fibra óptica, con distancia de transmisión de hasta 2 Km. El estándar FDDI cuenta con una arquitectura de anillo doble para proporcionar redundancia. FDDI provee comunicaciones para conmutación de paquetes y transmisión de datos en tiempo real.

Al hablar de redes interconectadas por medio de fibra óptica, generalmente se está hablando de FDDI. Como Token Ring, FDDI usa una topología con forma de anillo y un token eléctrico para pasar el control de la red de una estación a otra.

La mayor parte de las redes actuales con FDDI usan un doble anillo en donde cada nodo se une a los dos anillos independientes, transmitiendo los datos en sentidos opuestos. Esta configuración mejora la velocidad de transmisión así como la confiabilidad de la red.

4.5.6 Frame Relay

Frame Relay es un protocolo diseñado para manejar el aumento de información en las redes WAN y evitar retrasos, facilitar la interconexión de redes locales y mejorar sus tiempos de respuesta, además de realizarlo con transparencia y flexibilidad. Frame Relay transporta únicamente datos. Soporta velocidades hasta de 1.5 Mbps.

Frame Relay transporta datos dentro *de frames* y no maneja paquetes; ofrece casi cinco veces más velocidad en la conmutación; sus usuarios pueden compartir canales costosos; maneja con eficiencia un tráfico irregular e impredecible y suministra acceso de una sola línea a la red con conectividad lógica hacia cualquier otro destino, esto último reduce los requerimientos de hardware, y simplifica el diseño de la red.

Este protocolo no incluye un mecanismo de control de flujo que reduzca las ventanas de transmisión, sino que señala los problemas de congestión. Descarta los frames que lo provocan y deja que un protocolo de nivel más alto retransmita los mensajes correspondientes.

4.5.7 ATM (Asynchronous Transfer Mode)

Modo de Transferencia Asíncrono es una tecnología de comunicaciones de datos de conmutación de paquetes de banda ancha diseñada para combinar las características de los multiplexores por división de tiempo con retardo dependiente (TDM) y redes locales de retardo variable. Vamos a analizar más profundamente esta definición:

Una red de banda ancha es aquella, en que las señales viajan como señales de radiofrecuencia por canales separados. Se soporta la transmisión simultánea de datos, voz y video por varios canales.

La conmutación de paquetes es la capacidad de enviar pequeñas unidades de información (paquetes) por canales ATM. Los paquetes (llamados celdas en ATM) son situados en un canal ATM, y generalmente son mezclados con otros paquetes (multiplexados). En el extremo receptor, los paquetes son reensamblados.

La multiplexación por división de tiempo es un método para combinar señales separadas en una única transmisión de alta velocidad. Con ATM, se transmiten celdas provenientes de muchas fuentes. Pueden mezclarse, pero cada una tiene su dirección de destino específica. En la multiplexación por división de tiempo, las señales llegan en orden en intervalos de tiempo regulares. En otras palabras, todas las celdas son del mismo tamaño, tanto en bytes como en tiempo.

El retardo variable es habitual en las redes locales, debido a que cada método de red puede utilizar un tamaño de paquete distinto. ATM divide los paquetes largos para adaptarlos a su tamaño de celda y los envía por el canal de datos; éstos son reensamblados en el otro extremo.

En resumen, ATM ofrece un método para enviar simultáneamente información en paquetes procedente de varias fuentes sobre una línea de alta velocidad, donde es reensamblada y enviada a cada sistema destino. La característica más interesante de ATM es que se aplica a un amplio rango de comunicaciones de datos, desde el bus de datos de una central de cableado, hasta un sistema internacional de comunicación de datos. ATM maneja velocidades de transferencia desde 1.5 Mbps hasta 9.6 Gbps dependiendo del tipo de enlace que se utilice.

4.6 Elementos de Conectividad

Desde hace algunos años y cada día con mayor grado, se escucha hablar de conectividad. Los grandes monstruos de equipo de cómputo en toda su línea de equipos están preparados para la conectividad.

En el presente apartado se tratarán los aspectos fundamentales y básicos de la conectividad en el campo de las redes.

Conectividad es la capacidad de interconectar equipos de cómputo, de igual o diferente naturaleza. Esto es, se puede conectar una computadora personal a otra, una computadora a una red LAN, redes de computadoras a redes de computadoras, etc.

Cuando las necesidades informáticas de una empresa u organismo crecen, se llega a la necesidad de interconectar redes locales de computadoras con otras redes. Esto es posible de realizar por medio de una gran variedad de productos, que describiremos a continuación.

4.6.1 Puentes

De manera general se puede decir que un puente (bridge) permite la interconexión de redes ya que tiene la capacidad de comunicar diversos protocolos de red.

Los puentes trabajan muy cercanos al hardware de la red. Básicamente los puentes toman los paquetes de una red y los ponen en la otra. De hecho son más que un repetidor, ya que tienen suficiente información sobre los paquetes que maneja, aunque no conoce la estructura propia de éstos. El trabajo de un puente solo se realiza en los niveles uno y dos del modelo ISO/OSI (el modelo ISO/OSI se explicará en el siguiente apartado).

La principal ventaja de utilizar puentes para la interconexión de redes locales, es que se logran canales de alta velocidad. Su principal desventaja es que no se divide el tráfico entre las redes a conectar, por el contrario se incrementa, ya que el tráfico en la red es el generado por todos los nodos de las distintas redes conectadas por el puente.

Los puentes cubren tres necesidades básicas:

Unir Redes.- Por una parte, permiten que redes locales sean unidas entre sí, sin importar la topología de éstas. Consecuentemente es posible tener dos o más redes con topologías diversas trabajando entre sí mediante un puente.

Duplicar Distancias.- El uso de un puente permite también duplicar las distancias sin la necesidad de un repetidor, debido a que en cada extremo del puente se tendrá una red que podrá cubrir la distancia máxima, según su medio de comunicación.

Alarga Vida del Hardware.- La implantación de un puente también alarga la vida útil de interfaces de red con tecnologías en desuso, esto es, se puede hacer interactuar una pequeña red de tecnología atrasada con otra de tecnología más reciente mediante un puente, ya que éste se encargará de hacer la traducción de protocolos, a efecto de que la comunicación entre las dos redes se lleve a cabo de manera óptima.

4.6.2 Routers

Son dispositivos que generan la ruta que tomará la información a transmitir desde el nodo origen hacia el nodo destino. Los routers direccionan paquetes de una a otra red con base a la información de la capa tres del modelo de referencia ISO/OSI, siendo capaces de escoger el mejor camino a seguir para cada paquete (dependiendo de los protocolos que maneje el router). Una de sus principales funciones es dividir tráfico entre redes y así reducir la cantidad del flujo de información que se envía a través de

la red. Los routers también cubren las características de los puentes, es decir, nos permiten unir redes, duplicar distancias y alargar la vida del hardware.

Para conformar un router se requiere de un componente en hardware y un componente en software. En la parte de hardware se requiere de las interfaces de Red convencionales, estas pueden ser de igual o diferente tecnología (Ethernet, Token Ring, FDDI, etc.) que serán instaladas en una sola computadora que se habilitará como servidor de comunicaciones. El componente en software consiste en un conjunto de programas, instalados en el servidor de comunicaciones, que permiten la conversión de comunicación de protocolos (Token Passing, CSMA /CD, etc.) de las diferentes interfaces que fueron instaladas.

4.6.3 Gateways

El objetivo de un Gateway es lograr la comunicación desde cualquier estación de trabajo de una red a otro ambiente, a través de una sola línea, que regularmente es un equipo mayor.

Para que se comprendan las principales características de los Gateways, es necesario definir el término de sesión.

Una **Sesión** es el establecimiento de una conexión lógica entre un dispositivo y la computadora. Por ejemplo, en el momento en que una PC sea capaz de emular una terminal, se tendrá una sesión con el host, si al mismo tiempo se abre alguna ventana y además se tiene asignada una impresora, entonces se tendrán tres sesiones en uso.

Los términos anteriores son importantes, porque normalmente las capacidades de los Gateways a equipos mayores, están medidas en su capacidad de LU's, lo cual es sinónimo del número de sesiones que pueden manejar.

4.6.4 Concentradores (Hub's)

Son dispositivos que sirven para enlazar nodos de una red con topología física de estrella y topología lógica de bus. El Hub simplifica y centraliza el cableado de las redes locales, además de facilitar su administración y reducir fallas.

4.6.5 Switches

Son dispositivos de la capa de enlace de datos que permiten la interconexión de múltiples segmentos físicos de la red. Realizan una conmutación de circuitos, en la cual, se establece una conexión entre dos puntos de la red y se finaliza cuando la sesión de transferencia ha terminado. Su principal función es la segmentación de tráfico en la red, en base a las direcciones MAC (Media Access Control).

4.7 Modelo ISO/OSI

Las tecnologías que el hombre ha inventando para comunicarse, siempre han seguido ciertas normas o reglas para su aceptación en un grupo social que pueden ir desde una pequeña comunidad hasta todo una gran sociedad. En la época moderna las normas que rigen a las comunicaciones deben tener carácter universal. Hablando de comunicaciones digitales las normas o reglas universales están representadas por el

modelo *ISO/OSI (Organización Internacional de Normalización / Interconexión de Sistemas Abiertos)*.

Las normalizaciones en redes locales tratan de encuadrarse dentro de este modelo. Además, las redes locales deberán acoplarse a las redes públicas de área extendida, actualmente existentes y en permanente expansión. El modelo para la interconexión de sistemas abiertos se ha convertido en una referencia obligada para todo lo relacionado con la intercomunicación de computadoras.

Un sistema abierto es el conjunto de una o más computadoras con su software, periféricos y terminales, capaces de procesar y transmitir información. Es un modelo que está relacionado con las funciones que tienen que ser desarrolladas por el hardware y el software para obtener una comunicación fiable e independiente de las características específicas de la máquina. Es decir, está pensada para la interconexión de sistemas heterogéneos.

El sistema está compuesto por siete niveles, mediante los cuales dos sistemas informáticos se comunican entre sí. Las características del modelo las podemos resumir como sigue:

- Cada nivel está representado por una entidad de nivel. Los niveles equivalentes en dos sistemas diferentes se comunican de acuerdo con unas reglas y convenios denominados protocolos de nivel o protocolos de pares.
- Cada nivel proporciona un conjunto definido de servicios al nivel superior y a su vez utiliza los servicios que le proporciona el nivel inmediatamente inferior.
- La comunicación se realiza a través de los niveles inferiores, siendo el protocolo de pares una abstracción lógica de relación entre las dos entidades comunicantes.
- Si un nivel N desea transmitir una unidad de datos a otro nivel N homólogo en otro sistema informático, se la pasará al nivel inmediatamente inferior, el cual le añadirá información delimitadora propia y a su vez pasará esta información a su nivel inmediatamente inferior.
- En el sistema receptor cada nivel separará la parte del mensaje que le corresponde y pasará el resto a su nivel inmediatamente superior, que hará lo propio. Así el mensaje del nivel N viaja horizontalmente hasta su nivel homólogo en recepción.

Los tres primeros niveles tratan los protocolos asociados con la red de conmutación de paquetes utilizada para la conexión y pueden agruparse dentro del llamado bloque de transmisión. El nivel cuatro enmascara a los niveles superiores los detalles de trabajo de los niveles inferiores dependientes de la red, y junto con ellos forma el bloque de transporte. Los tres niveles superiores, del quinto al séptimo, son los usuarios del bloque de transporte y aíslan la comunicación de las características específicas del sistema informático.

4.7.1 Nivel uno: Físico

Este nivel engloba los medios mecánicos, eléctricos, funcionales y de procedimiento para acceder al medio físico. Es el encargado de la activación y desactivación física de la conexión (se incluyen las dimensiones físicas de cables y los tipos de señales que van a circular por ellos). En este nivel se encuentran definidos Ethernet, Token Ring, FDI, entre otros.

4.7.2 Nivel dos: Enlace

Un enlace de datos se establece siempre entre dos puntos físicos de conexión del sistema. En el caso de una red de datos de conmutación de paquetes, el nivel de enlace es responsable de la transferencia fiable de cada paquete al nivel de red. Las normas Ethernet y Token Ring también se encuentran definidas en este nivel.

4.7.3 Nivel tres: Red

Este nivel enmascara todas las particularidades del medio real de transferencia. Es el responsable del encaminamiento de los paquetes de datos a través de la red. Cada vez que un paquete llega a un nodo, el nivel tres de ese nodo deberá seleccionar el mejor enlace de datos por el que envíe la información.

Los paquetes de datos deben ir provistos de la dirección de destino. Por lo tanto, entre las funciones fundamentales del nivel de red se encuentran las de establecer, mantener y liberar las conexiones necesarias para la transferencia de los paquetes de datos. Además, son funciones de este nivel la definición de la estructura de datos de los paquetes, las técnicas de corrección de errores, la entrega en secuencia correcta al nivel de transporte de los paquetes recibidos, así como otras de reiniciación y control de flujo. Entre los protocolos más utilizados en este nivel se encuentran IP (Internet Protocol) e IPX (Internetwork Packet Exchange).

4.7.4 Nivel cuatro: Transporte

Este nivel es responsable de una transferencia de datos transparente entre dos entidades del nivel de sesión, liberando a dichas entidades de todo lo referente a la forma de llevar a cabo dicho transporte.

Los protocolos que maneja este nivel suelen llamarse protocolos end-to-end, o protocolos entre puntos finales, debido a que este nivel se encarga de realizar una conexión lógica entre dos estaciones de transporte de los sistemas informáticos que quieren comunicarse, independientemente de donde se encuentren éstos. Entre los protocolos de este nivel más utilizados se encuentran TCP (Transmisión Control Protocol) y SPX (Sequenced Packet Exchange)

4.7.5 Nivel cinco: Sesión

Su función es establecer y gestionar un camino de comunicación entre dos procesos del nivel de aplicación. Este nivel establece una sesión y se encarga de controlar la comunicación y sincronizar el diálogo.

La información que se envía se fracciona en pedazos y se generan unos puntos de sincronización. En caso de interrumpirse la sesión por alguna falla en la comunicación, los datos pueden ser recuperados y se conoce con precisión por ambos interlocutores hasta qué punto de sincronización la comunicación fue correcta. Al reanudarse la sesión no será necesario transmitir de nuevo toda la información, sino solamente a partir del punto donde se quedó el último paquete de información válido.

Este nivel también es responsable de dirigir el diálogo entre las entidades de nivel de presentación. Para ello, cuando se establece una conexión de sesión, es necesario

que ambos niveles cinco se pongan de acuerdo sobre el papel a desempeñar en la comunicación, por ejemplo, quién envía información, cuándo y por cuánto tiempo.

4.7.6 Nivel seis: Presentación

Este nivel se ocupa de la representación de los datos usados por los procesos de aplicación del nivel siete. Por lo tanto, si es necesario, realizará la transformación de los datos que reciba de o para el nivel de aplicación. Esto en el caso de que el proceso origen y el receptor tuvieran versiones de datos sintácticamente diferentes, pero también puede darse el caso de que, para una determinada aplicación distribuida exista un conjunto de caracteres normalizados diferentes de los del origen y el receptor, en cuyo caso, los niveles de presentación respectivos deberían de hacer las transformaciones necesarias.

Otra función que se puede encargar al nivel seis, es la de velar por la seguridad de los datos, siendo responsable de la encriptación de mensajes confidenciales antes de su transmisión. La función inversa será realizada por el nivel de presentación del sistema receptor.

4.7.7 Nivel siete: Aplicación

Este nivel se preocupa de proporcionar un conjunto de servicios distribuidos a los procesos de aplicación de los usuarios. El usuario se comunicará directamente con este nivel a través de la correspondiente interfaz o agente de usuario.

Dada la importancia de la familia de protocolos TCP/IP, vamos a hacer un apartado especial a este punto.

4.7.8 Familia de protocolos TCP/IP

TCP/IP (Transmission Control Protocol / Internet Protocol) es una familia de protocolos para interconectar computadoras de diversas naturalezas. Lo que se ha venido observando al paso de los años es que TCP/IP es un protocolo fuerte que no se ha visto desplazado por otros protocolos como se pensaba.

Desde su planeación, TCP/IP se pensó para ser independiente del medio físico de enlace, es esto precisamente lo que ha hecho que sea un protocolo ampliamente usado en enlaces de redes locales entre sí, o bien, con redes amplias WAN.

Los ambientes que usan TCP/IP se basan en que cada elemento de la red tenga su dirección IP. El propósito de lo anterior es identificar de forma única a cada nodo de la red.

A los nodos que son computadoras se les denomina *hosts* y los Gateways son el equipo que tiene realmente funciones de ruteador; es importante notar que la connotación de estos términos bajo TCP/IP es diferente a la que normalmente nos hemos referido.

Las direcciones de IP tienen como objetivo:

- Identificar de manera única cada nodo de una red o un grupo de redes.

- Identificar también a miembros de la misma red.
- Direccionar información entre un nodo y otro, aún cuando ambos estén en distintas redes.
- Direccionar información a todos los miembros de una red o grupo de redes.

IP hace el trabajo de llevar y traer paquetes entre todas las redes que estén unidas. Sin embargo, utilizar este protocolo no nos garantiza que éstos lleguen a su destino. Para remediar esto está TCP, el cual tiene funciones importantes, tales como:

- Secuenciamiento y reconocimiento de paquetes.
- Control del flujo de la información.

TCP partirá en paquetes la información y la enviará. A cada paquete se le asigna un número. El reconocimiento significa que cuando un nodo recibe varios paquetes, debe informar al que los está enviando que efectivamente los está recibiendo, de esta manera se logra un control sobre la información que se está transmitiendo.

La Entidad de TCP en cada extremo de una conexión debe asegurar que los datos que se entreguen a su aplicación local lleguen:

- Precisos
- En secuencia
- Completos
- Sin datos duplicados.

Existen una serie de tareas que TCP/IP realiza y que son de suma utilidad, tales como la emulación de terminales para entrar a diversos equipos, así como la transferencia de archivos entre computadoras. Estos detalles los veremos en el siguiente apartado.

4.8 Internet

Internet es una red de redes informáticas distribuidas por todo el mundo que intercambian información entre sí mediante la familia de protocolos TCP/IP.

Internet surgió de un programa de investigación realizado por la Agencia de Proyectos de Investigación Avanzadas de Defensa (DARPA) de los Estados Unidos de América sobre la conexión de redes informáticas. El resultado fue ARPANet en 1969. A principios de los ochenta se conectó con CSNet y MILNet, dos redes independientes, lo que se consideró como el nacimiento de Internet (International Network of Computers). También forma parte de esta red NSI (NASA Science Internet) y NSFNet (National Science Foundation Net).

Durante muchos años Internet ha servido para que muchos departamentos de investigación de distintas universidades distribuidas por todo el mundo pudieran colaborar e intercambiar información. Además, Internet ha comenzado a formar parte de los negocios y de nuestra vida cotidiana.

Internet ha hecho que el mundo empresarial se haya replanteado sus sistemas de comunicación internos y externos, y en la mayoría de los casos los haya encauzado vía Internet. Esto ha dado lugar a dos subgrupos dentro de Internet: Intranets y Extranets.

4.8.1 Intranet

Una Intranet es una red local que utiliza los mismos protocolos que Internet, independientemente de que esté o no conectada a Internet. Usualmente una Intranet está dotada de una velocidad bastante mayor que la habitual en Internet, lo que posibilita una comunicación muy fluida, incluso cuando se trata aplicaciones multimedia.

4.8.2 Extranet

Una Extranet es la unión de dos o más Intranets utilizando como puente Internet. Es un sistema barato y seguro, ya que sólo requiere que las Intranets estén conectadas a Internet, y si una determinada Intranet se viene abajo, no hay problema de rutas alternativas, ya que Internet es una red que casi siempre asegura una ruta completa.

4.8.3 Servicios en Internet

Las posibilidades que ofrece Internet se denominan servicios. Cada servicio es una manera de sacarle provecho a la red.

Hoy en día, los servicios más usados en Internet son: correo electrónico, conexiones remotas (Telnet), transferencia de archivos (FTP), grupos de noticias, World Wide Web (WWW), servicios de telefonía, entre otros.

4.8.3.1 Correo Electrónico (e-mail)

El servicio de correo electrónico se proporciona a través del protocolo SMTP (Simple Mail Transfer Protocol), y permite enviar mensajes a otros usuarios de la red.

Es tal vez el principal servicio de Internet, y sin duda el de mayor importancia histórica. El correo electrónico sirve para enviar y recibir mensajes de otros usuarios en cualquier lugar del mundo. La primera parte de una dirección identifica habitualmente a la persona y la segunda a la empresa u organización para la que trabaja, o al proveedor de Internet a través del que recibe la información (ej. `juan@servidor.unam.mx`).

Por este medio se puede enviar texto, gráficos, hojas de cálculo, algunos programas ejecutables, etc. El correo electrónico es, actualmente, común en la mayoría de las empresas y usado por la mayoría de las personas, es casi instantáneo, a diferencia del correo postal, y además muy barato.

4.8.3.2 Conexión Remota (Telnet)

El protocolo diseñado para proporcionar el servicio de conexión remota (remote login) recibe el nombre de Telnet, el cual forma parte del conjunto de protocolos TCP/IP y depende del protocolo TCP para el nivel de transporte.

El protocolo Telnet es un emulador de terminal que permite acceder a los recursos y ejecutar los programas de una computadora remota en la red, de la misma forma que si se tratara de una terminal real directamente conectada al sistema remoto.

Los programas clientes de Telnet deben ser capaces de emular las terminales en modo texto más utilizados para asegurarse la compatibilidad con otros sistemas, lo que incluye una emulación del teclado.

4.8.3.3 Transferencia de Archivos (FTP)

El protocolo FTP (File Transfer Protocol) forma parte de la familia de protocolos TCP/IP y sirve para proporcionar el servicio de transferencia de archivos en Internet. FTP depende del protocolo TCP para las funciones de transporte, y guarda alguna relación con Telnet.

El protocolo FTP permite acceder a algún servidor que disponga de este servicio y realizar tareas como moverse a través de su estructura de directorios, ver y descargar ficheros al ordenador local, enviar ficheros al servidor o copiar archivos directamente de un servidor a otro de la red. Lógicamente y por motivos de seguridad se hace necesario contar con el permiso previo para poder realizar todas estas operaciones.

4.8.3.4 Grupos de noticias

Los grupos de noticias son el servicio más apropiado para entablar debates sobre temas técnicos y cualquier tema actual. Se basa en el servicio de correo electrónico. Los mensajes que enviamos a los grupos de noticias se hacen públicos y cualquier persona puede enviarnos una contestación. Este servicio es de gran utilidad para resolver dudas.

4.8.3.5 World Wide Web (WWW)

La World Wide Web (habitualmente conocida como la Web) es un servicio basado en el sistema de hipertexto, el cual es un método para presentar información a través de la conexión entre documentos. Además de texto podemos encontrar fotos, sonido y video. Una de las mayores ventajas de la Web es la posibilidad de descargar recursos muy útiles sin costo alguno, como herramientas de desarrollo, procesadores de lenguajes y programas de prueba, así como información actualizada sobre todo tipo de tecnologías, ya que el mundo de Internet y de la Web están cambiando continuamente.

En resumen: hemos visto los temas más importantes relacionados con las redes de computadoras y los servicios de mayor relevancia proporcionados en Internet. Debido a que el sistema a desarrollar es una aplicación Web, este capítulo contribuye a definir los conceptos necesarios para el mejor entendimiento de la misma.

CAPÍTULO V

ESTUDIO INICIAL DEL SISTEMA

5.1 Introducción

Para desarrollar un proyecto es importante entender las necesidades del mismo, además de realizar un análisis de las herramientas necesarias para llevarlo a cabo. De esta manera se puede obtener la propuesta de la solución del sistema.

En el presente capítulo se examina el problema que originó el desarrollo del sistema (SIPIEX), así como el análisis de la metodología de desarrollo del software, las tecnologías seleccionadas para el desarrollo del software y por último la propuesta de la arquitectura de software con base a las tecnologías seleccionadas.

5.2 Planteamiento del problema

La Facultad de Ingeniería de la UNAM cuenta con la Unidad de Administración de Proyectos de Ingresos Extraordinarios (UAPIEX, abordada en el Capítulo I).

Esta unidad surgió por la necesidad de crear un sistema encargado exclusivamente de gestionar los proyectos que generan ingresos extraordinarios, esto significa que la mayoría de la información relacionada con los proyectos se concentre en dicha unidad y cualquier entidad involucrada en el proceso de los proyectos pueda consultarla sin entorpecer el flujo de la información, con la finalidad de hacer más transparente y fluido su manejo, además de la posible eliminación de entidades innecesarias.

Antes de la intervención de la UAPIEX existía un desconocimiento de detalles importantes de cada proyecto, esto es, cada entidad involucrada llevaba su propia información, por esto, no existía comunicación entre las instancias envueltas y originaba un desconocimiento de los avances obtenidos.

Otros inconvenientes eran: no existía un seguimiento adecuado de las variables cualitativas y presupuestales de cada proyecto (Ver Anexo A), además no se notificaba si un proyecto había sido aceptado para ser desarrollado o en qué etapa se encontraba. A esto hay que agregarle que no existían archivos históricos sobre los proyectos para su futura consulta.

Por lo anterior la UAPIEX se propuso desarrollar un sistema de información (SIPIEX) con el fin de proporcionar informes oportunos y confiables a las entidades involucradas en los proyectos, además de comunicar a la sociedad en general del desarrollo e investigación que se realiza en la Universidad (por medio de la Memoria Tecnológica) y que esto pueda convertirse en un estímulo para futuras inversiones.

Resumiendo, el SIPIEX dará a conocer los datos más relevantes de los proyectos, así como su progreso, proporcionando resúmenes de la información, admitiendo consultas posteriores al término de los mismos. De igual forma la implantación del sistema permitirá la consulta de la información vía Internet con el propósito de ser accesible desde cualquier lugar en cualquier momento.

5.3 Metodología de desarrollo de software

La metodología de desarrollo de software a considerar para el desarrollo del sistema es el Proceso Unificado. Se eligió esta metodología considerando que tiene las características y facilidades de uso óptimas para el desarrollo de nuestra propuesta.

5.3.1 Proceso Unificado

El Proceso Unificado (PU) agrupa las mejores técnicas para desarrollo de software en un procedimiento que permite a los desarrolladores disminuir los tiempos de liberación, además de predecir el comportamiento del mismo, lo cual conlleva a la reducción de riesgos y por ende a disminución de pérdidas económicas significativas.

El Proceso Unificado impulsa una visión común al interior de un equipo de desarrollo, lo que facilita la comunicación entre los miembros del equipo. Esto asegura la buena administración de actividades y proporciona a los líderes de proyectos guías específicas para establecer un desarrollo exitoso.

Este proceso apoya el desarrollo del software con guías completas e información sobre los diversos pasos involucrados en la captura de los requerimientos. El PU se puede adaptar a la mayoría de aplicaciones en la industria, además se ajusta a proyectos de variados tamaños y complejidad.

En general el Proceso Unificado establece una estructura que integra los ciclos, las fases, los flujos de trabajo, mitigación de riesgos, control y aseguramiento de calidad, gestión del proyecto y control de configuración, las cuales analizaremos a continuación.

5.3.1.1 Características principales

El PU contiene un conjunto de procedimientos, herramientas, técnicas y un soporte documental que lo han llevado a ser una de las metodologías de desarrollo mejor organizadas y planteadas, además de que permite una interacción continua con el usuario desde el inicio. Las principales características que identifica esta metodología son:

- Manejado por Casos de Uso
- Centrado en la Arquitectura
- Basado en componentes
- El lenguaje de modelado utilizado es UML
- Iterativo e incremental
- Gestión de Riesgos

Manejado por Casos de Uso

El Proceso Unificado es manejado por Casos de Uso (Capítulo II), desde el análisis de requerimientos hasta el mantenimiento. Por medio de los casos de uso se logra construir una guía de todas las actividades que realiza el sistema, que finalmente son el marco y la pauta de todo el desarrollo.

Centrado en la arquitectura

Está centrado en la arquitectura, esto significa que se basa en las vistas que pueda contener el sistema como son: vista lógica, vista de implantación, vista de actividades, etc. (Capítulo II). Por otro lado la arquitectura del sistema y su modelo de casos de

uso no son independientes ya que todo producto tiene funcionalidad (caso de uso) y forma (arquitectura).

Asimismo la arquitectura involucra otros elementos como es: el sistema de gestión de la base de datos, el sistema operativo, etc. Esta característica del Proceso Unificado nos ofrece una perspectiva global del sistema aunque no detallada.

Desarrollo basado en componentes

El sistema se divide en módulos para trabajar independiente y finalmente ensamblarse.

Lenguaje UML

El lenguaje de modelado que utiliza esta metodología es UML.

Iterativo e incremental

El PU es iterativo, esto es, el trabajo se divide en pequeños bloques de actividades, cada uno con un enfoque completo de desarrollo.

A medida que se termina una iteración se obtiene un avance del proyecto, esto es conocido como incremento.

Gestión de riesgo

Para asegurar un desarrollo sin mayores contratiempos se recomienda realizar un análisis de riesgo, conocido como gestión de riesgo, donde el objetivo es tener un mayor control sobre el proyecto y sobre los problemas que se puedan derivar del desarrollo.

5.3.1.2 Modelo iterativo

El Proceso Unificado sugiere abordar una iteración por flujos de trabajo. El flujo de trabajo está compuesto por: *recopilación de requisitos, análisis, diseño, implantación y pruebas*. Esto en realidad es un desarrollo completo conocido como mini proyecto, cuyo núcleo está constituido por una o más iteraciones.

A continuación se menciona el flujo de trabajo que sigue una iteración en el PU:

- La *recopilación de requisitos* realiza la planificación de dicho flujo de trabajo y se lleva a cabo un estudio de riesgos.
- El *análisis* estudia los casos de uso relevantes y escenarios de diseño de opciones arquitectónicas.
- El *diseño* crea un modelo utilizando como guía la arquitectura elegida.
- La *implantación* establece el desarrollo del código del diseño realizado y/o integración del nuevo código con el existente de iteraciones anteriores.
- Las *pruebas* son para realizar una evaluación y preparación de la entrega del producto obtenido, así como la entrega de documentación e instalación del prototipo.

En la Figura 5.1 se observa que en cada fase del Proceso Unificado se lleva a cabo una iteración y dentro de cada iteración pueden existir una o más iteraciones.

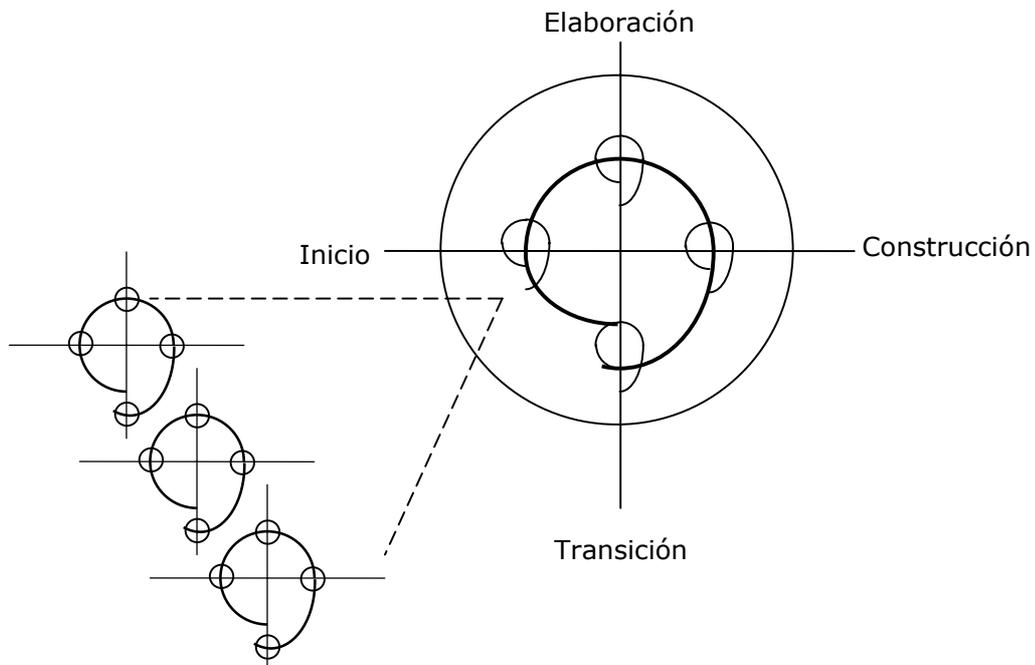


Figura 5.1 Iteraciones con las etapas del Proceso Unificado

Si una iteración cumple sus objetivos el proceso continúa con la siguiente iteración, en caso contrario, se deben revisar las decisiones previas e intentar una nueva aproximación. Por otro lado lo recomendable para no retrasar una iteración es eliminar algunos requisitos y dejarlos para la siguiente iteración.

Las iteraciones permiten reducir tiempos de entrega del proyecto debido a que la planificación de los riesgos más altos se realiza en las primeras fases, es decir, cuando el equipo de desarrollo está menos presionado y con mayor disponibilidad de analizar mejor estos aspectos. También el ritmo de trabajo es mayor cuando existen objetivos a corto plazo.

Los requisitos generalmente no se establecen completamente desde el principio, sino que se van detallando conforme se va avanzando en el proyecto, o sea, en cada iteración.

El mayor beneficio de la iteración es que reduce considerablemente los riesgos de costo, ya que si es necesario repetir una iteración sólo se pierde el esfuerzo de la misma y no el valor del producto completo.

5.3.1.3 Modelo iterativo e incremental

En la Figura 5.2 podemos observar que en cada incremento se distribuye equitativamente el esfuerzo. Además cada incremento representa un avance en el proyecto.

Un incremento es el resultado del crecimiento del proyecto por medio de iteraciones. Las iteraciones se refieren a pasos en el flujo de trabajo y los incrementos se refieren a partes que componen el producto final.

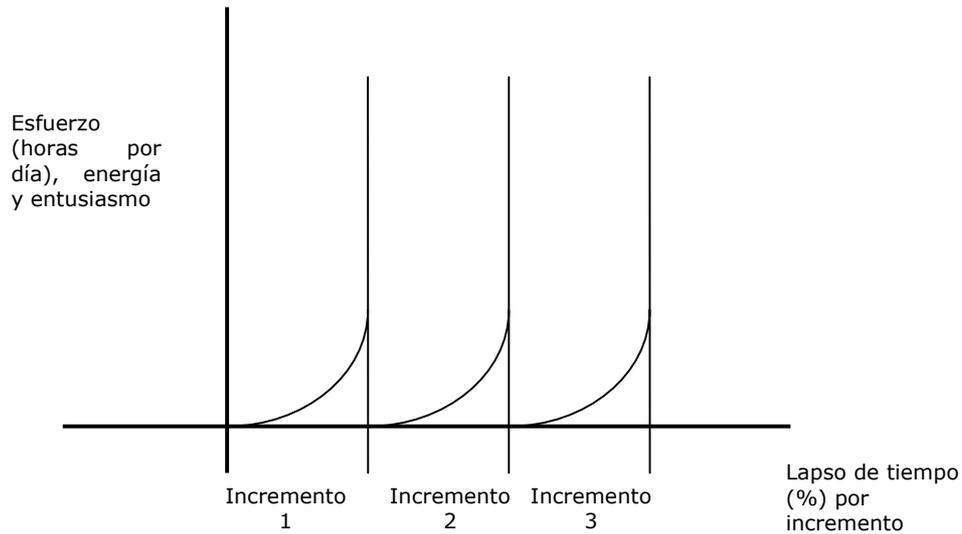


Figura 5.2 Línea de tiempo de un proyecto

5.3.1.4 Fases del ciclo de desarrollo

El Proceso Unificado considera que cualquier desarrollo de un sistema debe pasar por cuatro etapas: *Inicio*, *Elaboración*, *Construcción* y *Transición*, las cuales se muestran en la Figura 5.3, además de mostrar los diferentes flujos de trabajo concernientes a una iteración.

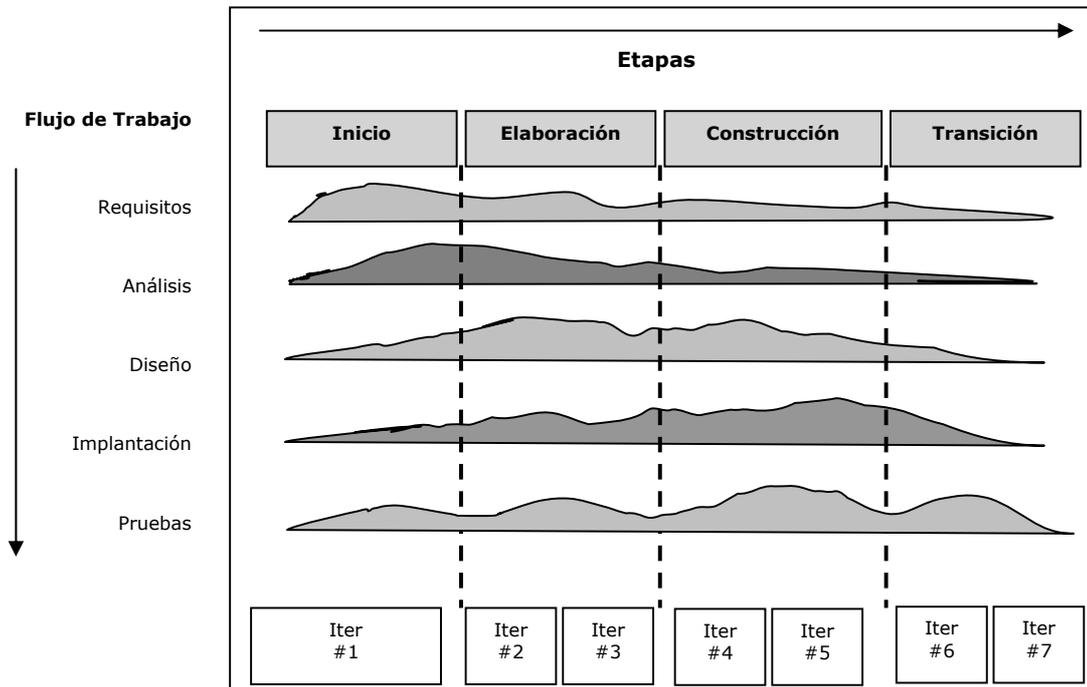


Figura 5.3 Etapas del Proceso Unificado con los flujos de trabajo

En términos generales los objetivos de las fases del Proceso Unificado son los siguientes:

- Inicio: En esta fase se definen los alcances y objetivos del proyecto.
- Elaboración: Aquí se define la funcionalidad del sistema y una arquitectura básica.
- Construcción: En esta fase se obtiene un producto listo para operar.
- Transición: Aquí se libera el producto y se entrega al usuario para su uso real.

Cada fase concluye con determinados *elementos de control*, estos se definen para evaluar el progreso del proyecto y considerar si se continúa con el desarrollo. Con este control se pretende comparar las expectativas del proyecto con lo que se está obteniendo, además de identificar los riesgos y evaluar su situación global.

Fase de inicio

El objetivo de la fase de inicio es desarrollar el análisis del sistema hasta el punto necesario para justificar la puesta en marcha del proyecto. Una de las principales características de esta fase es que presenta un desarrollo principalmente secuencial. En esta fase se incluyen los siguientes aspectos:

- Criterios de éxito (¿Cuál es la visión del sistema?)
- Evaluación del riesgo (¿Es viable el sistema a desarrollar?)
- Estimaciones de recursos (¿Se puede comprar o hay que fabricar el sistema? ¿Cuánto va a costar?)

- Plan de fases (¿Seguimos adelante, hay que planificar o en caso contrario paramos el proyecto?)

El flujo de trabajo de esta fase es de la siguiente manera:

- **Recopilación de requisitos:** Este flujo de trabajo incluye identificar, clasificar y detallar los casos de uso pertinentes en cada fase. Además de contemplar la elaboración de un análisis de riesgo.
- **Análisis:** Los objetivos generales son analizar los requisitos, refinarlos y estructurarlos en un modelo de objetos que sirva como primera impresión del modelo de diseño. En esta fase el resultado es un modelo inicial de análisis.
- **Diseño:** Su objetivo es esbozar un modelo de diseño de la arquitectura candidata, con el objeto de incluirlo en la descripción de la arquitectura preliminar.
- **Implantación:** Normalmente esta fase aborda la descripción de la arquitectura candidata y de ser posible se crea un prototipo exploratorio que muestra el uso del nuevo sistema.
- **Prueba:** En paralelo con las actividades de análisis, diseño e implantación, los encargados de las pruebas se van poniendo al corriente de la naturaleza general del sistema propuesto, van considerando qué pruebas requerirá y van desarrollando algunos planes provisionales de pruebas.

Por otra parte los elementos de control que se toman en cuenta en esta fase son:

- Una lista de características.
- Una primera versión del modelo del sistema que describe el contexto del mismo.
- Un esbozo de los modelos que representan una primera versión del modelo de casos de uso, el modelo de análisis y el modelo de diseño.
- Un primer esquema de la descripción de una arquitectura candidata, que mejora las vistas de los modelos de casos de uso, análisis, diseño e implantación.
- Posiblemente, un prototipo exploratorio que muestra el uso del nuevo sistema.
- Una lista inicial de riesgos y una clasificación de casos de uso.
- Los principios de un plan para el proyecto en su totalidad, incluyendo el plan general de las fases.
- Un primer borrador del análisis del proyecto, que incluye el contexto del proyecto y criterios de éxito.

Fase de elaboración

Su principal objetivo es plantear la arquitectura del sistema. En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema y acumulando la información necesaria para el plan de construcción.

Al comienzo de la fase de elaboración, se recibe de la fase de inicio un plan para esta fase, un modelo de casos de uso parcialmente completo y una descripción de la arquitectura candidata.

Esta fase analiza el dominio del problema, establece una base arquitectónica sólida, desarrolla el plan del proyecto, elimina los elementos de más alto riesgo del proyecto y detalla los flujos de proceso, actores, diagramas de colaboración y posibles cambios de estado.

El flujo de trabajo durante esta fase es de la siguiente manera:

- **Recopilación de requisitos:** El objetivo de este punto es encontrar y detallar los casos de uso y los actores, además de determinar su prioridad. También se construyen prototipos de las interfaces de usuario.
- **Análisis:** Se realiza el análisis de la arquitectura, de casos de uso, de las clases y de los paquetes.
- **Diseño:** Se diseñan los aspectos arquitectónicamente significativos del sistema, tal como están descritos en la vista de la arquitectura del modelo de diseño. Esta vista incluye subsistemas, clases, interfaces y casos de uso.
- **Implantación:** Este flujo prueba los componentes más importantes a partir del diseño. El resultado es lo que se conoce como la base para la arquitectura.
- **Prueba:** Aquí el objetivo es asegurar que los subsistemas de todos los niveles y de todas las capas funcionen correctamente, además se planifican y realizan pruebas. Cabe señalar que únicamente se pueden probar los componentes ejecutables.

Para esta fase los elementos de control son los siguientes:

- Preferiblemente obtener un modelo completo del proyecto que describe el contexto del sistema.
- Una nueva versión de todos los modelos: casos de uso, análisis, diseño, despliegue e implantación.
- La base de la arquitectura, obtenida a partir de la implantación de esta fase.
- Una descripción de la arquitectura, incluyendo vistas de los modelos de casos de uso, análisis, diseño, despliegue.
- Una lista de riesgos actualizada.
- El plan del proyecto para las fases de construcción y transición.
- Un manual de usuario preliminar.

Cabe mencionar que los riesgos más importantes se evalúan en las dos primeras fases y cualquier riesgo restante se evalúa al principio de la siguiente fase, es por ello que se evalúa si existe todavía la probabilidad de poner en peligro el éxito del proyecto.

Fase de construcción

El objetivo de esta fase es obtener el software en forma operativa, conocida como "versión Beta" y deberá cumplir con los requerimientos previamente planteados.

El flujo de trabajo es el siguiente:

- **Recopilación de requisitos:** En esta etapa se realiza una recopilación completa de los requisitos, se completan los casos de uso en detalle y se desarrolla el prototipo de interfaz de usuario.
- **Análisis:** Se considera de nuevo el análisis de la arquitectura, esto es, se obtienen la gran mayoría o la totalidad de los casos de uso.
- **Diseño:** Normalmente, en esta fase se llevan a cabo los casos de uso y diagramas restantes.
- **Implantación:** Se realizan las pruebas de unidad de todos los componentes, principalmente a partir del modelo de diseño. El resultado es la versión operativa inicial.
- **Prueba:** Se llevan a cabo las pruebas de integración del sistema y se evalúan.

Los elementos de control para la fase de construcción son:

- Obtener el plan de proyecto para la fase de transición.
- Elaborar el sistema ejecutable.
- Obtener todos elementos (diagramas UML, código fuente, ejecutables, casos de uso y toda la información derivada del desarrollo), incluyendo los modelos del sistema.
- Obtención de la descripción de la arquitectura, modificada y actualizada.
- Una versión preliminar del manual de usuario.

Fase de transición

En esta fase el sistema ha alcanzado suficiente capacidad operativa, como para considerar que ofrece confianza al ejecutarse en el ambiente del usuario, esto no quiere decir que el sistema este libre de errores, sin embargo esta fase es para delinear adecuadamente el sistema final.

En esta etapa se realizan pruebas de aceptación, manuales de usuario y el producto se entrega como software final. De cualquier manera puede ocurrir que el usuario final señale características necesarias para añadir al proyecto. En caso de ser pequeñas modificaciones se pueden llevar a cabo sin que esto afecte el sistema, en caso contrario, el Proceso Unificado recomienda que lo mejor es añadirla en el siguiente ciclo de desarrollo, esto es, en la siguiente versión del sistema.

Debemos aclarar que el esquema de actividades es diferente respecto a las otras fases. Aquí es necesario preparar la versión de pruebas de aceptación a partir de la versión inicial, así como instalar la versión en los lugares elegidos. De esta manera el flujo de trabajo queda de la siguiente manera:

- **Recopilación de requisitos:** En esta flujo, la actividad es baja, ya que el trabajo fuerte fue realizado en las fases anteriores.
- **Análisis:** Casi todo el desarrollo se llevo a cabo en la fase de construcción, de esta manera la actividad en esta fase también es baja, por ello se pueden corregir los problemas encontrados durante las pruebas en el ambiente del usuario.
- **Diseño:** Al igual que los puntos anteriores, en este flujo casi no hay actividad.
- **Implantación y prueba:** La atención se desplaza totalmente a la corrección de errores, a eliminar los fallos que ocurran durante las pruebas del usuario y de ser

necesario se realizan pruebas de regresión para asegurar que estas modificaciones no introduzcan nuevos defectos.

Los elementos de control de esta fase son:

- Se obtiene un software ejecutable, incluyendo el software de instalación.
- Se libera y se entrega al usuario para su uso real.
- Se incluye la instalación, configuración, entrenamiento, soporte y mantenimiento.
- Se entregan documentos legales como contratos, licencias, renunciaciones de derechos y garantías.
- Se obtiene la versión completa y corregida incluyendo todos los modelos del sistema.
- Se obtiene además la descripción completa y actualizada de la arquitectura.
- Obtención de manuales y material de formación del usuario final y del administrador del sistema.
- Referencias para la ayuda del cliente y actualizaciones.

5.4 Tecnologías para el desarrollo del software

Las herramientas fueron seleccionadas conforme a las necesidades del desarrollo de nuestro sistema de información. Los rubros que tomamos en cuenta fueron el uso y el costo que nos ofrecen. Al mencionar uso nos referimos a las características que manejan cada una de las herramientas y en cuanto a costo nos referimos a que se basan en software libre, ventajas de peso para nuestra elección. A continuación se describirán las características más importantes de cada herramienta de acuerdo a los fines de nuestro proyecto.

5.4.1 Manejador de base de datos

El manejador de base de datos (DBMS Data Base Management System) que seleccionamos para nuestro desarrollo es PostgreSQL.

PostgreSQL nació en 1985 como una iniciativa de cientos de desarrolladores a nivel mundial, para ofrecer un sistema poderoso, a la altura de otros DBMS comerciales, pero con licencia pública general, es decir libre para su explotación, mejoramiento y distribución libre.

PostgreSQL está considerado como una de las tecnologías libres más avanzada en el mundo ya que proporciona un gran número de características que únicamente se encontraban en manejadores de bases de datos comerciales.

Algunas de las características de PostgreSQL son:

- Aproxima los datos a un modelo objeto-relacional y es capaz de implantar herencia y arrays.

- Tiene soporte para vistas.
- Implantación del estándar SQL92/SQL99.
- Es totalmente compatible con ANSI SQL.
- Las interfaces de programación compatibles son: Java, C, C++, PHP, Perl, TCL y Python.
- Es altamente extensible, esto es, soporta operadores, métodos de acceso y tipos de datos definidos por el usuario.
- Soporta características avanzadas como las uniones (joins).
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de la información en la base de datos.
- Tiene soporte para store procedures (funciones), rules y triggers.
- Además tiene un control de concurrencia para evitar bloqueos innecesarios, es decir no hay que esperar para acceder a la información de la base de datos hasta que los registros estén disponibles.
- Tiene un sistema llamado Write Ahead Logging que almacena las transacciones y en caso de una caída de la base de datos es posible restaurarla.

5.4.2 Lenguaje de programación

El lenguaje de programación que decidimos utilizar fue Java, tomando en cuenta que es un software orientado a objetos utilizado para generar aplicaciones de propósito general. En los últimos años ha tenido gran relevancia en el uso de aplicaciones Web y debido a que nuestro sistema será accedido vía Internet consideramos que es la mejor opción.

Java es un lenguaje de alto nivel, considerado así por que su programación es muy cercana a nuestro lenguaje común. Debido a que es orientado a objetos permite abordar muchos problemas de programación de manera más sencilla.

Java tiene la ventaja de tener un ambiente completo de programación, esto es, cuenta con un compilador, un intérprete, un depurador, bibliotecas de clases para facilitar la generación de aplicaciones, entre otras cosas.

Pueden mencionarse muchas características de Java, sin embargo para los propósitos del nuestro proyecto mencionaremos sólo las más significativas para la implantación del mismo:

- **Orientado a objetos.** Java hace uso de la definición de entidades formadas por métodos y atributos que reciben el nombre de clases. Un objeto es la instancia de una clase. Este lenguaje soporta las tres características propias del paradigma de orientación a objetos que son: encapsulación, herencia y polimorfismo. C++ es uno de los lenguajes en el que se basaron para la creación de Java, sin embargo simplificó varios elementos como son: el manejo de memoria dinámica y la herencia múltiple.
- **Robusto e independiente de plataforma.** Se considera robusto por que fue diseñado para ser altamente fiable, esto lo logra proporcionando numerosas comprobaciones en compilación y tiempo de ejecución para detectar errores. También realiza la verificación de los tipos de datos y los byte codes. Los byte codes son el resultado de la compilación, creados para que puedan ser leídos en cualquier sistema operativo, lo que significa que Java crea un formato intermedio indiferente al sistema operativo utilizado, diseñado para transportar código a múltiples plataformas, dando como resultado un lenguaje independiente de plataforma. Esto último es conocido como portabilidad.

De igual manera sustenta el concepto de robustez prescindiendo de conceptos como son los punteros (utilizados por ejemplo en C o C++). Añade también cualidades muy útiles como son el garbage collector (reciclador de memoria dinámica), esto significa que no es necesario preocuparse por liberar memoria ya que el reciclador se encarga de ello.

- **Multitarea.** Esto significa que permite la ejecución concurrente de varios procesos o tareas, también conocidos como multi-thread, sin embargo, estas capacidades multi-tarea dependen en gran parte del sistema operativo en el cual se ejecuten.

Por otra parte, en un principio la idea fundamental de los navegadores era presentar páginas Web con información sin posibilidad de interactuar con ellas por parte del cliente. Generalmente eran páginas HTML almacenadas en un servidor web.

Sin embargo las tendencias en el mundo de la información han evolucionado pasando por las primeras páginas escritas en HTML, hasta aquellas que se generan al vuelo, como respuesta de una acción del cliente y cuyo contenido varía según las circunstancias, esto es, las aplicaciones web se encuentran orientadas hacia la creación de páginas dinámicas.

Asimismo el modo de generar páginas dinámicas también ha cambiado, desde la utilización de CGI (Common Gateway Interface), ASP, PHP hasta los Servlets y JavaServer Pages. Todas estas tecnologías son conocidas como Server-Side, o sea que se ejecutan en el lado del servidor web.

Un JSP es una página web con código Java incrustado y HTML estático, mientras que un Servlet es un programa que recibe peticiones y a partir de ellas genera una página web.

Nosotros haremos uso de los **JSP's** (JavaServer Pages) y **Servlets**, debido a que heredaron las características que ofrece Java y brinda características adicionales en el desarrollo de páginas dinámicas.

Su principal ventaja a diferencia de otras tecnologías utilizadas para generar páginas dinámicas, como son ASP, PHP, CGI, es que no comienzan a ejecutarse cada vez que se recibe una petición, sino que persisten de una petición a otra, de manera que no se pierde tiempo invocándolos. Esta característica les permite realizar otros procesos de manera más eficiente, por ejemplo, conexiones a bases de datos, manejo de sesiones, entre otros.

Además con la API (Application Program Interface) de los servlets la codificación para servicios específicos web resulta sencilla. En general los servlets pueden llamar a otros servlets y así distribuir de manera funcional el trabajo a realizar.

5.4.3 Servidor web

El servidor Web seleccionado para el desarrollo del proyecto es Jakarta-Tomcat. Este servidor es un contenedor para las tecnologías Servlets y JavaServer Pages.

Jakarta nace del desarrollo de soporte para las tecnologías Java orientadas al mundo web. Jakarta está compuesto por varios sub-proyectos que dan soluciones a problemas muy específicos en aplicaciones web y otros aspectos.

Entre estos sub-proyectos se encuentra Tomcat que es, quizá la herramienta más conocida del proyecto Jakarta. Tomcat es un servidor web que permite la utilización de Servlets y JSP's, y reemplaza al antiguo proyecto JServ.

El servidor Jakarta-Tomcat es un nuevo paso dentro del desarrollo de la tecnología orientada a la construcción de aplicaciones web. Esta tecnología es patrocinada por Apache.org, la misma que desarrolla el servidor web más utilizado actualmente, conocido como Apache.

5.4.4 Sistema operativo

La plataforma para nuestro sistema que elegimos es Linux Red Hat el cual, también es considerado software libre.

Linux es un sistema con varios años en el mercado, ofreciendo eficiencia, estabilidad, disponibilidad, robustez y mucha documentación, por lo que es la opción óptima para desarrollar nuestro sistema. Aunado a esto es compatible con nuestras herramientas elegidas, por ejemplo PostgreSQL ofrece un excelente rendimiento en esta plataforma.

Entre las características principales que presenta Linux tenemos:

- **Multitarea.** Se pueden ejecutar varios programas que compartirán el tiempo del procesador, pareciendo que estos programas se ejecutan simultáneamente. No se producen errores en las ejecuciones debido a la planificación de las mismas.
- **Multiusuario.** Permite que varios usuarios accedan a la misma máquina, además la ejecución de los procesos están protegidos.
- **Multiplataforma.** El kernel de Linux se puede utilizar en múltiples arquitecturas, como las antiguas 386, 486, Amiga, Atari y más actuales como Pentium, entre otras.

- **Estable.** No se presentan fallas generales del sistema ya que aísla los procesos y su ejecución; gracias a esta cualidad es una buena opción para ser utilizado como servidor.
- Debido a su buen rendimiento como servidor, y por las características de conectividad, soporta protocolos como TCP/IP, IPX o Appletalk, lo que le permite comunicación con otros protocolos.
- Proporciona seguridad en redes como: firewalls, métodos de encriptación, entre otras técnicas de seguridad.

5.5 Propuesta de arquitectura del sistema

La arquitectura permite representar la estructura de un sistema y sirve de medio de comunicación con los diferentes interesados en el proyecto, por ello no debe ser demasiado compleja.

Para que ésta se convierta en una herramienta útil dentro del desarrollo y mantenimiento de los sistemas, es necesario que contemos con una manera precisa de representarla, en nuestro caso la metodología UML nos ofrece estas ventajas.

En esta sección vamos a exponer los siguientes puntos:

- Arquitectura preliminar del sistema
- Tipos de clases a utilizar en el sistema
- Acoplamiento y Cohesión
- Modelo multicapas
- Modelo MVC
- Despliegue de la arquitectura

5.5.1 Arquitectura preliminar del sistema

En la Figura 5.4 se muestra la arquitectura preliminar del sistema, utilizando las tecnologías expuestas anteriormente.

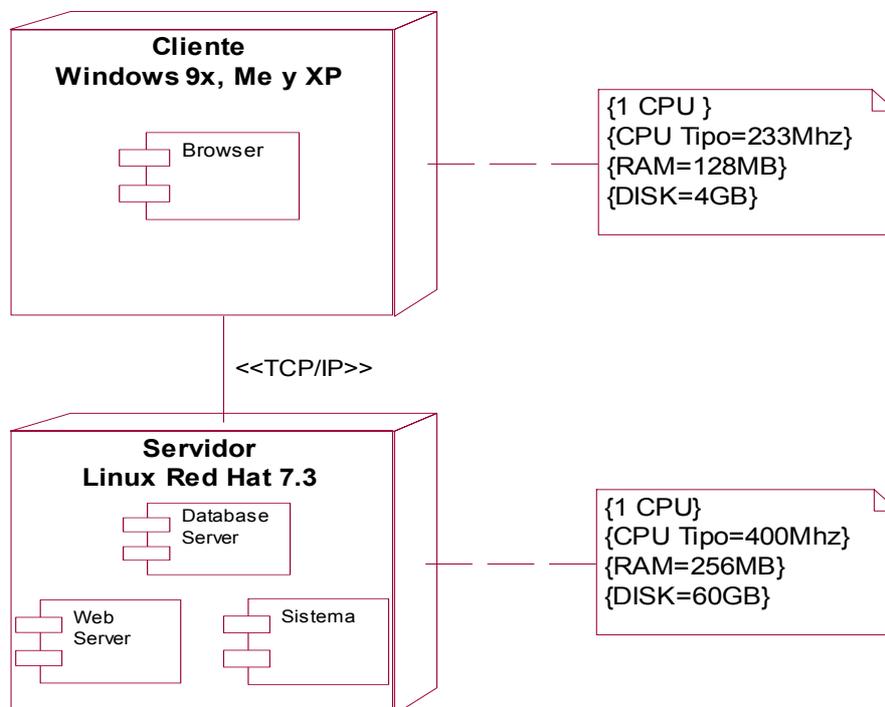


Figura 5.4 Arquitectura preliminar del SIPIEX

5.5.2 Tipos de clases a utilizar en el sistema

En nuestro sistema utilizaremos distintos tipos de clases, como son: de entidad, de límite y de control, las cuales definirán nuestro marco de trabajo del sistema y serán descritas enseguida:

- **Entidad:** Las clases de entidad representan el núcleo de la aplicación. Éstas mantienen la información persistente, lo que significa que se almacena en archivos o, en nuestro caso, en una base de datos relacional, para que esté disponible aun después del término de la aplicación. Esto no implica que una clase de entidad se convierta en una entidad de una base de datos, debido a que el mapeo *no siempre* es uno a uno. En nuestro caso, estas clases terminarán siendo JavaBeans. Básicamente un JavaBean es un componente reusable de java que maneja objetos llamados beans y éstos pueden tener comunicación con otros ambientes que soporten JavaBeans.
- **Límite:** Este tipo de clases sirven para delimitar a los actores externos que quieren interactuar con el sistema y las clases de entidad. La mayoría de estas clases son componentes para la interfaz del usuario, las cuales tienen la apariencia de pantallas usadas para interactuar con la aplicación. En nuestro proyecto estas clases serán JavaServer Pages.
- **Control:** Las clases de control coordinan las actividades necesarias para llevar a cabo una trayectoria a través de un caso de uso. Este tipo de clases normalmente

contienen información que no es persistente, por lo que existen solo el tiempo necesario para llevar a cabo la operación deseada.

La Figura 5.5 muestra cómo interactúan los tres tipos de clases descritos.

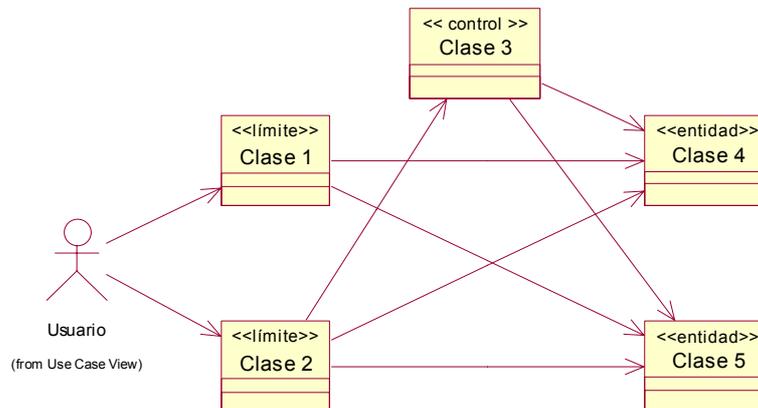


Figura 5.5 Interacción entre clases de entidad, límite y de control

5.5.3 Acoplamiento y cohesión

Las clases son consideradas *poco acopladas* si las asociaciones o dependencias entre ellas son mínimas. Si el modelo necesita una asociación entre clases se debe agregar, sin embargo, si una nueva clase, generalmente una clase de control o de límite, se puede utilizar para enviar los mensajes necesarios y así reducir el número de asociaciones, entonces hay que seguir ese esquema. Empleando esta ruta se podrán extraer de manera independiente clases o componentes y reutilizarlos en futuros proyectos si fuese necesario.

Existen asociaciones que por su propia naturaleza permiten que las clases sean poco acopladas, por ejemplo una relación de agregación/composición (Ver Capítulo II). Este tipo de relaciones previene a sus componentes de futuros cambios y hace la solución mucho más flexible.

Un elemento de diseño es considerado que presenta *alta cohesión* si realiza una labor única en el sistema, no desempeñada por el resto de los elementos y auto identificable. Ejemplo claro de baja cohesión son las clases que realizan demasiadas operaciones.

Las clases deben de planearse para que el trabajo que realicen no sea demasiado complejo. En caso de tener clases muy complejas se pueden crear otras clases para distribuir el trabajo que realizaba una sola, así las clases resultantes realizarán los procesos de forma más sencilla, aunque esto signifique que sólo realicen una tarea. En nuestro sistema las clases tendrán como propósito ser poco acopladas y con alta cohesión.

5.5.4 Modelo multicapas

La evolución de las tecnologías para adaptarse al crecimiento informático ha hecho que el desarrollo de los sistemas cambie el enfoque que se tenía de construir aplicaciones. Ahora para la creación de sistemas más flexibles y robustos es necesario utilizar un diseño multicapas.

El diseño multicapas nos permite la creación de un sistema independiente entre las operaciones que realiza, así en caso de requerirlo, se realizarán cambios fácilmente, por ejemplo, en la presentación de la aplicación, inclusive utilizar un manejador de base de datos distinto (Oracle, MS SQL Server, etc.). Generalmente las capas son llamadas capas lógicas y las principales se explican a continuación:

- **Capa de presentación:** Esta capa es tradicionalmente gráfica, pudiendo tomar formas de reportes o pantallas para captura de datos, se encarga de presentar la información y reunir datos. Es usualmente llamada GUI (Graphical User Interface).
- **Capa de negocios:** Esta capa es un puente entre el usuario y los servicios de datos. Estas acciones se llevan a cabo aplicando procedimientos formales y reglas de negocio (políticas que controlan el flujo de las operaciones esenciales del proyecto). Generalmente es la capa que más suele cambiar.
- **Capa de datos:** Esta capa es regularmente la que menos cambios tendrá y es la encargada de interactuar con la base de datos con acciones como almacenar, recuperar y mantener los datos de manera íntegra.

Considerando esta distribución de operaciones por proceso lógico y debido a que necesitamos mayor flexibilidad, la estrategia que seguiremos será un modelo de **seis-capas-lógicas**. Las seis capas se describen a continuación:

- **Capa de presentación:** Provee la interfaz para el usuario (ej. pantallas que interactúan con el cliente). En nuestro caso van a ser formas HTML y Java Server Pages mostradas en un navegador Web. También aquí se puede incluir JavaScript para la validación de los datos en las formas HTML.
- **Capa de contexto del negocio:** Provee la edición de la sintaxis y de contexto para los servicios de negocio (ej. peticiones del cliente). Para nuestra aplicación manejaremos Servlets y JavaServer Pages.
- **Capa de las reglas del negocio:** Implanta las reglas del negocio. En nuestro caso los casos de uso y sus trayectorias serán implantados con clases de control, los cuales serán JavaBeans.
- **Capa de conversión de datos:** Traduce la capa anterior a una lenguaje adecuado para las peticiones requeridas (ej. lenguaje SQL). En nuestro caso son JavaBeans, los cuales utilizan la API JDBC (Java Database Connectivity es una interface de acceso a bases de datos).
- **Capa de acceso a los datos:** Establece una conexión entre la capa anterior y el acceso a la base de datos. En nuestro caso la implantación es vía JDBC.

- **Capa de base de datos:** Soporta físicamente el manejador de base de datos seleccionado. En nuestro caso es PostgreSQL.

La Figura 5.6 es un diagrama de componentes que muestra las unidades de software que se implantarán en la solución de nuestro sistema.

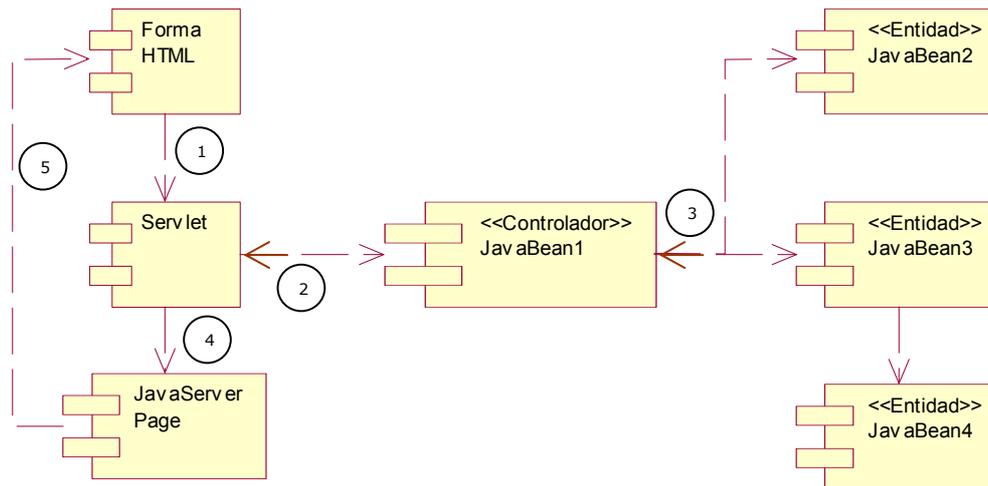


Figura 5.6 Estrategia de componentes con JavaBeans para el SIPIEX

Enseguida describiremos la arquitectura presentada siguiendo la numeración contenida en los círculos:

- **Paso 1:** La forma HTML hace una petición. Esta petición es dirigida al Servlet.
- **Paso 2:** El Servlet hace una instancia de una clase de control (JavaBean). Dependiendo de la acción que pida la forma HTML (Ej. Actualizar un registro), el Servlet invoca un mensaje al JavaBean de controlador.
- **Paso 3:** La misión del JavaBean de controlador es implantar la trayectoria del caso de uso en ejecución. Habrá una clase de control por cada caso de uso. Este controlador hará peticiones a los JavaBean de entidad esperando respuesta de la información solicitada por la forma HTML.
- **Paso 4:** El Servlet reenvía la petición al JavaServer Page apropiado.
- **Paso 5:** El JavaServer Page utiliza la información que el Servlet le proporcionó, formatea una página HTML y manda la información de regreso al navegador que solicitó la petición.

Cada JavaBean de entidad debe ser capaz de recoger información de la base de datos y de volverla a almacenar. Para hacer esto, implantaremos clases **DAO's** (Data Access Objects). Los DAO's cubren los requerimientos de las clases de entidad para acceder a los datos y deben ser capaces de insertar, actualizar y borrarse por ellas mismas.

Para disminuir el tráfico en la red debido a que los objetos de entidad (JavaBeans) suelen ser pesados, utilizaremos objetos imágenes (clases de Valor), que únicamente guardan el estado actual del objeto.

En nuestro modelo las clases DAO utilizarán a las clases de Valor para insertar, actualizar, encontrar o borrar información de la clase de entidad relacionada.

En la Figura 5.7 se muestra un ejemplo de la estrategia seleccionada entre los JavaBeans y las clases DAO:

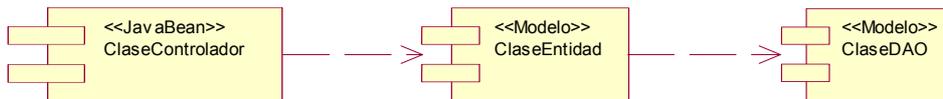


Figura 5.7 Estrategia entre los objetos JavaBean/DAO

5.5.5 Modelo MVC

Tener la capacidad de cambiar la presentación del sistema sin necesidad de mover funciones relevantes en la arquitectura es muy importante para cuestiones de diseño, de hecho la capa de presentación es la que más cambios tendrá dependiendo de las tecnologías que vayan apareciendo (ej. dispositivos PDA tales como celulares y palms).

Como ya lo habíamos mencionado, una manera de hacer más fácil el cambio a otra tecnología es proveer un diseño lo más flexible posible. En nuestro caso utilizaremos el modelo MVC (Model View Controller / Modelo Vista Controlador) para tal fin, ya que nos divide la presentación en tres capas que se describirán a continuación:

- **Modelo:** Esta parte representa los datos de la aplicación, incluyendo las clases de entidad. El objetivo es mantener el marco de trabajo separado de la interfaz del usuario.
- **Vista:** Aquí se hace referencia a la presentación visual que se muestra al usuario. El controlador proporciona la información necesaria a la vista y ésta se encarga de adaptar dicha información a las necesidades del cliente (un reporte, una forma HTML, etc.). Esta sección puede tener conocimiento de las secciones de modelo y del controlador. En nuestro proyecto la vista involucra formas HTML y JavaServer Pages (JSP's).
- **Controlador:** Esta sección está en medio de las otras dos. El controlador es el que actúa cuando se realiza algún cambio en la información del modelo o por alteraciones de la vista, proporcionando respuestas a las peticiones del usuario. El Servlet que sirve como el controlador, se encarga de hacer instancias de los objetos de control y encamina los mensajes a este tipo de objetos.

Un beneficio de la utilización del controlador es que mantiene la interfaz del usuario bastante ligera, ya que las formas HTML o JSP's casi no poseen información de las clases de entidad. Lo anterior hace un modelo altamente cohesivo y poco acoplado entre las distintas capas del modelo.

Como se muestra en la Figura 5.8 tenemos dos tipos de controladores en nuestra arquitectura: un controlador de la interfaz del usuario (ControladorIU) y un controlador

de caso de uso (ControladorCU). El primero es responsable de tratar con la tecnología en uso para la interfaz del usuario (Ej. Web, inalámbrica, etc.), y la segunda es responsable de ejecutar una trayectoria del caso de uso que controla.

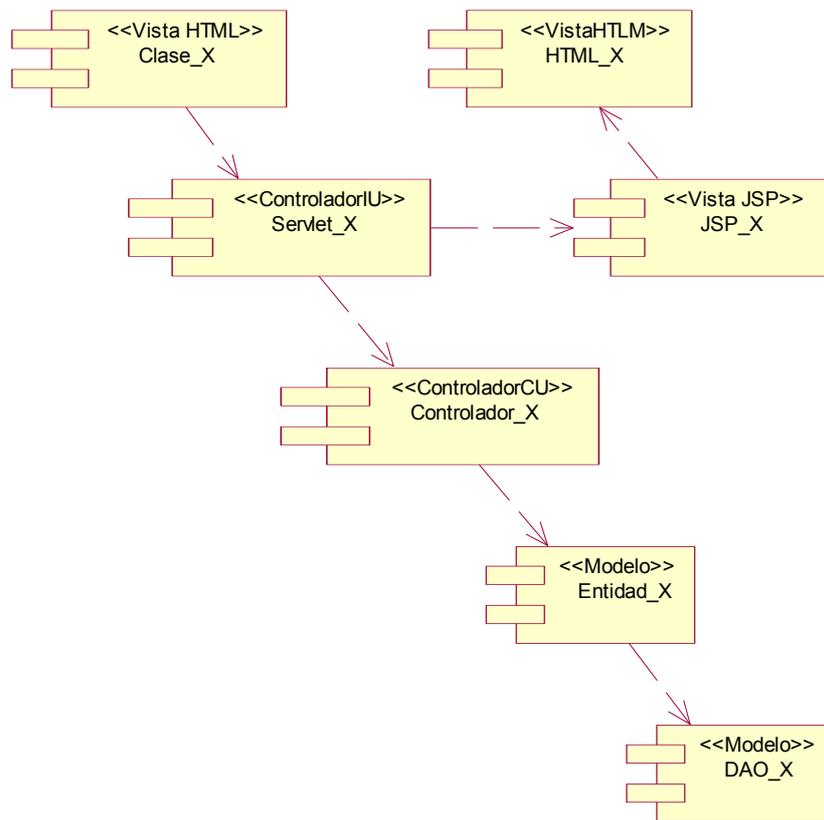


Figura 5.8 Marco de Trabajo MVC

La Figura 5.9 muestra la estrategia de despliegue de la arquitectura propuesta para el SIPIEX, podemos observar todos los componentes están en un mismo servidor.

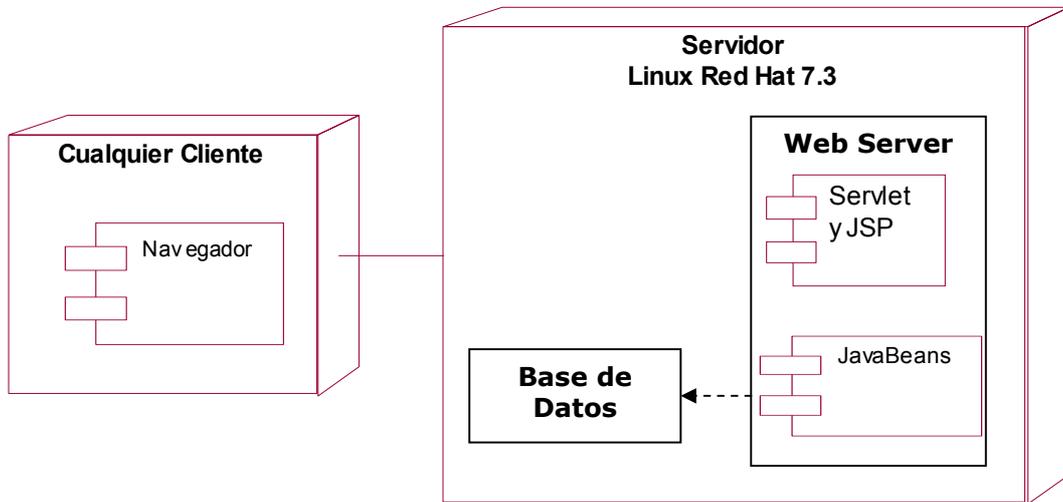


Figura 5.9 Despliegue de la Arquitectura del SIPIEX

Con esto concluimos con el análisis de la arquitectura propuesta para nuestro sistema. En el siguiente capítulo expondremos el análisis, diseño y desarrollo del SIPIEX.

CAPÍTULO VI

ANÁLISIS, DISEÑO Y DESARROLLO DEL SISTEMA

6.1 Introducción

Actualmente para crear software es necesario basarnos en un proceso bien definido de desarrollo de software que nos conduzca de la manera más sencilla a la obtención de los resultados deseados y además nos prevenga de posibles contratiempos o retrasos que pongan en riesgo el desarrollo y/o la entrega del mismo. Por estas razones la metodología para el desarrollo de software que se ajusta más a nuestras necesidades es el Proceso Unificado de Desarrollo (Capítulo V).

De esta manera el análisis, diseño y desarrollo que abordaremos sigue los lineamientos característicos que dicha metodología marca, ya que es iterativo, incremental y basado en el riesgo. Así los requerimientos se van recopilando y depurando con cada iteración e incremento; de esta manera las ventajas y desventajas de continuar con el desarrollo del proyecto se manifiestan y se solucionan desde las primeras fases.

Conforme avancemos en el presente capítulo mencionaremos en que etapa del Proceso Unificado no encontramos para hacer más clara la evolución del proyecto.

6.2 Análisis del sistema

No es de sorprender que en el análisis de requerimientos es donde muchos proyectos cometen su primer error, subestimando el tiempo y esfuerzo que debe dedicarse a esta fase inicial.

Por lo anterior, la recopilación de los requerimientos del SIPIEX fue realizada contemplando el desarrollo iterativo, incremental y basado en el riesgo, como lo marca el Proceso Unificado de Desarrollo.

El análisis de requerimientos nos proporcionará una idea de lo siguiente:

- La finalidad del sistema desde el punto de vista de los actores.
- Los eventos que el sistema debe soportar.
- Los casos de uso del sistema.
- Las trayectorias (principales, alternativas y de excepción) contenidas en cada caso de uso.
- Un plan de trabajo de para las cuatro fases del proyecto (inicial, de elaboración, de construcción y de transición) y un plan de iteraciones.

Viendo el flujo de trabajo y las actividades relacionadas con el Proceso Unificado, la fase inicial enfatiza los siguientes puntos:

- Entender las necesidades de los actores.
- Analizar el problema.
- Definir el sistema.
- Manejar el alcance del sistema.

A continuación se muestra una gráfica del Proceso Unificado con sus diferentes fases y el flujo de trabajo que debemos abordar a lo largo del proyecto (Figura 6.1).

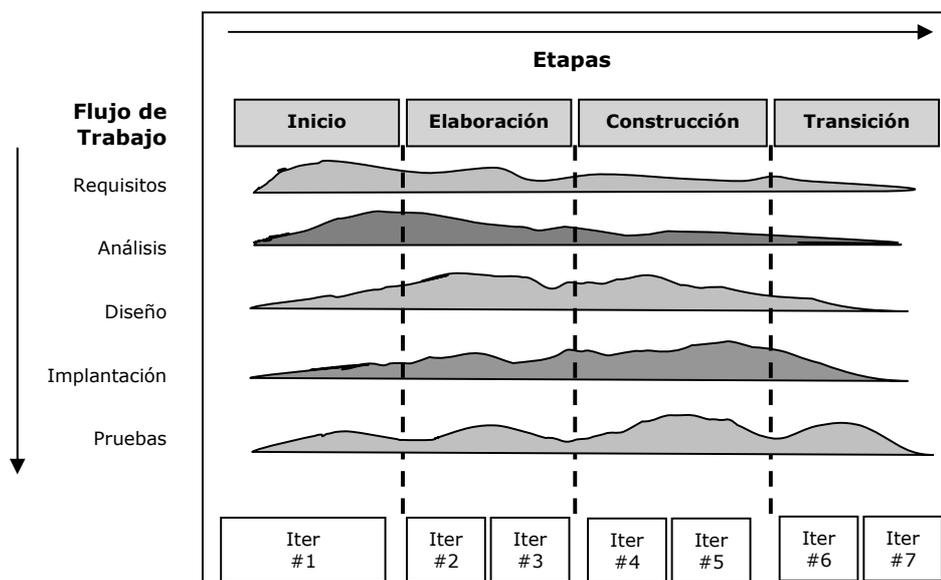


Figura 6.1 Etapas y flujos del Proceso Unificado

Para comenzar con el análisis de requerimientos, es necesario llenar la Tabla 6.1, la cual nos proporcionará una visión general del sistema a desarrollar.

Tópico	Propósito
Propósito del Proyecto	Realizar un sistema de información para el control y administración de los proyectos que generan ingresos extraordinarios en la Facultad de Ingeniería de la UNAM.
Objetivos del Proyecto	Lograr una administración más clara y transparente de los proyectos, disminuir el número de entidades involucradas en los proyectos y entregar información relacionada con los mismos en tiempo y forma a las instancias involucradas.
Características deseadas	Acceso al sistema vía Internet, seguridad de los proyecto dependiendo del usuario (permisos de lectura y escritura), información siempre actualizada (técnica, jurídica y financiera) y disponible para las instancias que así lo requieran.
Factores para el éxito del proyecto	Contar con el Responsable o Líder de Proyecto adecuado para la realización del proyecto, entregar a tiempo cada fase del sistema y dentro del presupuesto establecido.
Restricciones	Entregar en el tiempo establecido y dentro del presupuesto contemplado.
Riesgos	El equipo de trabajo no ha utilizado UML como lenguaje de modelado y tiene poca experiencia en aplicaciones Web.
Roles y responsabilidades	Líder de Proyecto: Ubaldo Acosta Soto. Programador Analista: Marcela G. Gamiño García. Programador Analista: Ubaldo Acosta Soto.
Actores	Ver Tabla 6.2
Tabla de eventos	Ver Tabla 6.3

Tópico	Propósito
Arquitectura preliminar	El sistema tendrá acceso vía Internet, utilizando el lenguaje de programación Java sobre TCP/IP en una red Fast Ethernet usando un diseño multicapas. El manejador de base de datos es PostgreSQL, pero todo el acceso a la base es vía JDBC. Tomcat es el Servidor Web seleccionado y Linux Red Hat el sistema operativo.

Tabla 6.1 Visión general del sistema

6.2.1 Actores

Los actores son aquellas entidades que estimulan el sistema, tales como: personas, sistemas, hardware, software y todo aquello que intervengan en el sistema.

Para encontrar a los actores es necesario hacernos las siguientes preguntas:

- ¿Quién o qué estará interesado en el sistema?
- ¿Quién o qué querrá cambiar la información del sistema?
- ¿Quién o qué solicitará información del sistema?

La Tabla 6.2 muestra los actores encontrados para el SIPIEX y algunas de las acciones que desempeñan.²

Actor	Acción
Director	Consulta información (técnica, jurídica y financiera) de todos los proyectos en promoción y en operación, además consulta información de la memoria tecnológica.
Secretario Administrativo	Consulta información (técnica, jurídica y financiera) de todos los proyectos en promoción y en operación, además de suministrar información financiera.
Jefe del CISI	Consulta información (técnica, jurídica y financiera) de todos los proyectos en promoción y en operación, además consulta información de la memoria tecnológica.
Unidad Jurídica	Consulta y suministra información jurídica de los proyectos en operación (convenios generales, de colaboración y anexos de convenios).
Responsable de Proyecto	Consulta información (técnica, jurídica y financiera) de los proyectos a su cargo.
Líder de proyecto	Proporciona información técnica de los proyectos a su cargo, ya sea en promoción o en operación, llena la apertura de proyecto, proporciona información de la memoria tecnológica y consulta información (técnica, jurídica y financiera) de los proyectos a su cargo.
Responsable del Patrocinador	Consulta información técnica de proyectos en promoción y en operación relacionados con el Patrocinador que representa.

² En algunos casos nos referiremos a "todas las entidades", lo que significa que todos los actores pueden estar involucrados en el evento o caso de uso mencionado.

Actor	Acción
DBA (Administrador de la base de datos)	Consulta información (técnica, jurídica y financiera) de todos los proyectos, ya sean en promoción o en operación, introduce información (técnica, jurídica y financiera), llena aperturas de proyectos y se encarga del mantenimiento de la base de datos.
Público en general (Profesores, alumnos y público interesado)	Consulta la Memoria Tecnológica.

Tabla 6.2 Actores del SIPIEX

6.2.2 Tabla de eventos

Una vez que hemos identificado a los actores, debemos hacer una tabla de eventos que el sistema debe responder desde el punto de vista de los actores. La Tabla 6.3 nos muestra dichos eventos:

Sujeto(s)	Verbo	Objeto
Líder de Proyecto y DBA	Agregar	Proyecto en Promoción
Líder de Proyecto y DBA	Modificar	Proyecto en Promoción
Director, Secretario Administrativo, Jefe del CISI, Responsable de Proyecto, Líder de Proyecto, Responsable del Patrocinador y DBA	Consultar	Proyecto en Promoción
DBA	Eliminar	Proyecto en Promoción
Líder de Proyecto y DBA	Modificar	Proyecto en Operación
Director, Secretario Administrativo, Jefe del CISI, Responsable de Proyecto, Líder de Proyecto, Responsable del Patrocinador y DBA	Consultar	Proyecto en Operación
DBA	Eliminar	Proyecto en Operación
Líder del Proyecto y DBA	Agregar	Responsable o Líder de Proyecto
Líder de Proyecto y DBA	Modificar	Responsable o Líder de Proyecto
Todas las entidades	Consultar	Responsable o Líder de Proyecto
DBA	Eliminar	Responsable o Líder de Proyecto
Líder de Proyecto y DBA	Agregar	Contratante
Líder de Proyecto y DBA	Modificar	Contratante
Líder de Proyecto y DBA	Consultar	Contratante

Sujeto(s)	Verbo	Objeto
DBA	Eliminar	Contratante
Líder de Proyecto y DBA	Agregar	División
Líder de Proyecto y DBA	Modificar	División
Todas las entidades	Consultar	División
DBA	Eliminar	División
Secretario Administrativo y DBA	Agregar	Información Financiera
Secretario Administrativo y DBA	Modificar	Información Financiera
Director, Secretario Administrativo, Jefe del CISI, Responsable de Proyecto, Líder de Proyecto y DBA	Consultar	Información Financiera
DBA	Eliminar	Información Financiera
Líder de Proyecto y DBA	Agregar	Apertura de proyecto
Líder de Proyecto y DBA	Modificar	Apertura de proyecto
Director, Secretario Administrativo, Jefe del CISI, Responsable de Proyecto, Líder de Proyecto y DBA	Consultar	Apertura de proyecto
DBA	Eliminar	Apertura de proyecto
Unidad Jurídica y DBA	Agregar	Información Jurídica
Unidad Jurídica y DBA	Modificar	Información Jurídica
Director, Secretario Administrativo, Jefe del CISI, Unidad Jurídica, Responsable de Proyecto, Líder de Proyecto y DBA	Consultar	Información Jurídica
DBA	Eliminar	Información Jurídica
Todas las entidades	Consultar	Memoria Tecnológica

Tabla 6.3 Tabla de Eventos

6.2.3 Diagrama de casos de uso

Habiendo encontrado los actores y los eventos de interés para el sistema, vamos a asignar ambos a casos de uso. Los casos de uso nos muestran qué debe hacer el sistema, más no cómo.

Cada caso de uso contiene la etiqueta de *procesar*, los que se refiere a las siguientes acciones³ :

³ Existe una excepción en el evento "agregar proyecto en operación". Dicho evento no está permitido, ya que solo se permite agregar un proyecto en promoción, y una vez firmado su convenio se puede mover a la etapa de operación (ver capítulo I).

- Agregar
- Modificar
- Consultar
- Eliminar

La Figura 6.2 muestra el diagrama de casos de uso para el SIPIEX⁴.

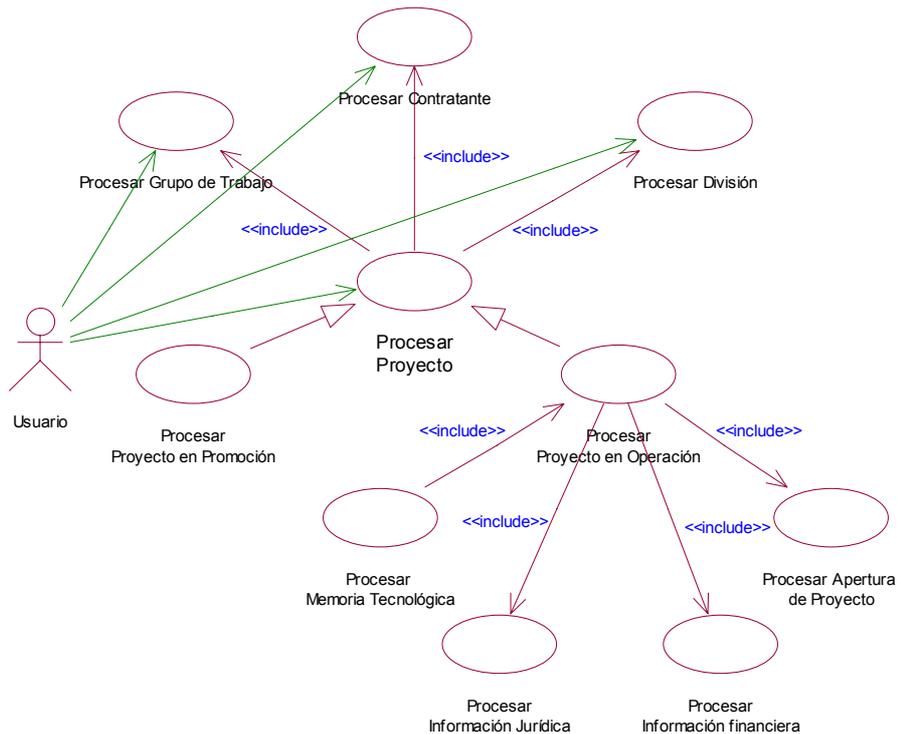


Figura 6.2 Casos de Uso para el SIPIEX

6.2.4 Trayectorias de los casos de uso

Una trayectoria es el camino que debe seguir el sistema para proporcionar el resultado a un evento generado por un actor. Las trayectorias nos sirven para estimar los incrementos y las estrategias de liberación de los productos, es decir, podemos estimar tiempos y costos (Ver Anexo B).

Existen tres tipos de trayectorias: principal, alternativa y de excepción, las cuales describiremos a continuación.

6.2.4.1 Trayectoria principal

La trayectoria principal o *happy path* es el camino que más se presenta en un caso de uso, representando una situación ideal en la que nada erróneo sucede. La Tabla 6.4 detalla las trayectorias principales del SIPIEX.

⁴ La notación se muestra en el capítulo II.

Caso de Uso	Trayectoria principal
Procesar Proyecto en Promoción	El Director, Secretario Administrativo, Jefe del CISI, Responsable de Proyecto, Líder de Proyecto, Responsable del Patrocinador o DBA consulta un Proyecto en Promoción
Procesar Proyecto en Operación	El Director, Secretario Administrativo, Jefe del CISI, Responsable de Proyecto, Líder de Proyecto, Responsable del Patrocinador o DBA consulta un Proyecto en Operación
Procesar Grupo de Trabajo	El Líder de Proyecto o DBA agrega Responsable o Líder de Proyecto
Procesar Contratante	El Líder de Proyecto o DBA agrega un Contratante
Procesar División	El Líder de Proyecto o DBA agrega una División
Procesar Información Financiera	El Secretario Administrativo o DBA consulta Información Financiera
Procesar Apertura de Proyecto	El Líder de Proyecto o DBA agrega una Apertura de Proyecto
Procesar Información Jurídica	La Unidad Jurídica o DBA agrega Información Jurídica
Procesar Memoria Tecnológica	Cualquier entidad consulta información de la Memoria Tecnológica

Tabla 6.4 Trayectorias principales del SIPIEX

6.2.4.2 Trayectoria alternativa

La trayectoria alternativa es todavía considerada como un camino sin errores, pero no es el más comúnmente usado por el actor. La Tabla 6.5 muestra los caminos alternativos para el SIPIEX.

Caso de Uso	Trayectorias alternativas
Procesar Proyecto en Promoción	El Líder de Proyecto o DBA agrega Proyecto en Promoción El Líder de Proyecto o DBA modifica un Proyecto en Promoción El DBA elimina un Proyecto en Promoción
Procesar Proyecto en Operación	El Líder de Proyecto o DBA modifica un Proyecto en Operación El DBA elimina un Proyecto en Operación
Procesar Grupo de Trabajo	El Líder de Proyecto o DBA modifica el Responsable o Líder de Proyecto Cualquier entidad consulta el Responsable o Líder de Proyecto El DBA elimina el Responsable o Líder de Proyecto
Procesar Contratante	El Líder de Proyecto o DBA modifica un Contratante Cualquier entidad consulta un Contratante El DBA elimina un Contratante

Caso de Uso	Trayectorias alternativas
Procesar División	El Líder de Proyecto o DBA modifica una División Cualquier entidad consulta una División El DBA elimina una División
Procesar Información Financiera	El Secretario Administrativo o DBA agrega Información Financiera El Secretario Administrativo o DBA modifica Información Financiera El DBA elimina Información Financiera
Procesar Apertura de Proyecto	El Líder de Proyecto o DBA modifica una Apertura de Proyecto El Director, Secretario Administrativo, Jefe del CISI, Responsable de Proyecto, Líder de Proyecto o DBA consulta una Apertura de Proyecto El DBA elimina una Apertura de Proyecto
Procesar Información Jurídica	La Unidad Jurídica o DBA modifica Información Jurídica El Director, Secretario Administrativo, Jefe del CISI, Unidad Jurídica, Responsable de Proyecto, Líder de Proyecto o DBA consulta Información Jurídica El DBA elimina Información Jurídica

Tabla 6.5 Trayectorias alternativas del SIPIEX

6.2.4.3 Trayectoria de excepción

No siempre las cosas suceden como un actor espera, así que deben existir caminos que puedan predecir posibles errores en la ejecución de una trayectoria. Una excepción es una condición de error, la cual debe ser suficientemente importante para ser tomada en cuenta.

La Tabla 6.6 describe solo algunas de las trayectorias de excepción para el SIPIEX, ya que mencionar todas sería muy extenso, además de que este tipo de trayectorias se van implantando conforme se desarrollan las trayectorias principales y alternativas.

Caso de Uso	Trayectorias de excepción
Procesar Proyecto en Promoción	El Líder de Proyecto o DBA intenta modificar un Proyecto en Promoción y éste no es encontrado
Procesar Proyecto en Operación	El Líder de Proyecto o DBA intenta modificar un Proyecto en Operación y éste no es encontrado
Procesar Grupo de Trabajo	El Líder de Proyecto o DBA intenta modificar Responsable o Líder de Proyecto y éste no es encontrado
Procesar Contratante	El Líder de Proyecto o DBA intenta modificar un Contratante y éste no es encontrado
Procesar División	El Líder de Proyecto o DBA intenta modificar una División y ésta no es encontrada

Caso de Uso	Trayectorias de excepción
Procesar Información Financiera	El Secretario Administrativo o DBA intenta modificar Información Financiera y ésta no es encontrada
Procesar Apertura de Proyecto	El Líder de Proyecto o DBA intenta modificar una Apertura de Proyecto y ésta no es encontrada
Procesar Información Jurídica	La Unidad Jurídica o DBA intenta modificar Información Jurídica y ésta no es encontrada
Procesar Memoria Tecnológica	El Líder de Proyecto o DBA intenta modificar información de la Memoria Tecnológica y ésta no es encontrada

Tabla 6.6 Trayectorias de excepción del SIPIEX

6.2.5 Incrementos

Lo que hemos hecho hasta ahora nos va a servir para determinar los incrementos del SIPIEX, pues recordemos que debemos disminuir el riesgo de fallo en el proyecto conforme éste avanza.

Para continuar con el Proceso Unificado haremos cada uno de los incrementos con sus iteraciones respectivas necesarias a través de las cuatro fases (inicio, elaboración, construcción y transición) basándonos en los casos de uso mencionados.

Los incrementos propuestos para el SIPIEX contienen los siguientes casos de uso:

Incremento 1

- Procesar Proyecto en Promoción
- Procesar Grupo de Trabajo
- Procesar Contratante
- Procesar División

Incremento 2

- Procesar Apertura de Proyecto
- Procesar Proyecto en Operación
- Procesar Información Financiera
- Procesar Información Jurídica
- Procesar Memoria Tecnológica

La Figura 6.3 nos muestra los incrementos del SIPIEX:

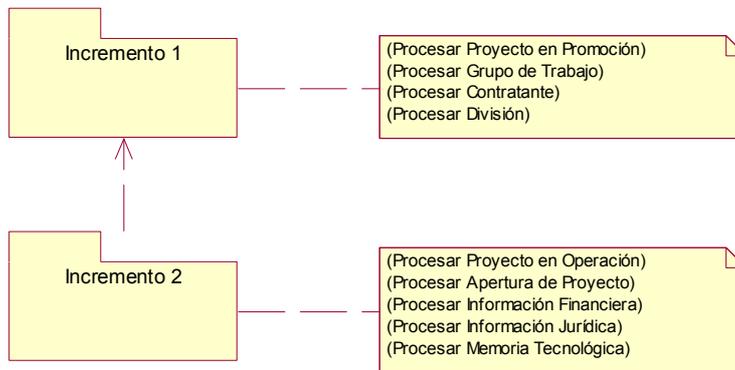


Figura 6. 3 Incrementos del SIPIEX

A continuación vamos a relacionar los incrementos del SIPIEX con las iteraciones del Proceso Unificado:

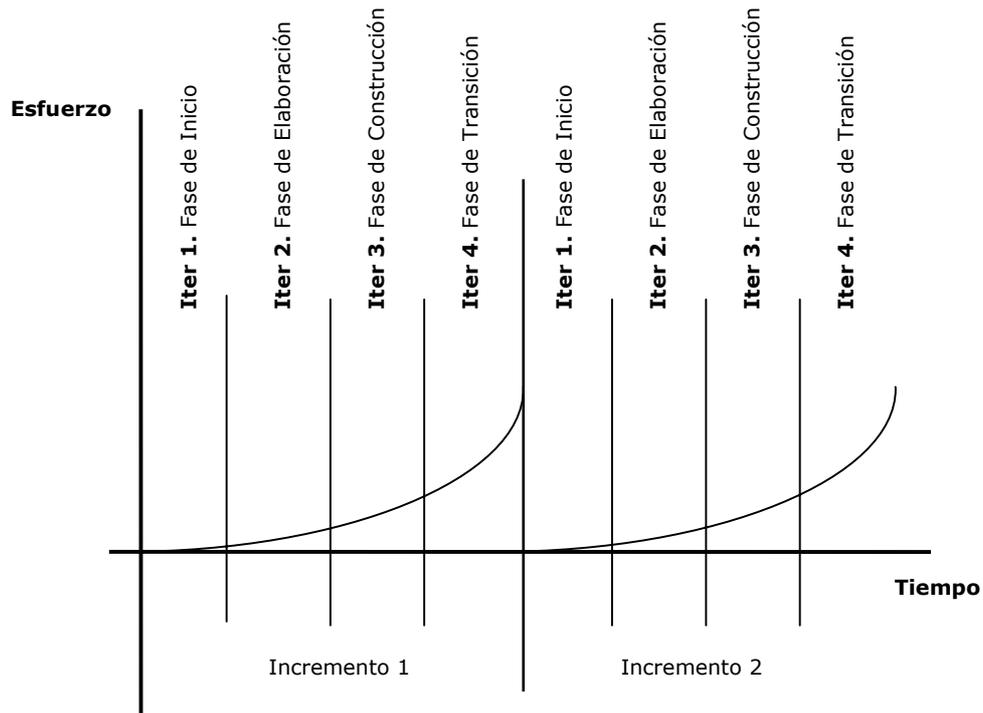


Figura 6.4 Mapeo de los paquetes con las iteraciones del SIPIEX

La Figura 6.4 muestra los incrementos y las iteraciones que estamos considerando en el desarrollo así como la fase que abordemos dependiendo de la iteración que estemos realizando.

6.3 Diseño del sistema

Aquí iniciamos la etapa de elaboración. Antes de comenzar, mencionaremos los flujos de trabajo y las actividades relevantes para esta fase del Proceso Unificado:

- Requisitos: Analizar el problema
- Análisis: Analizar el comportamiento
- Diseño: Diseñar componentes

6.3.1 Descripción de Clases

Una clase es una abstracción de algo que contiene estructura (atributos) y también comportamiento (operaciones). En las siguientes secciones describiremos las distintas clases manejadas en el sistema.

6.3.1.1 Clases de entidad

Las clases de entidad son las comúnmente mencionadas en nuestra documentación, debido a que son el corazón del sistema. Vamos a incluir su descripción detallada de las más importantes y la relación entre ellas, independientemente de que exista otro tipo de clases como son: de Valor, DAO, de control, de interfaz de usuario, entre otras (ver capítulo V).

A continuación expondremos las clases candidatas para el SIPIEX⁵:

- ProyectoPromocion
- ProyectoOperacion
- GrupoDeTrabajo
- Contratante
- Division
- InformacionFinanciera
- Apertura

6.3.1.2 Clases de límite

La manera más sencilla de identificar a las clases de límite es enfocándonos en los actores que interactúan con el sistema, ya que cada uno de ellos requerirá una interfaz. Usualmente existe una clase de límite por cada pantalla o cada reporte, así como por cada interfaz que interactúe con sistemas externos.

En nuestro caso, las clases de límite serán JavaServer Pages (JSP's), las cuales son compiladas como Servlets en tiempo de ejecución, así que podemos decir que nuestras clases de límite son clases de Java.

A continuación mostramos la lista con las posibles clases de límite según el modelo MVC refiriéndonos a la parte de vistas:

Vistas JSP

- proyectoPromocion.jsp

⁵ La documentación presentada ya incluye los dos incrementos, para evitar hacerlo demasiado extenso, aunque se podría presentar incremento por incremento.

- proyectoOperacion.jsp
- Contratante.jsp
- Division.jsp
- ResponsableDeProyecto.jsp
- LiderDeProyecto.jsp
- Apertura.jsp
- InformacionFinanciera.jsp
- InformacionJuridica.jsp
- MemoriaTecnologica.jsp

6.3.1.3 Clases de control

Definiremos una *clase de interfaz de usuario* por cada *clase de control de caso de uso* (ver Figura 5.8, capítulo V).

Crearemos una *clase de control de caso de uso* por cada caso de uso que tenga entradas independientes⁶. La clase de control debe tener por lo menos tantos métodos como trayectorias tiene el caso de uso que está implantando (sin incluir las trayectorias de excepción).

A continuación mostraremos las clases de control según el modelo MVC:

Clases de control de interfaz de usuario

- ProyectoServlet
- GrupoDeTrabajoServlet
- DivisionServlet
- ContratanteServlet

Clases de control de caso de uso

- ProyectoPromocionCU
- ProyectoOperacionCU
- GrupoDeTrabajoCU
- DivisiónCU
- ContratanteCU

6.3.1.4 Relaciones entre clases

Las relaciones entre clases son necesarias por que los sistemas orientados a objetos están basados en la colaboración entre objetos para llevar a cabo una tarea específica.⁷ La Figura 6.5 muestra el diagrama de clases de entidad para el SIPIEX.

6.3.1.5 Métodos y atributos de las clases

Una vez identificadas las clases candidatas podemos asignarles atributos y métodos dependiendo la función que desarrollen.

⁶ Un ejemplo de un caso de uso dependiente es *Procesar información financiera*, el cual depende de un proyecto en operación, por lo que no se puede agregar información financiera sin antes haber entrado a un proyecto, por ello no tendrá clase de control.

⁷ Para más detalles de notación de relaciones en UML consultar capítulo 2, sección 2.5.1.3

Existen relaciones llamadas transitivas que son necesarias para realizar un puente entre clases donde sus relaciones son muchos a muchos, estas clases nos permiten simplificar la relación. En nuestro caso las clases transitivas son:

- ProyectoContratante
- Rol
- Porcentaje

En la Figura 6.5 se muestra el diagrama de clases para el SIPIEX.

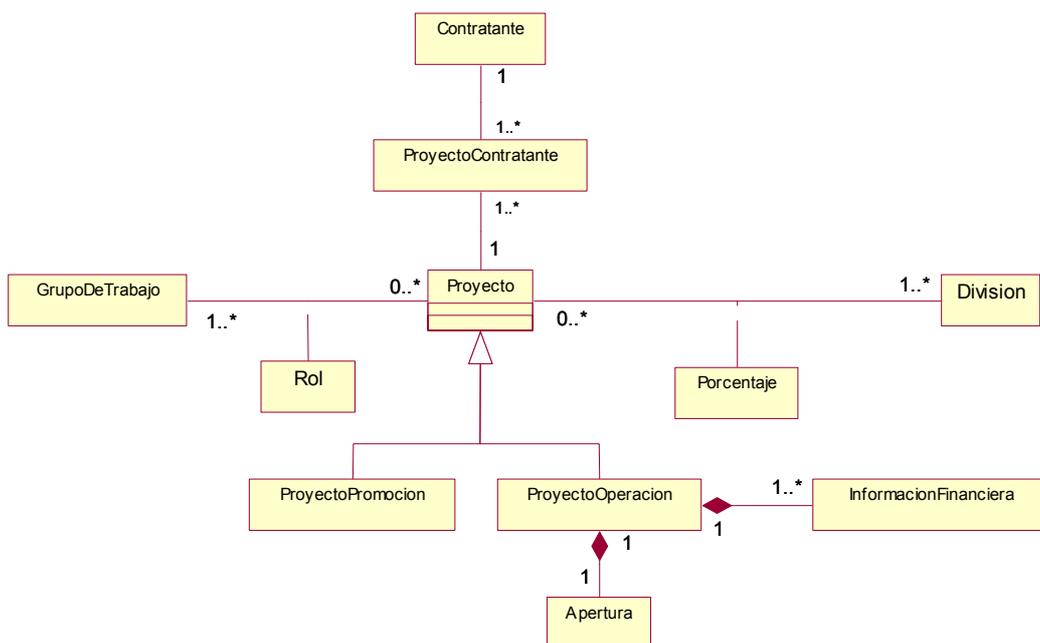


Figura 6.5 Diagrama de clases de entidad para el SIPIEX

Los atributos y métodos básicos para cada clase los describiremos en la Tabla 6.7.

Clase	Atributos	Métodos	Descripción
Proyecto	proyectoId clave importe nombre objetivo rolValor proyContValor porcentajeValor	Proyecto () get/setProyectoId () get/setClave () get/setImporte () get/setNombre () get/setObjetivo () get/setRolValor () get/setProyContValor () get/setPorcentajeValor () get/setProyectoValor () getRoles () getContratantes () getPorcentajes () encontrarPorClave () actualizarProyecto () insertarProyecto () eliminarProyecto () agregarRol () agregarContratante () agregarPorcentaje () encontrarTodos () encontrarPorId ()	Clase abstracta que contiene la mayoría de los métodos que un proyecto necesita y de ésta se partirá para crear las clases ProyectoPromocion y ProyectoOperacion.
ProyectoPromocion	duracion estatusConv	get/setDuracion get/setEstatusConv	Clase para manejar proyectos en promoción
ProyectoOperacion	avanceFisico ligaContrato fechaIniConv fechaFinConv noConvenio	get/setAvanceFisico () get/setLigaContrato () get/setFechaIniConv() get/setFechaFinConv() get/setNoConvenio get/VigenciaConvenio() get/Saldo()	Clase para manejar proyectos en operación
GrupoDeTrabajo	GrupoDeTrabajoId nombre ubicacion telefono email rolValor	get/setGrupoDeTrabajoId () get/setNombre () get/setUbicacion () get/setTelefono () get/setEmail () get/setRolValor ()	Clase para manejar a los responsables o líderes de un proyecto. Inclusive puede manejar personas que pertenezcan al grupo de trabajo del proyecto
Rol	rolId proyectoValor Responsable o Líder de ProyectoValor nombreRol	get/setRolId () get/setProyectoValor () get/setResponsable o Líder de ProyectoValor () get/setNombreRol ()	Clase para indicar si la clase GrupoDeTrabajo se refiere a un responsable, un líder o parte del grupo de trabajo

Clase	Atributos	Métodos	Descripción
Contratante	contratanteId nombre abreviatura encargado ubicacion telefono celular email proyContValor	Contratante() get/setContratanteId () get/setNombre () get/setAbreviatura () get/setEncargado () get/setUbicacion () get/setTelefono () get/setCelular () get/setEmail () get/setProyContValor () getProyectos () encontrarPorNombre () actualizarContratante () insertarContratante () eliminarContratante () agregarProyecto () encontrarTodos () encontrarPorId ()	Clase para manejar los contratantes de un proyecto
ProyectoContratante	proyContId proyectoValue contratanteValue	get/setProyContId get/setProyectoValue get/setContratanteValue	Clase para manejar la información entre los contratantes y los proyectos
Division	divisionId nombre abreviatura porcentajeValor	get/setDivisionId () get/setNombre{ () get/setAbreviatura () get/setPorcentajeValor () actualizarDivision () insertarDivision () eliminarDivision () encontrarTodos () encontrarPorcentajes () encontrarPorNombre () encontrarPorId ()	Clase para manejar la información de una división
Porcentaje	porcentajeId proyectoValor divisionValor porcentaje	get/setPorcentajeId () get/setProyectoValor () get/setDivisionValor () get/setPorcentaje ()	Clase para indicar que porcentaje respecto al monto de un proyecto le corresponde a cada división
InformacionFinanciera	infFinancieraId proyectoValor tipoTramite concepto fecha importe tipo	get/setInfFinancieraId () get/setProyectoValor () get/setTipoTramite () get/setConcepto () get/setFecha () get/setImporte () get/setTipo () actualizarInfFinanciera () insertarInfFinanciera () elimnarInfFinanciera () encontrarPorId ()	Clase para almacenar la información financiera de un proyecto en operación

Clase	Atributos	Métodos	Descripción
Apertura	aperturaId proyectoValor benfAcademicos benfEconomicos futurosProyectos	get/setAperturaId () get/setProyectoValor () get/setBenfAcademicos () get/setBenfEconomicos () get/setFuturosProyectos () actualizarApertura () insertarApertura () eliminarApertura () encontrarPorId ()	Clase para almacenar los datos de la apertura de proyecto para un proyecto de operación

Tabla 6. 7 Atributos y Métodos de las clases de entidad del SIPIEX

6.3.2 Diagramas de Secuencia

En este apartado vamos a agregar aspectos dinámicos al sistema, lo que mostrará la colaboración entre los objetos de las clases encontradas para el SIPIEX, con el único propósito de satisfacer las trayectorias (principales y alternativas) anteriormente descritas.

De las Figuras 6.6 a la 6.44 mostramos los diagramas de secuencia para cada trayectoria, principal y alterna, de cada caso de uso:

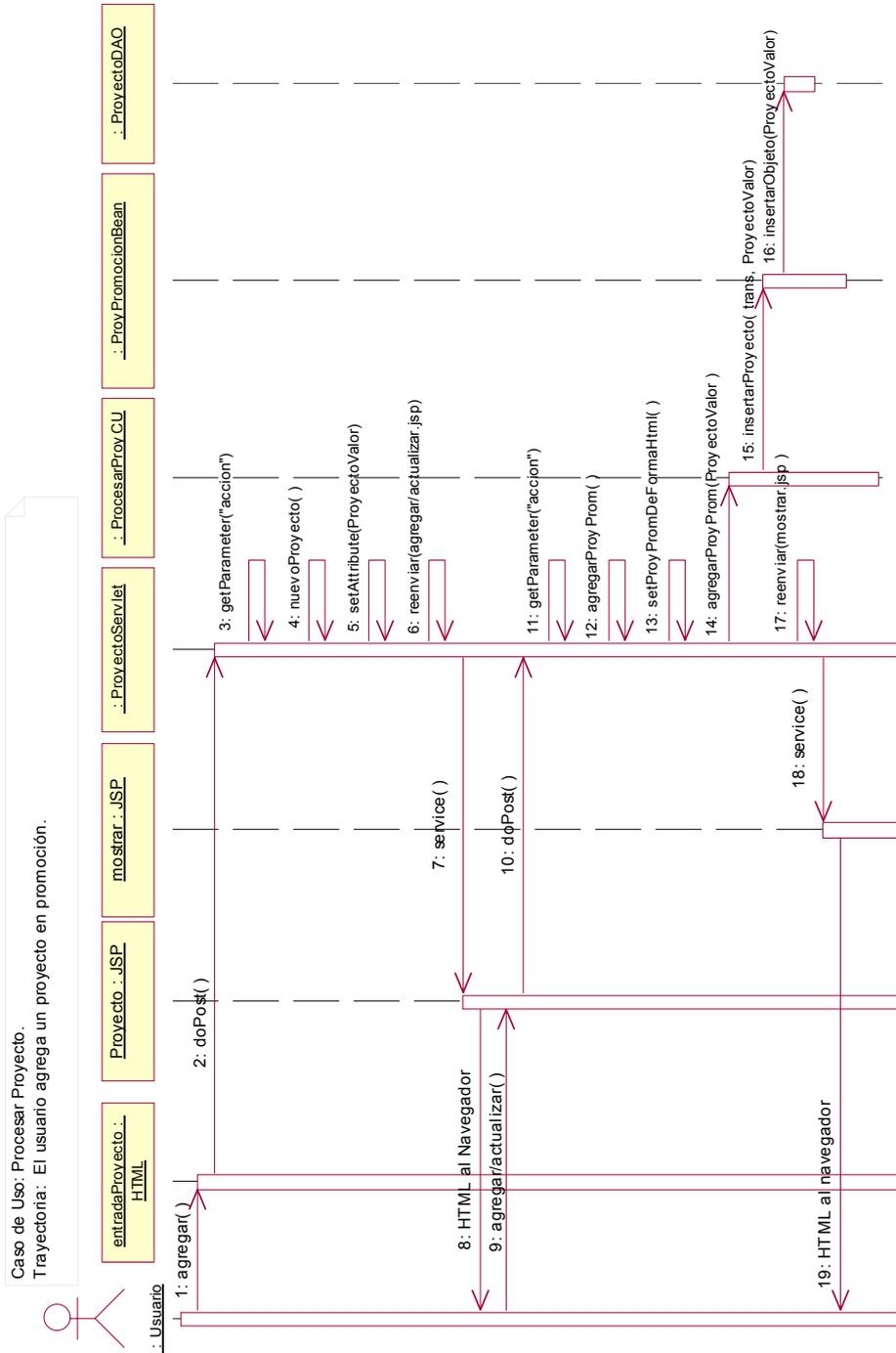


Figura 6. 6 El usuario agrega un proyecto en promoción

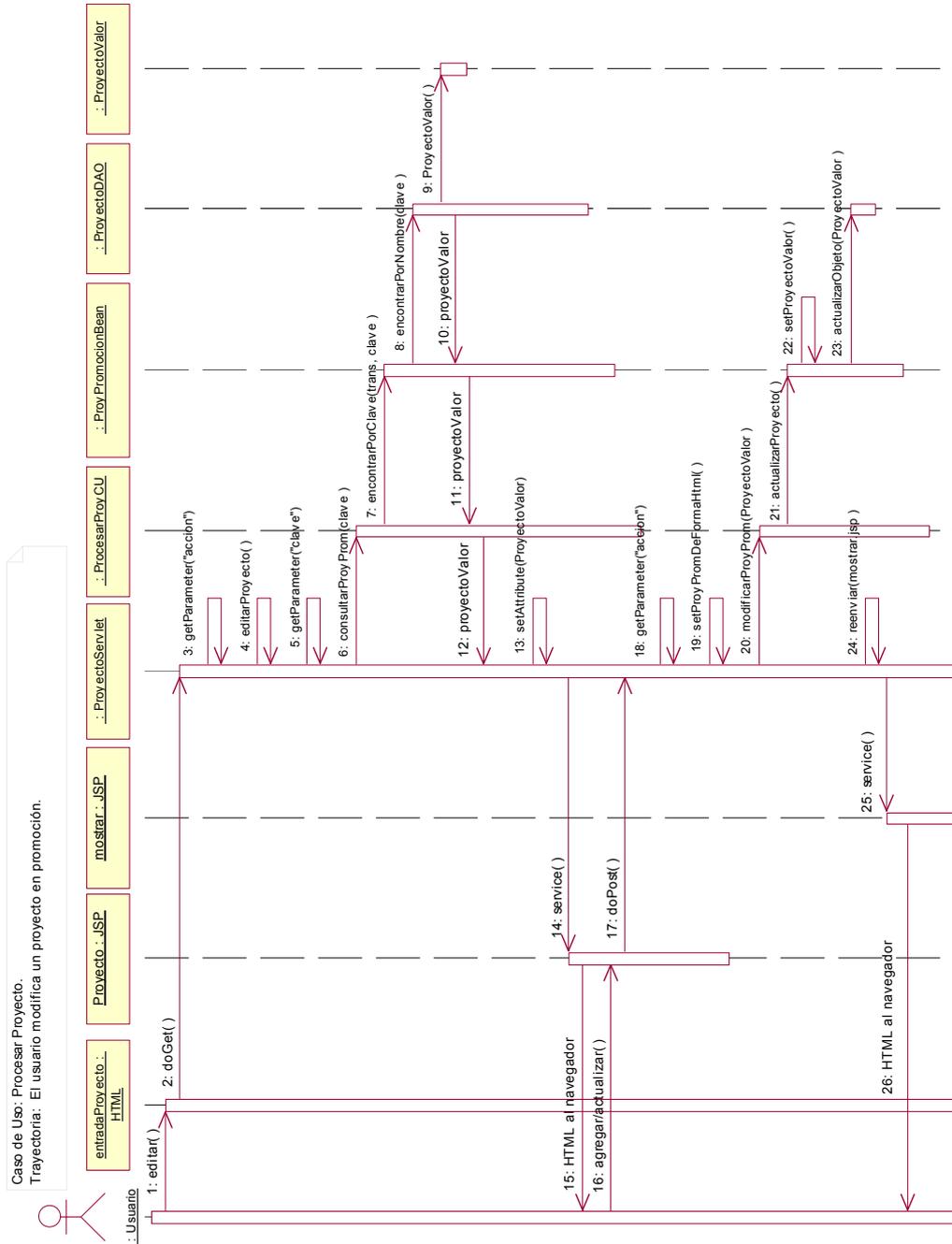


Figura 6.7 El usuario modifica un proyecto en promoción

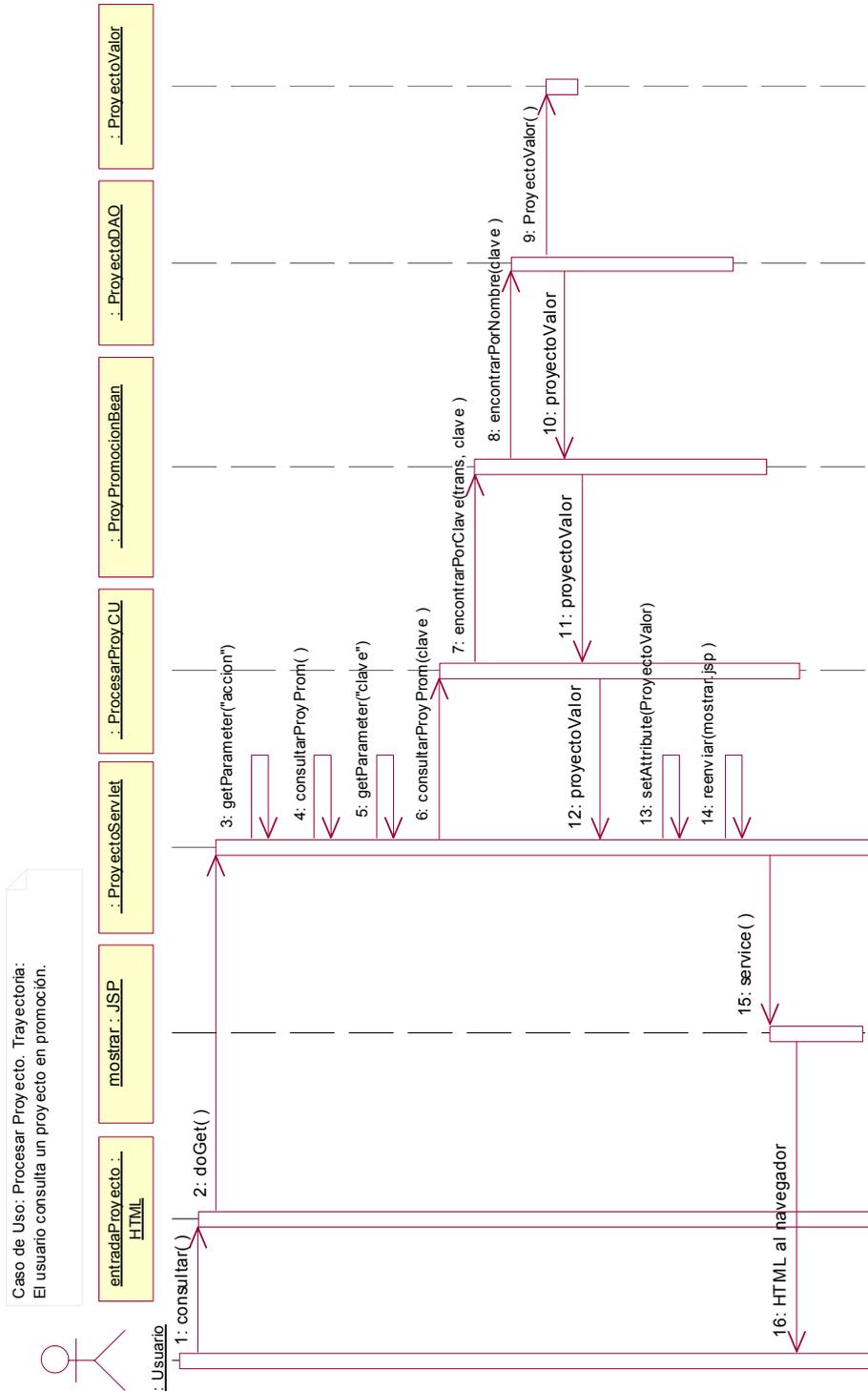


Figura 6.8 El usuario consulta un proyecto en promoción

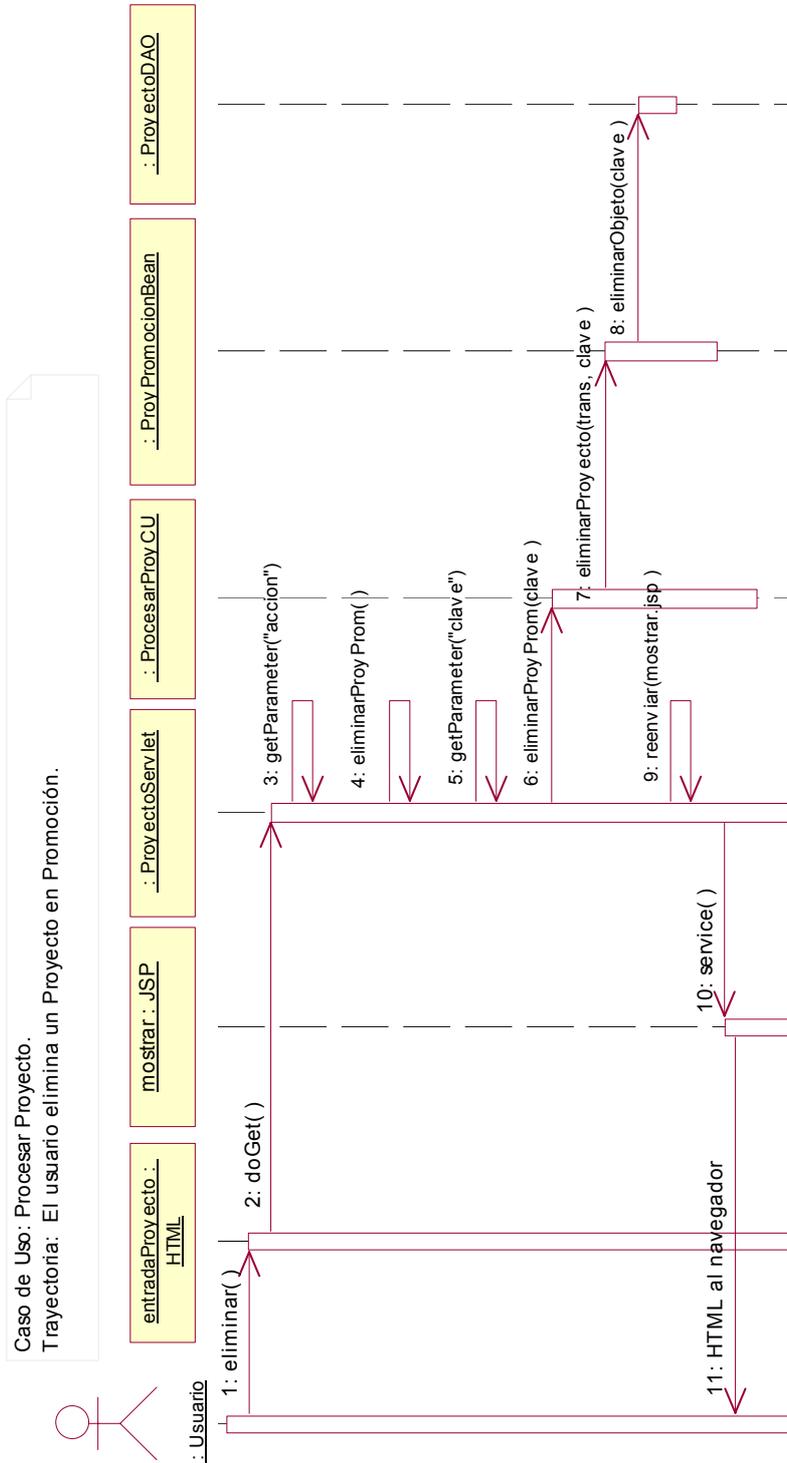


Figura 6.9 El usuario elimina un proyecto en promoción

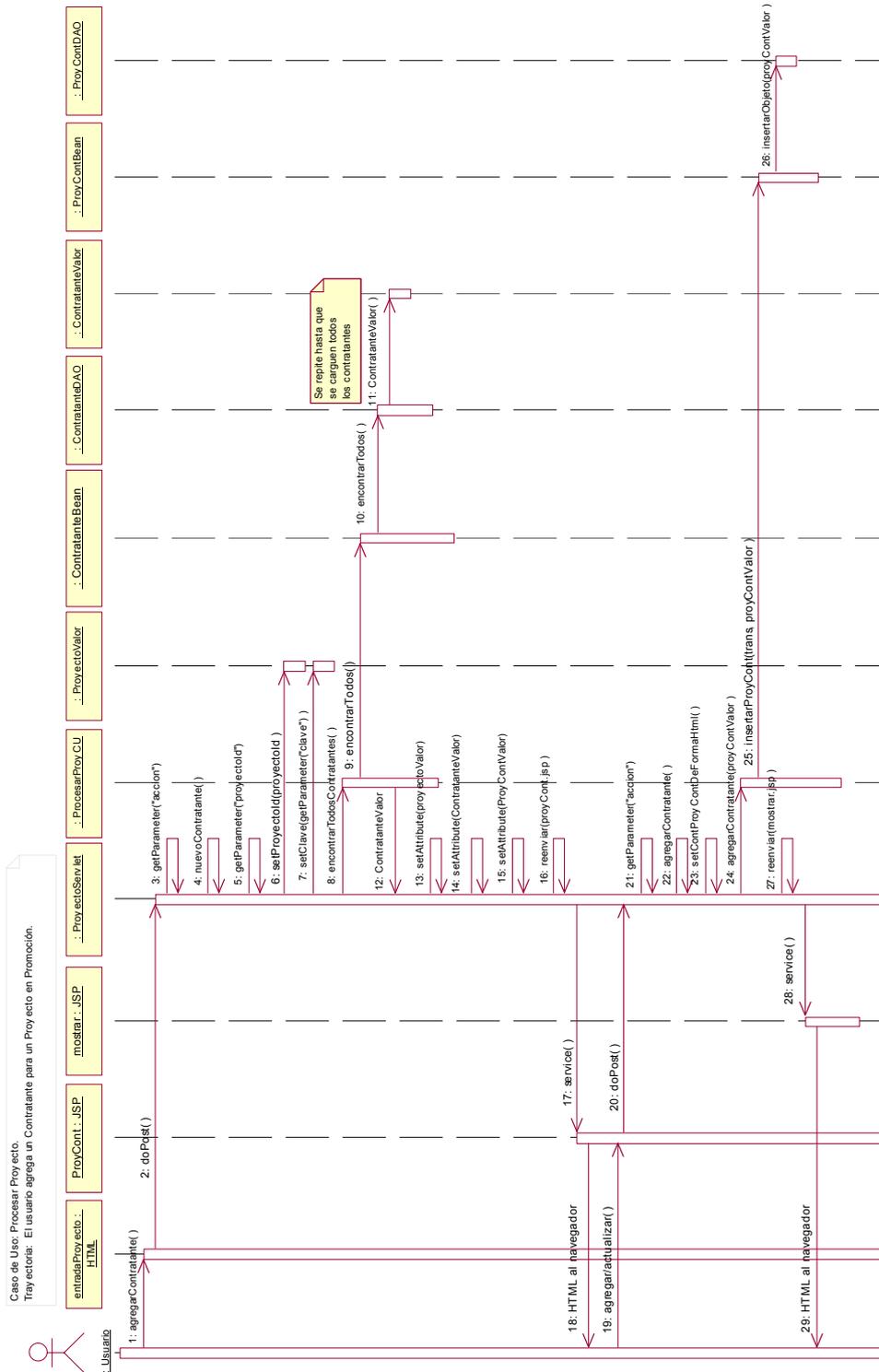


Figura 6.10 El usuario agrega un Contratante para un proyecto en promoción/operación

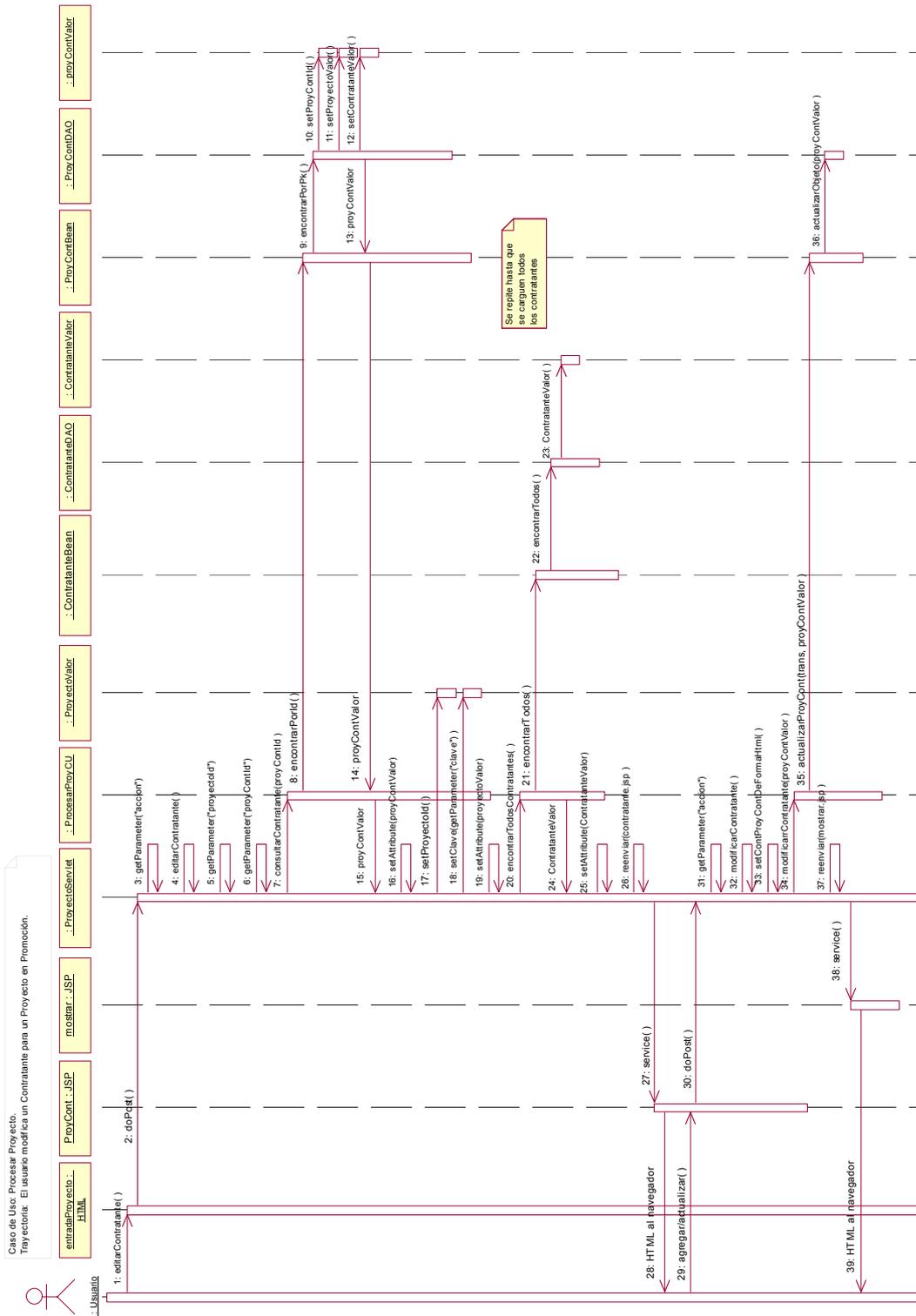


Figura 6.11 El usuario modifica un Contratante para un proyecto en promoción/operación

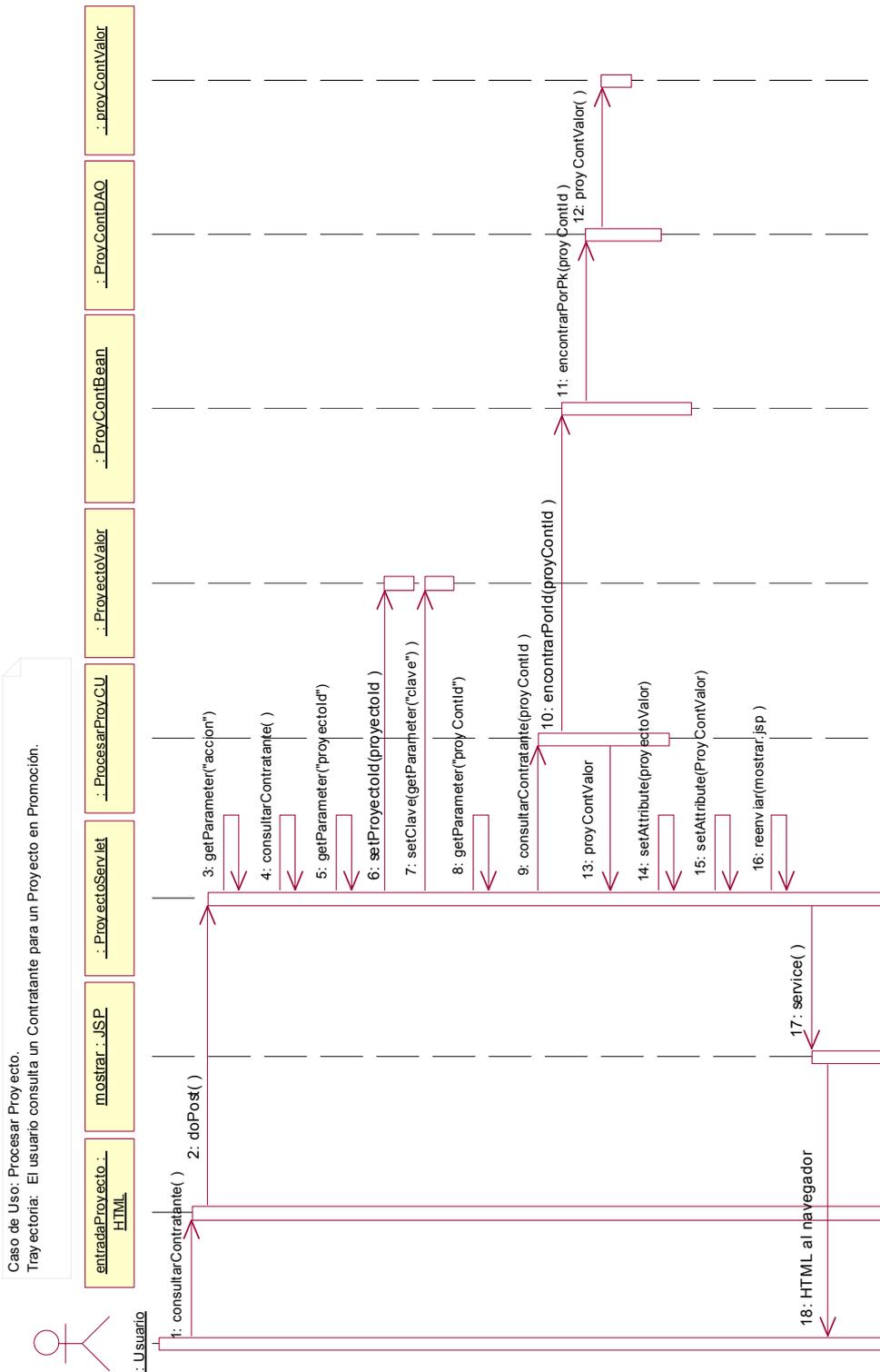


Figura 6.12 El usuario consulta un Contratante para un proyecto en promoción/operación

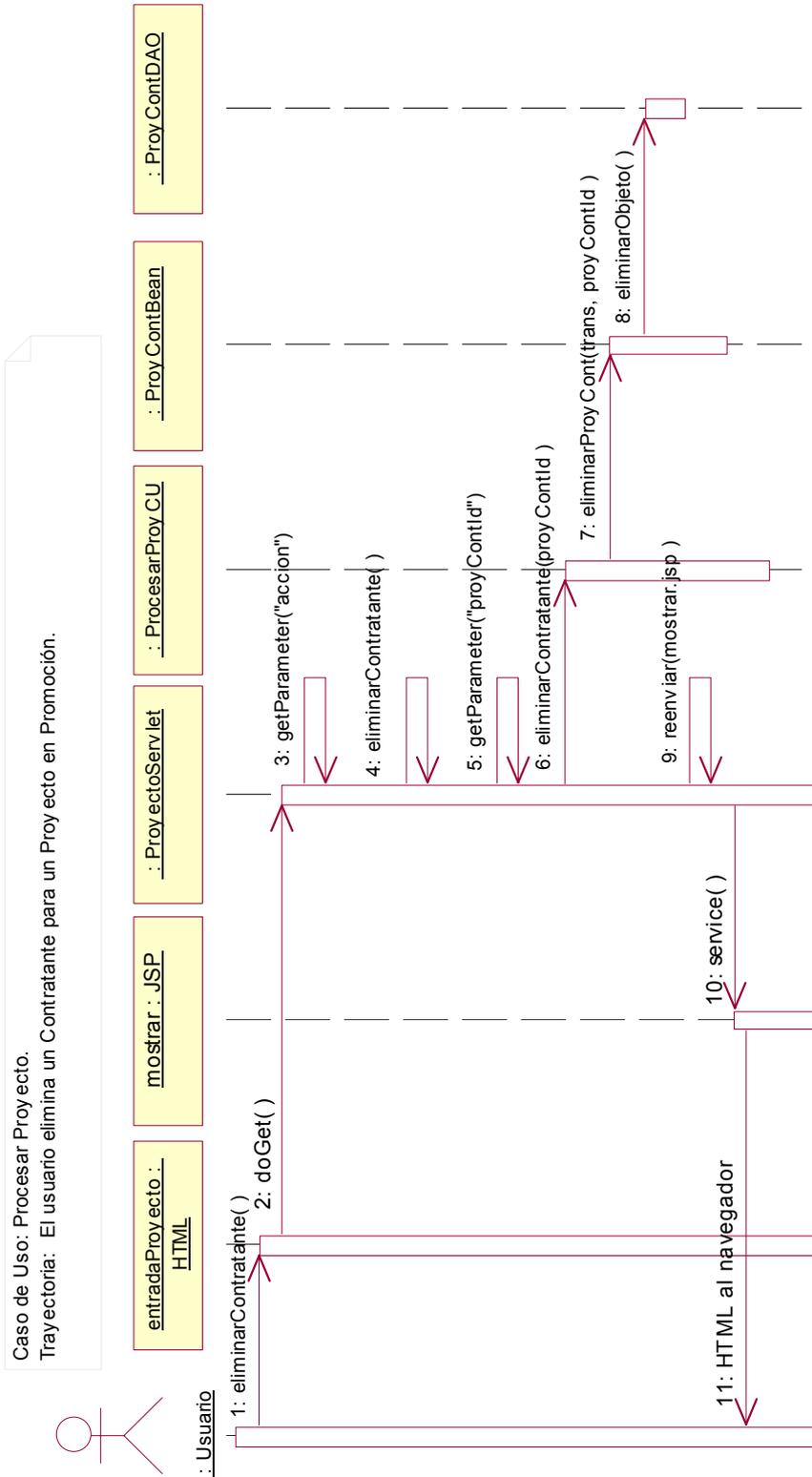


Figura 6.13 El usuario elimina un Contratante para un proyecto en promoción/operación

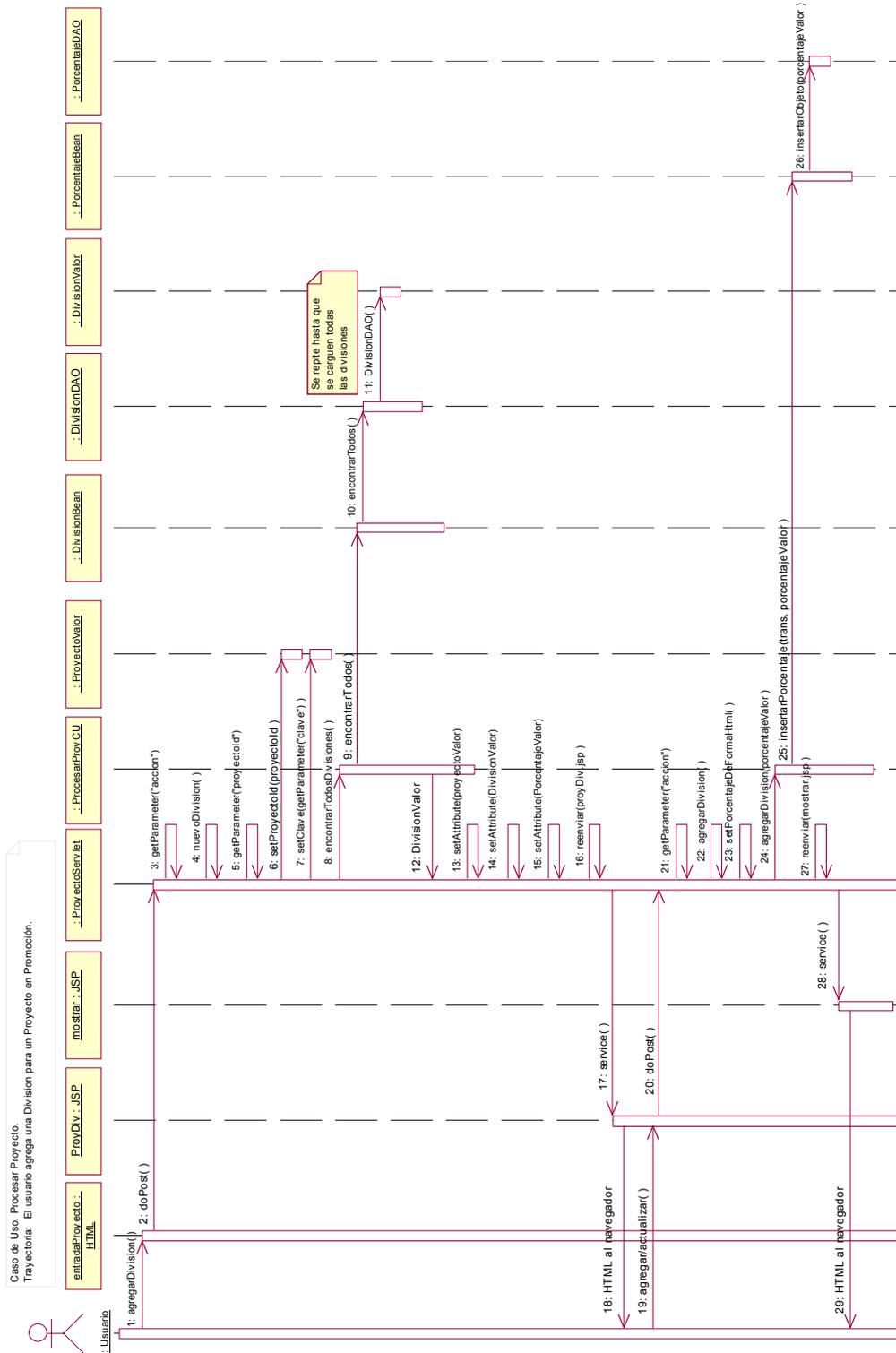


Figura 6.14 El usuario agrega una División para un proyecto en promoción/operación

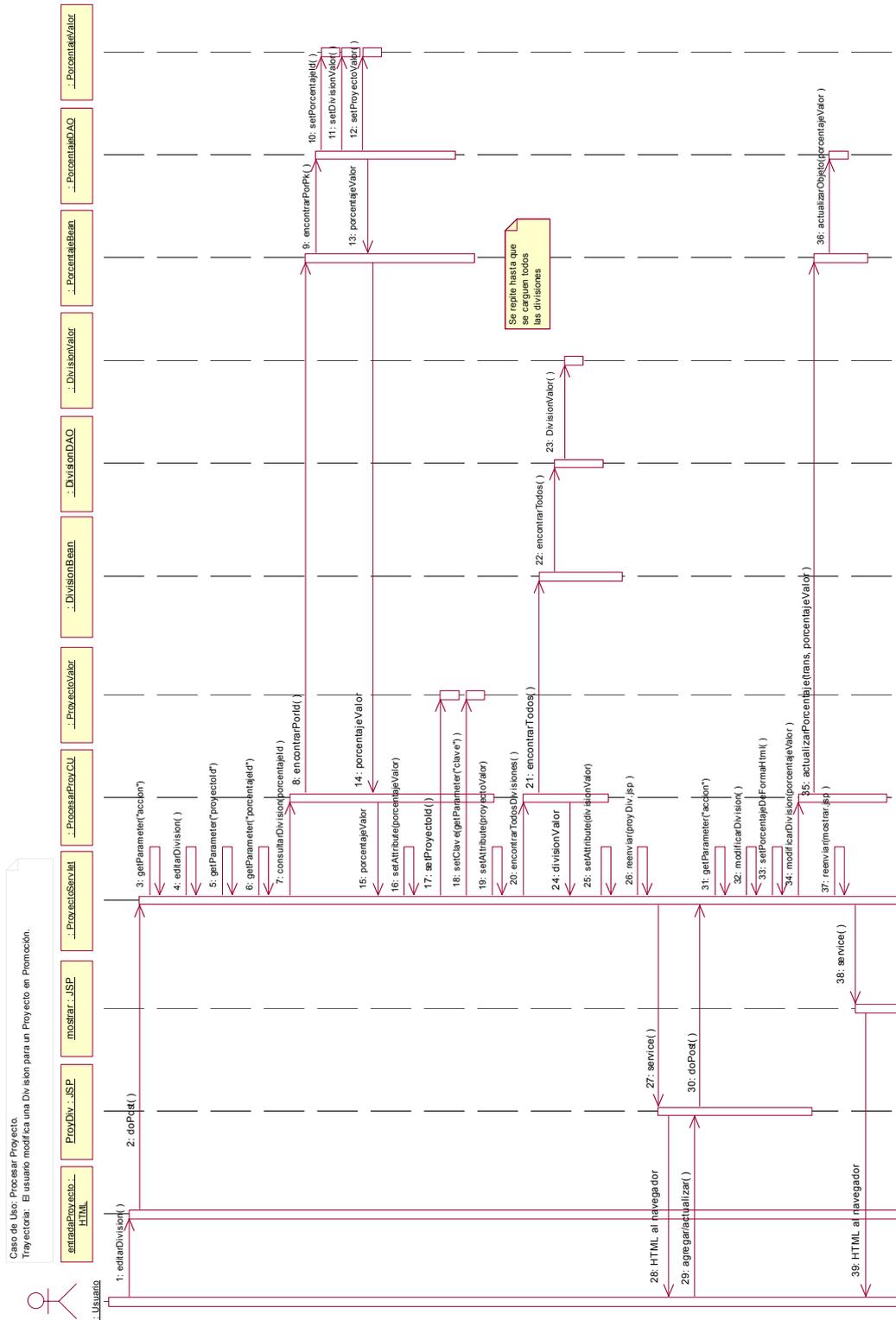


Figura 6.15 El usuario modifica una División para un proyecto en promoción/operación

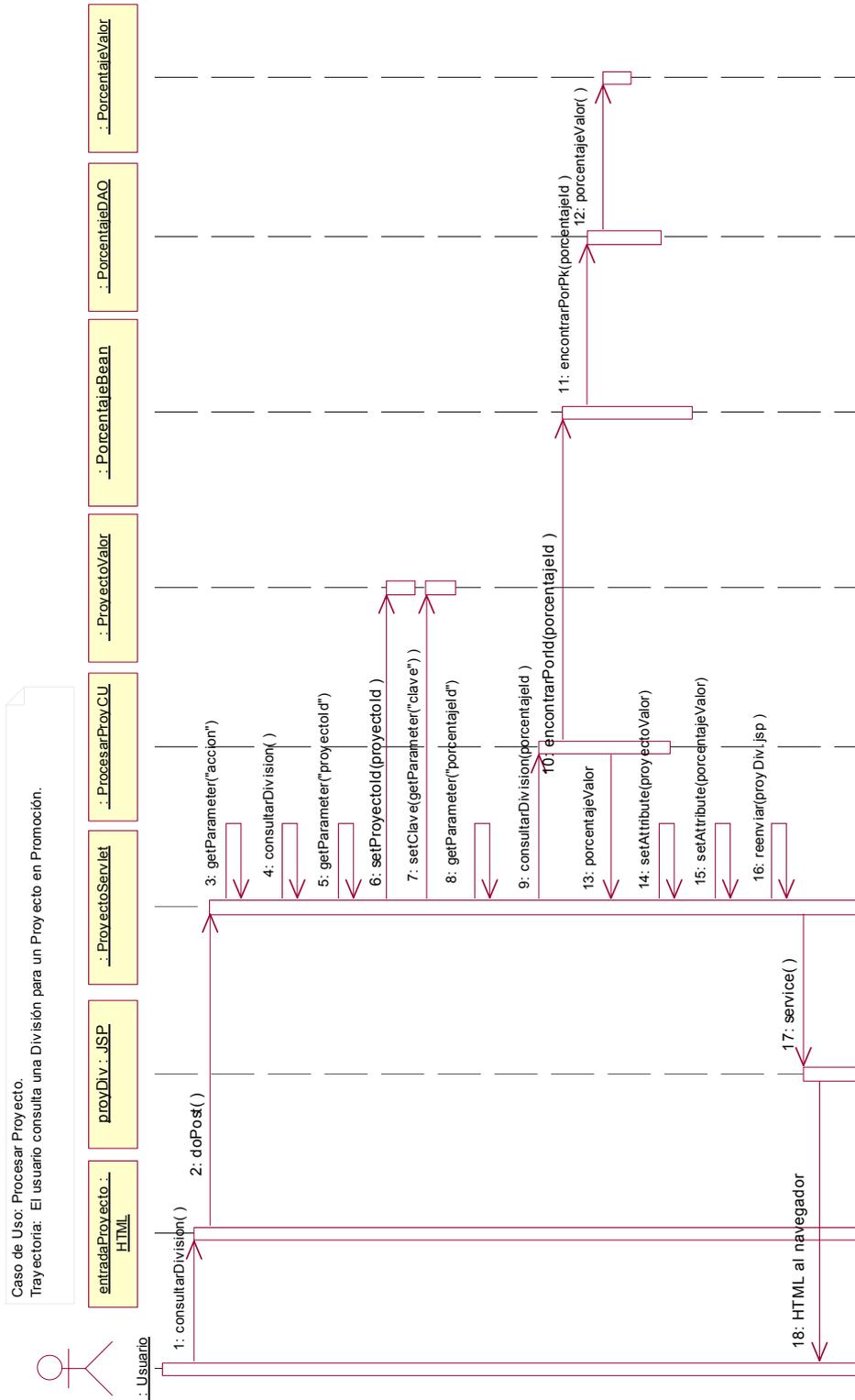


Figura 6.16 El usuario consulta una División para un proyecto en promoción/operación

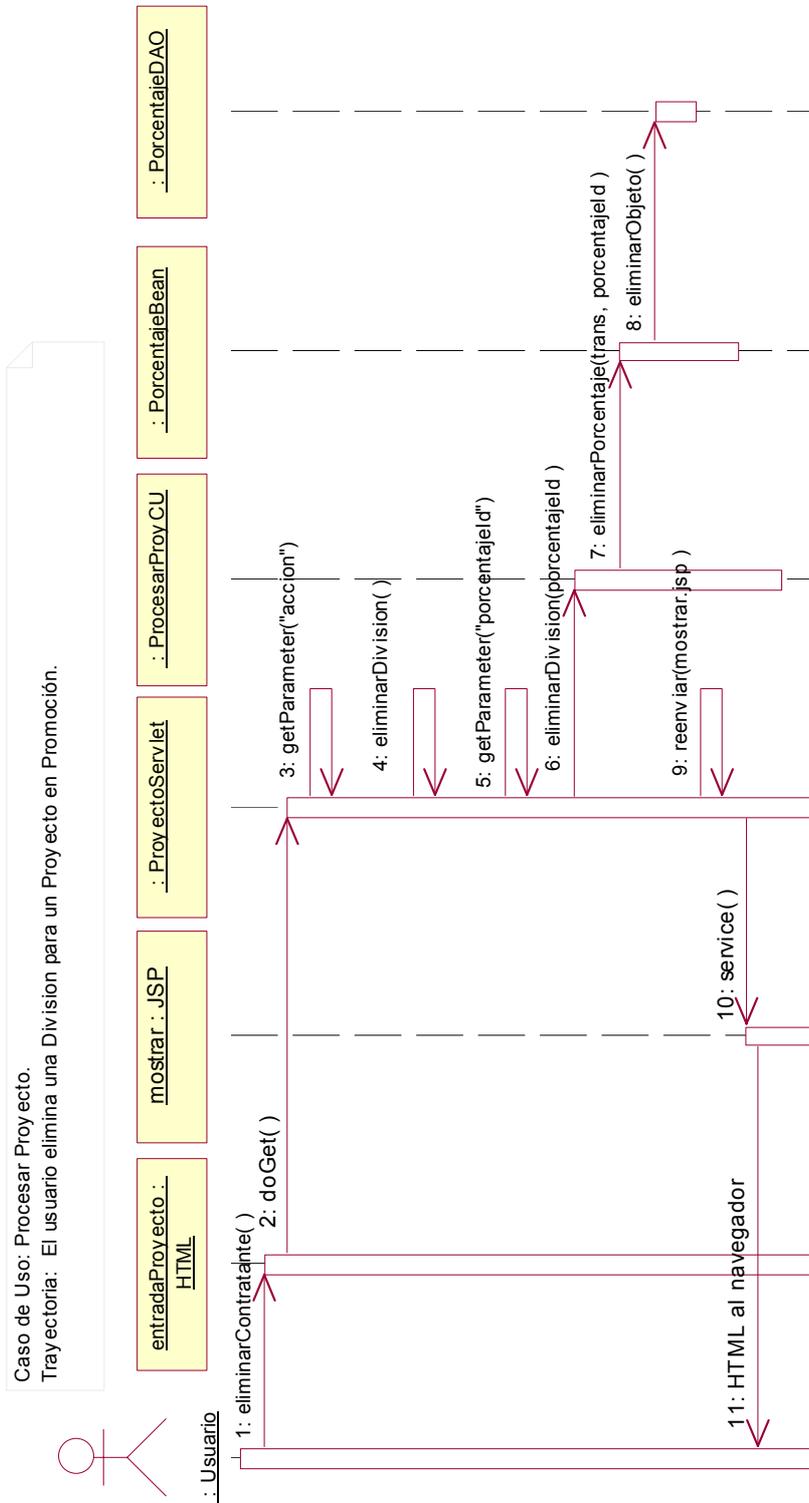


Figura 6.17 El usuario elimina una División para un proyecto en promoción/operación

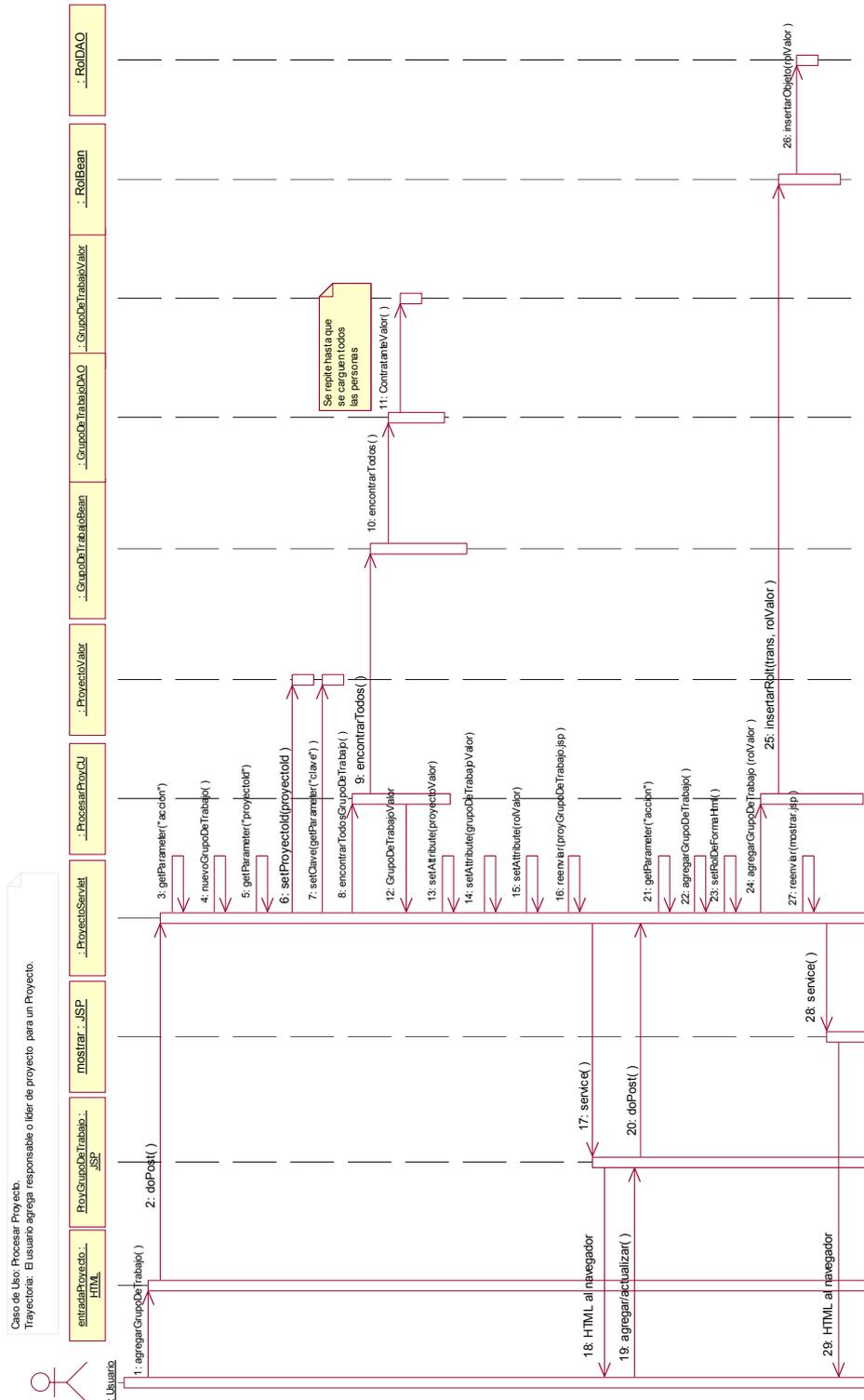


Figura 6.18 El usuario agrega Responsable o Líder de Proyecto para un proyecto

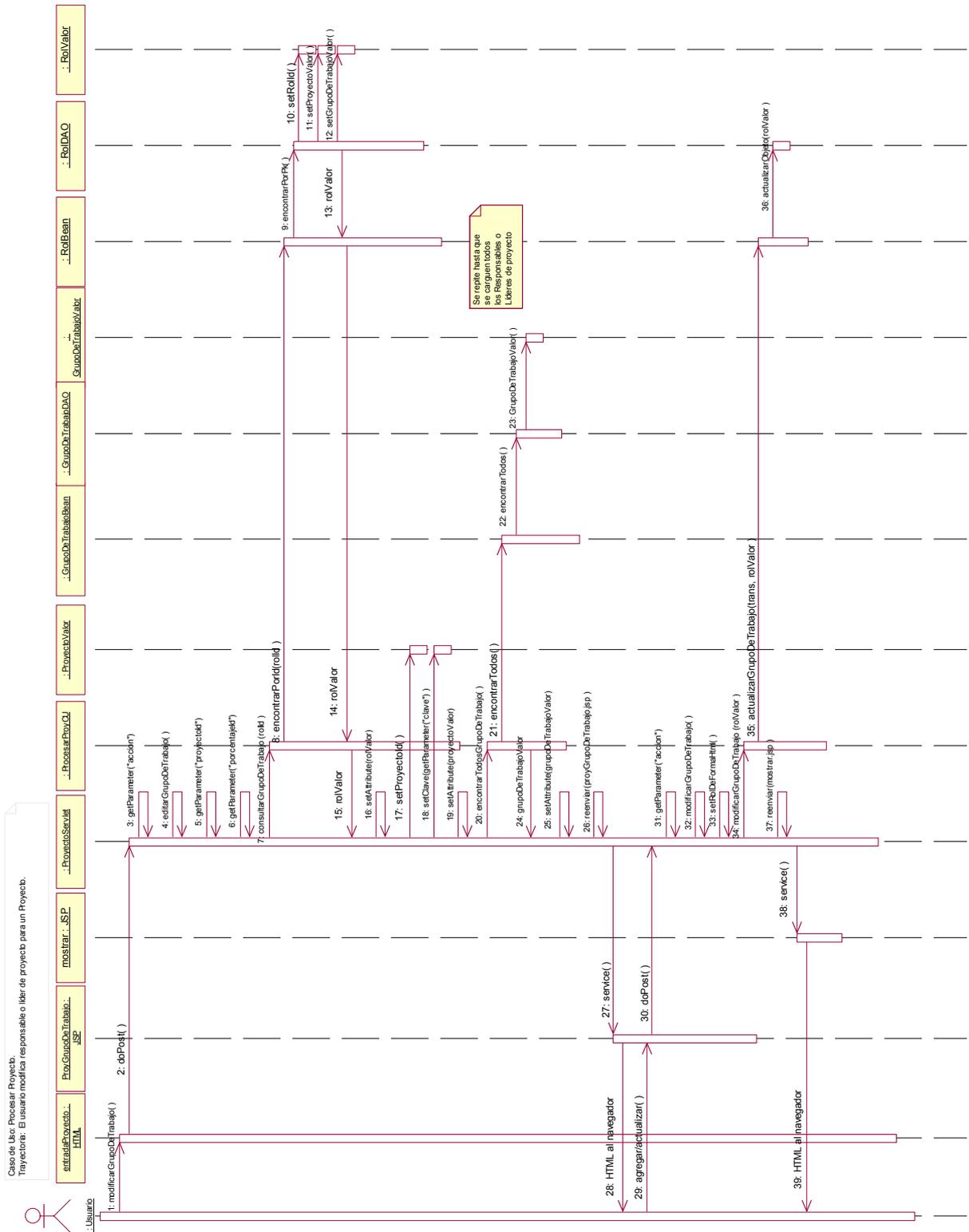


Figura 6.19 El usuario modifica Responsable o Líder de Proyecto para un proyecto

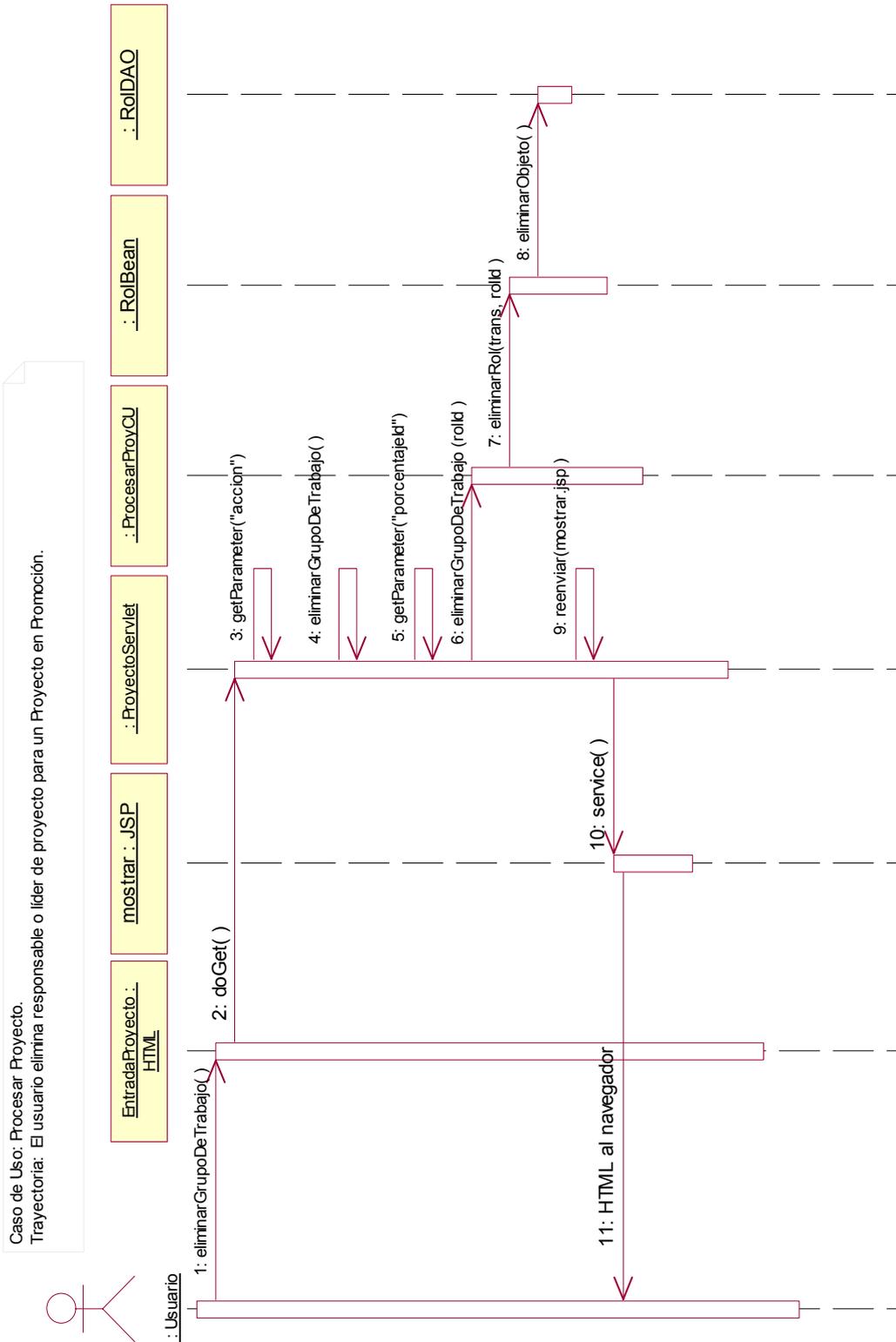


Figura 6.20 El usuario elimina Responsable o Líder de Proyecto para un proyecto

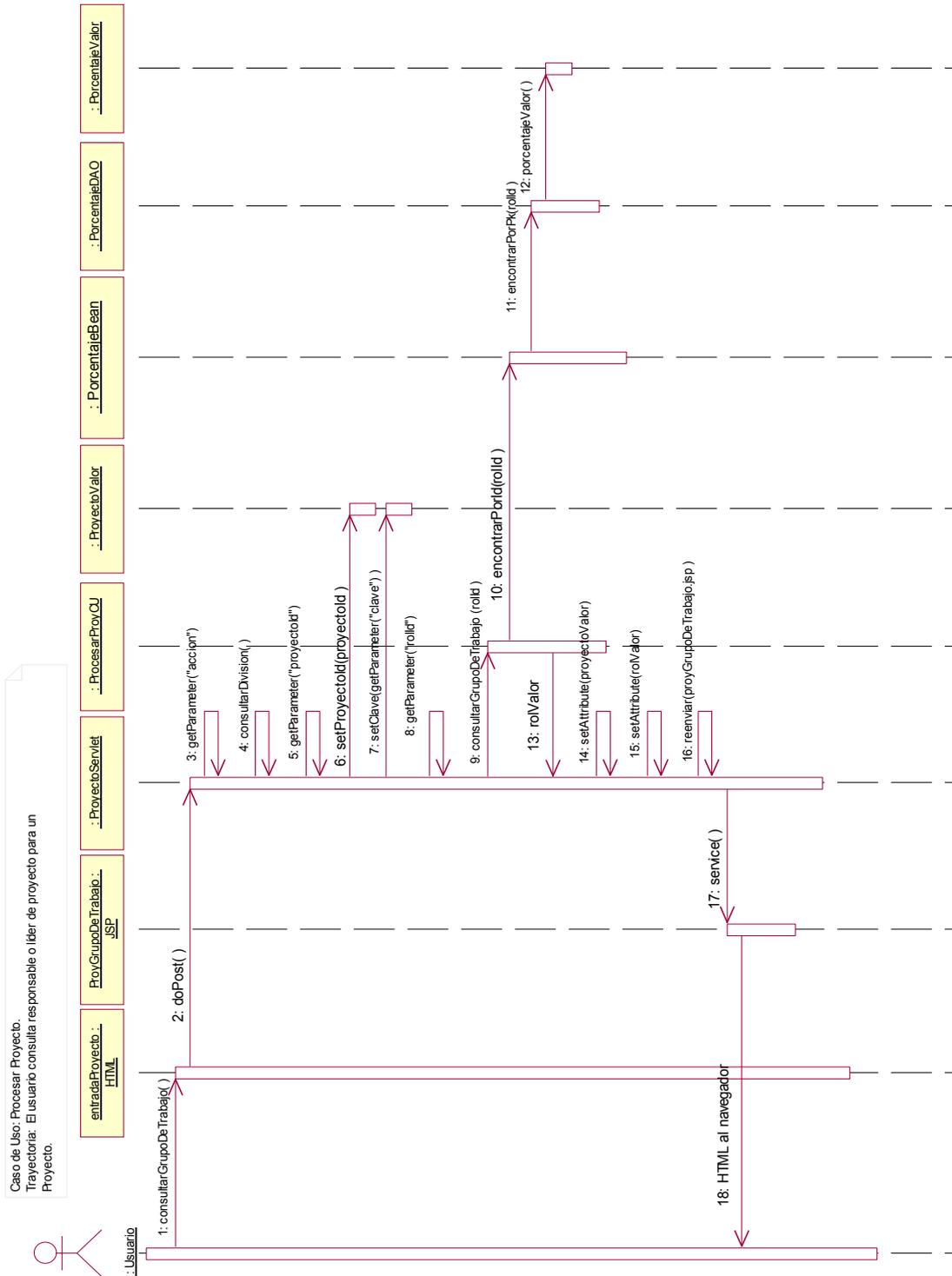


Figura 6.21 El usuario consulta Responsable o Líder de Proyecto para un proyecto

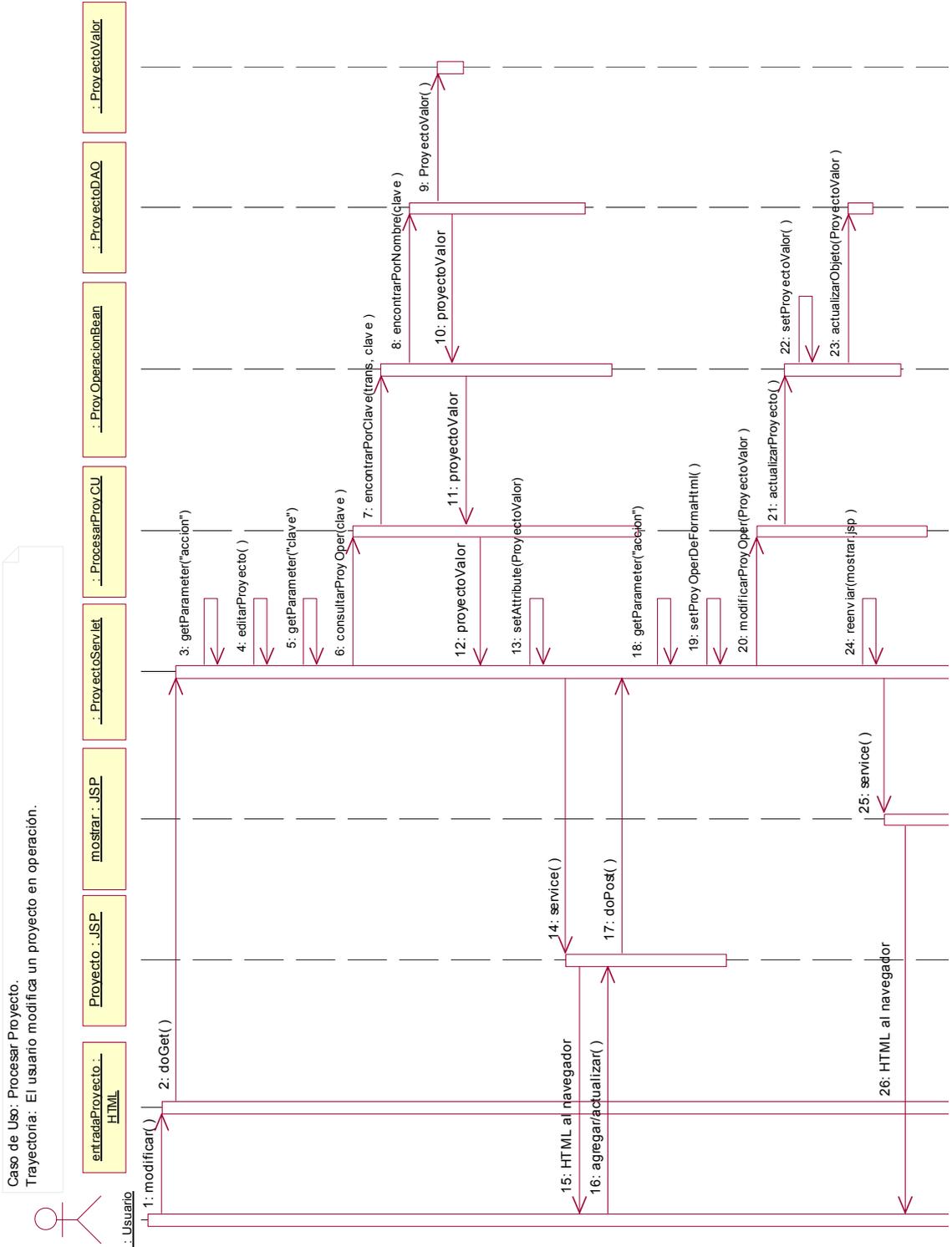


Figura 6.22 El usuario modifica un proyecto en operación

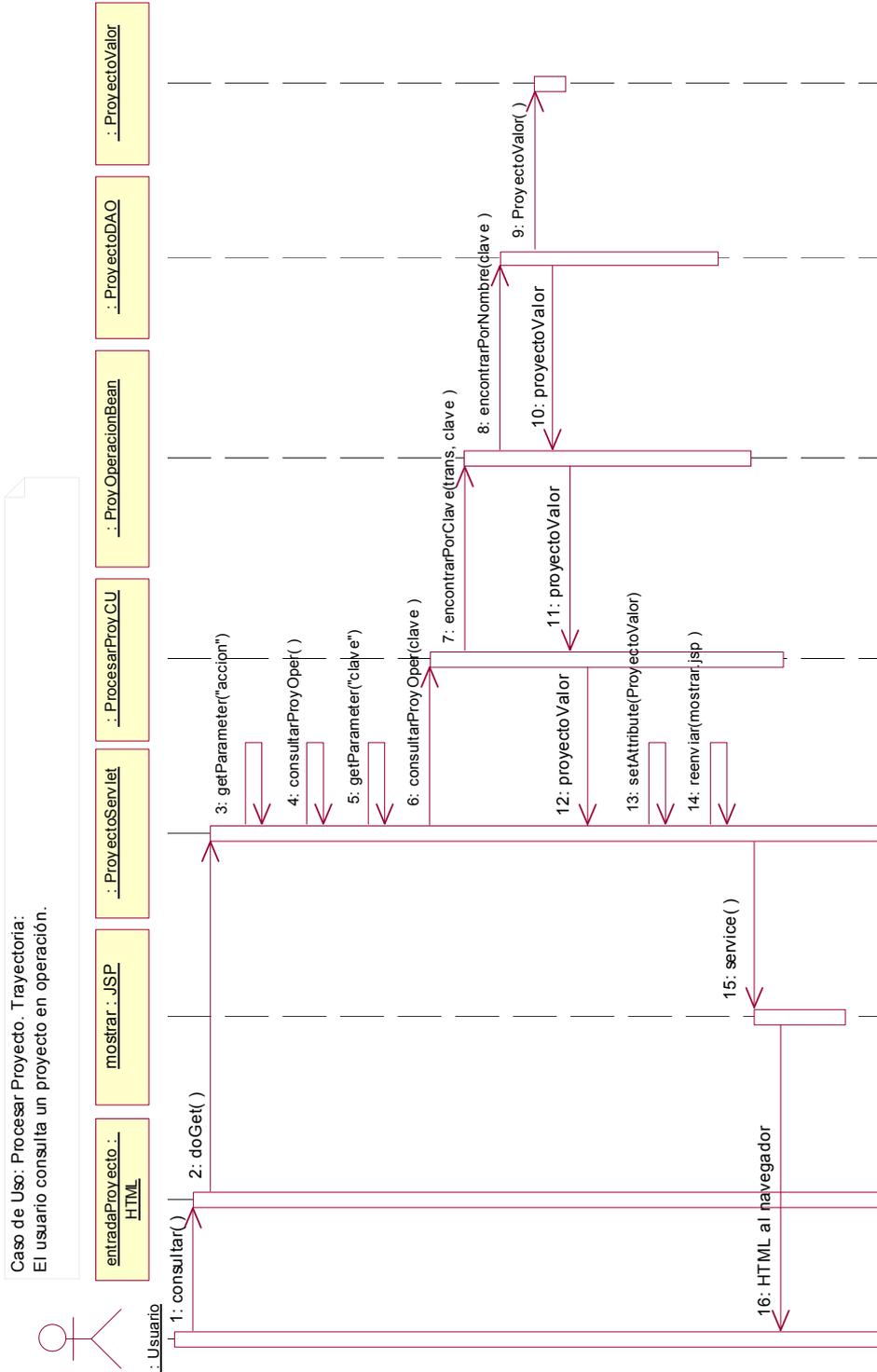


Figura 6.23 El usuario consulta un proyecto en operación

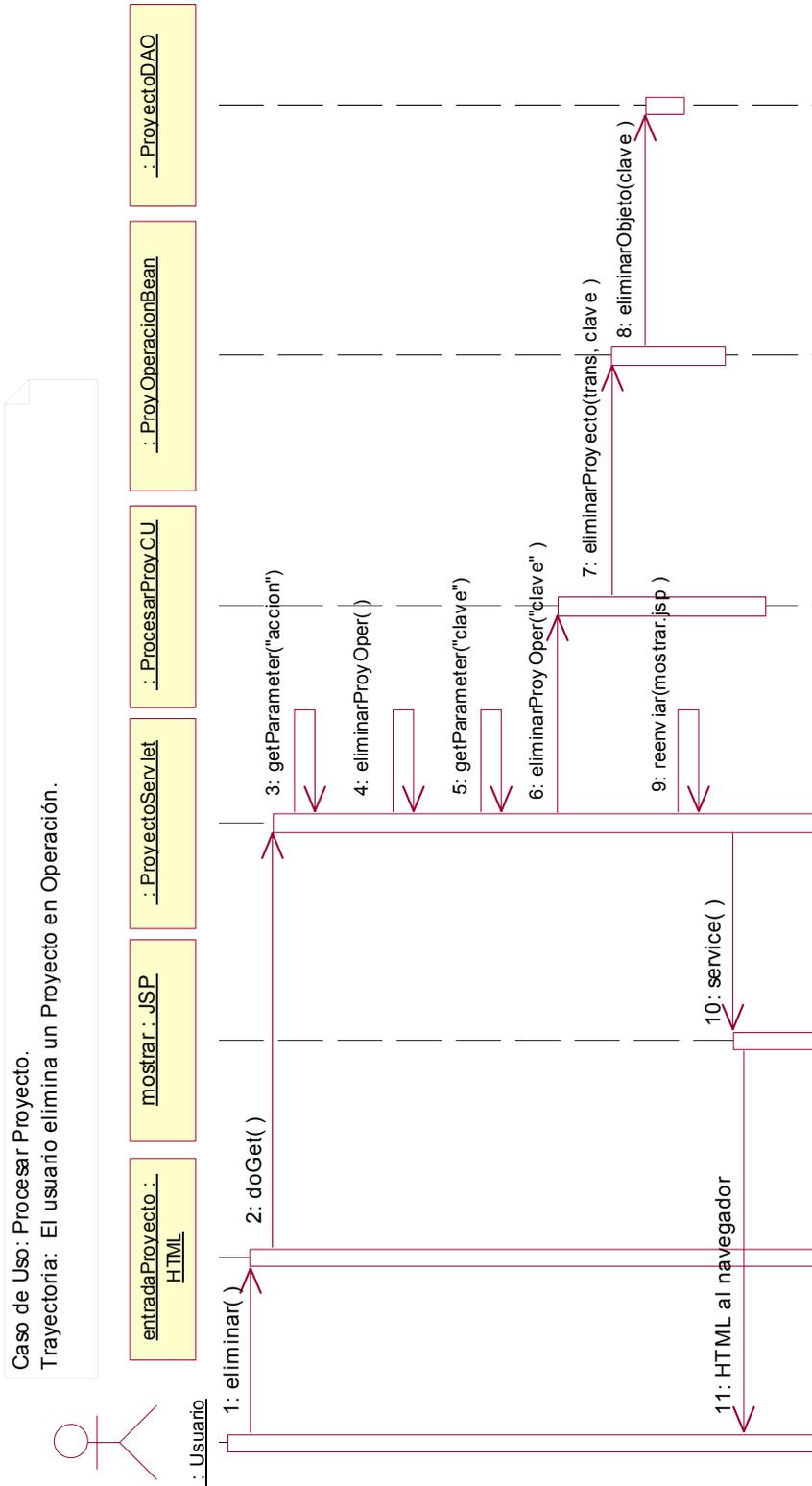


Figura 6.24 El usuario elimina un proyecto en operación

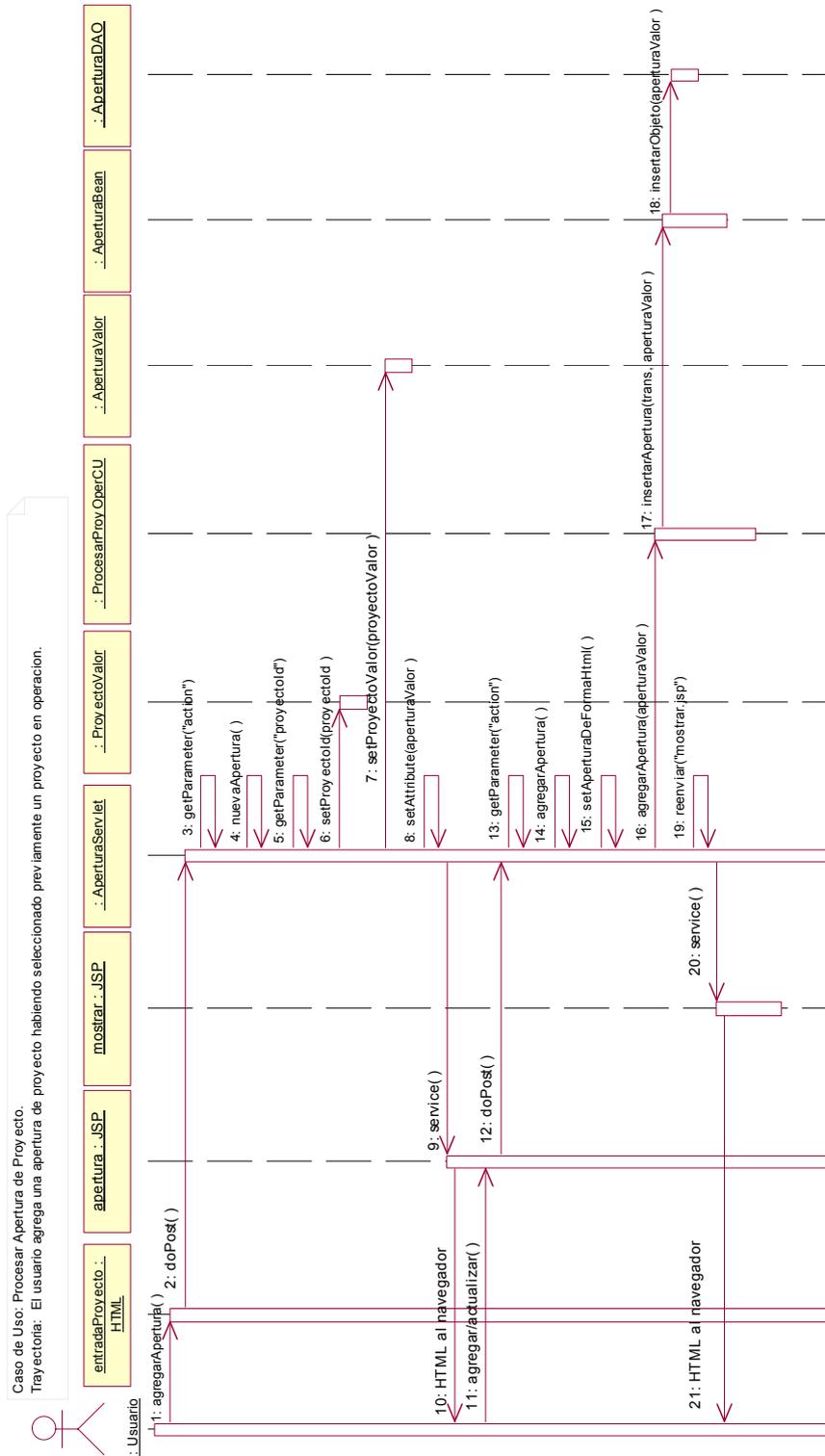


Figura 6.25 El usuario agrega una Apertura de Proyecto para un proyecto en operación

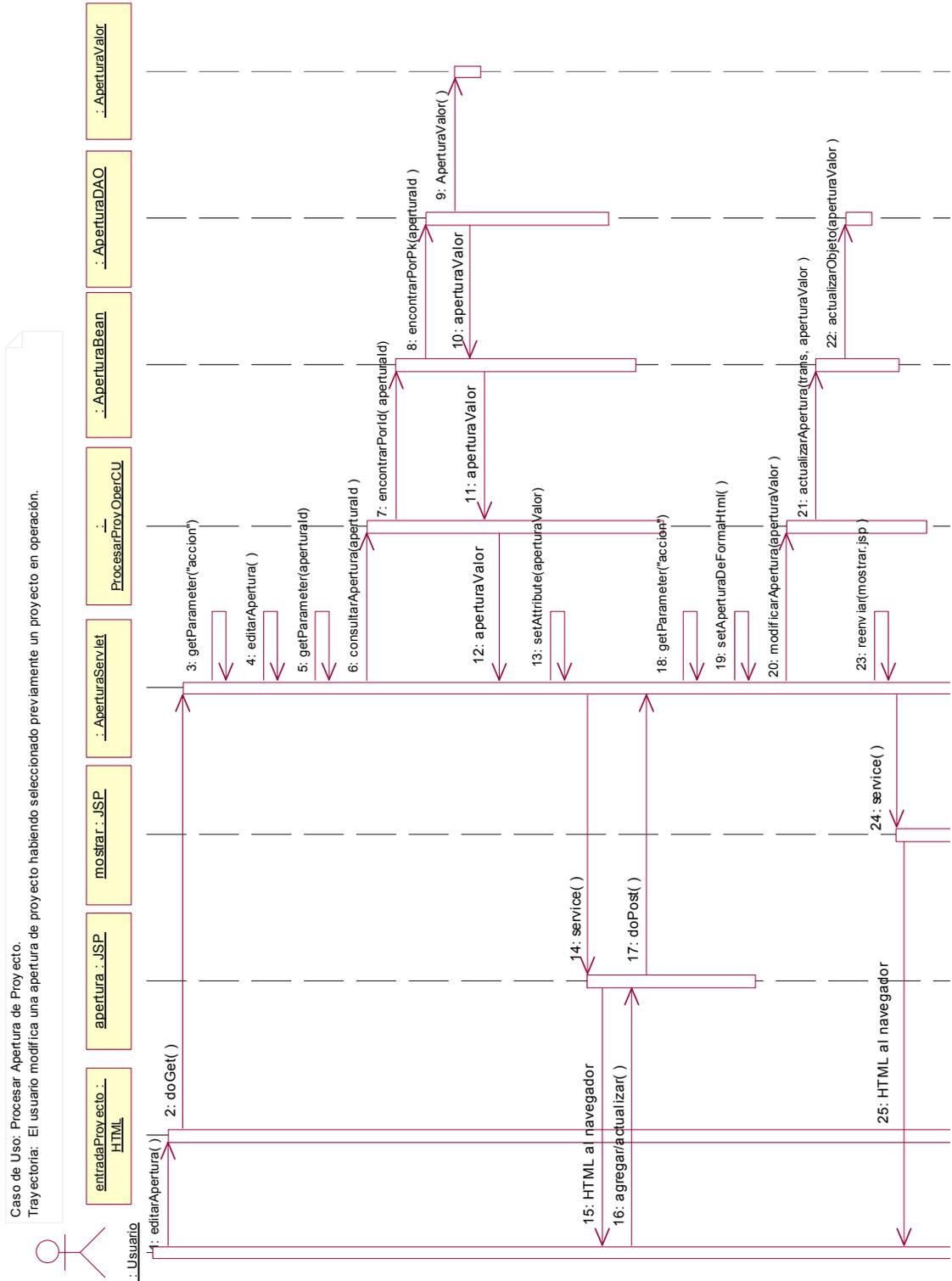


Figura 6.26 El usuario modifica una Apertura de Proyecto para un proyecto en operación

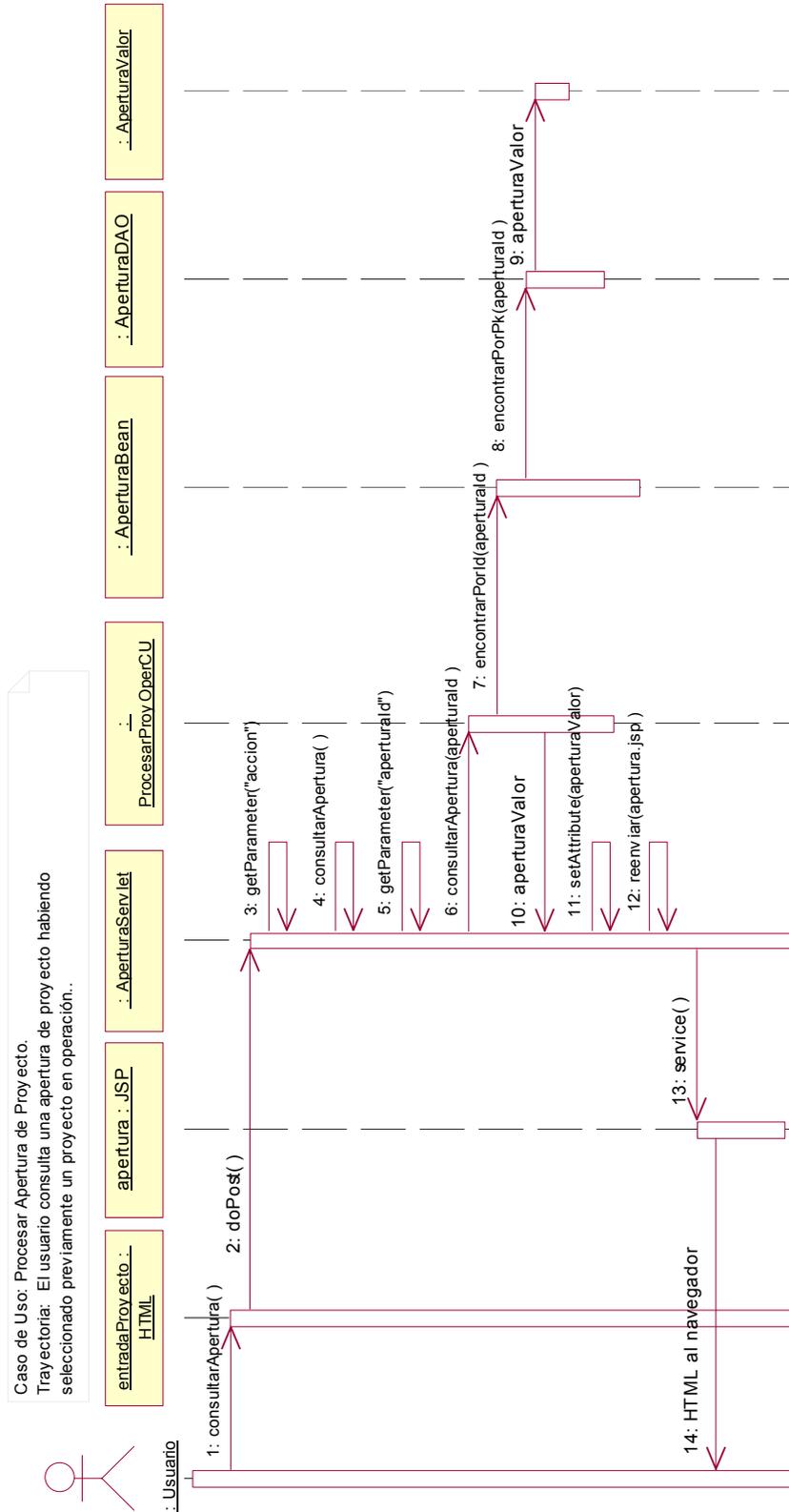


Figura 6.27 El usuario consulta una Apertura de Proyecto para un proyecto en operación

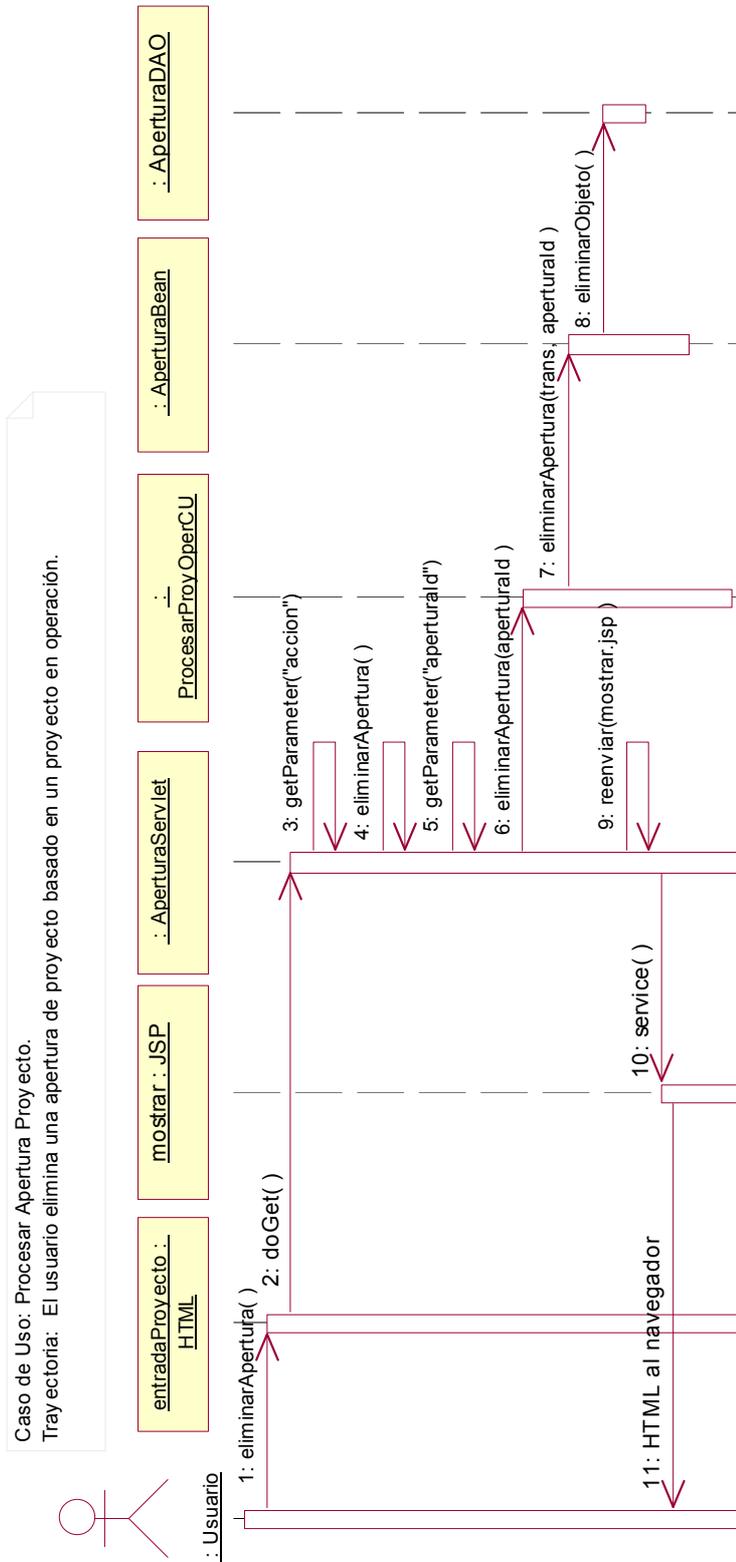


Figura 6.28 El usuario elimina una Apertura de Proyecto para un proyecto en operación

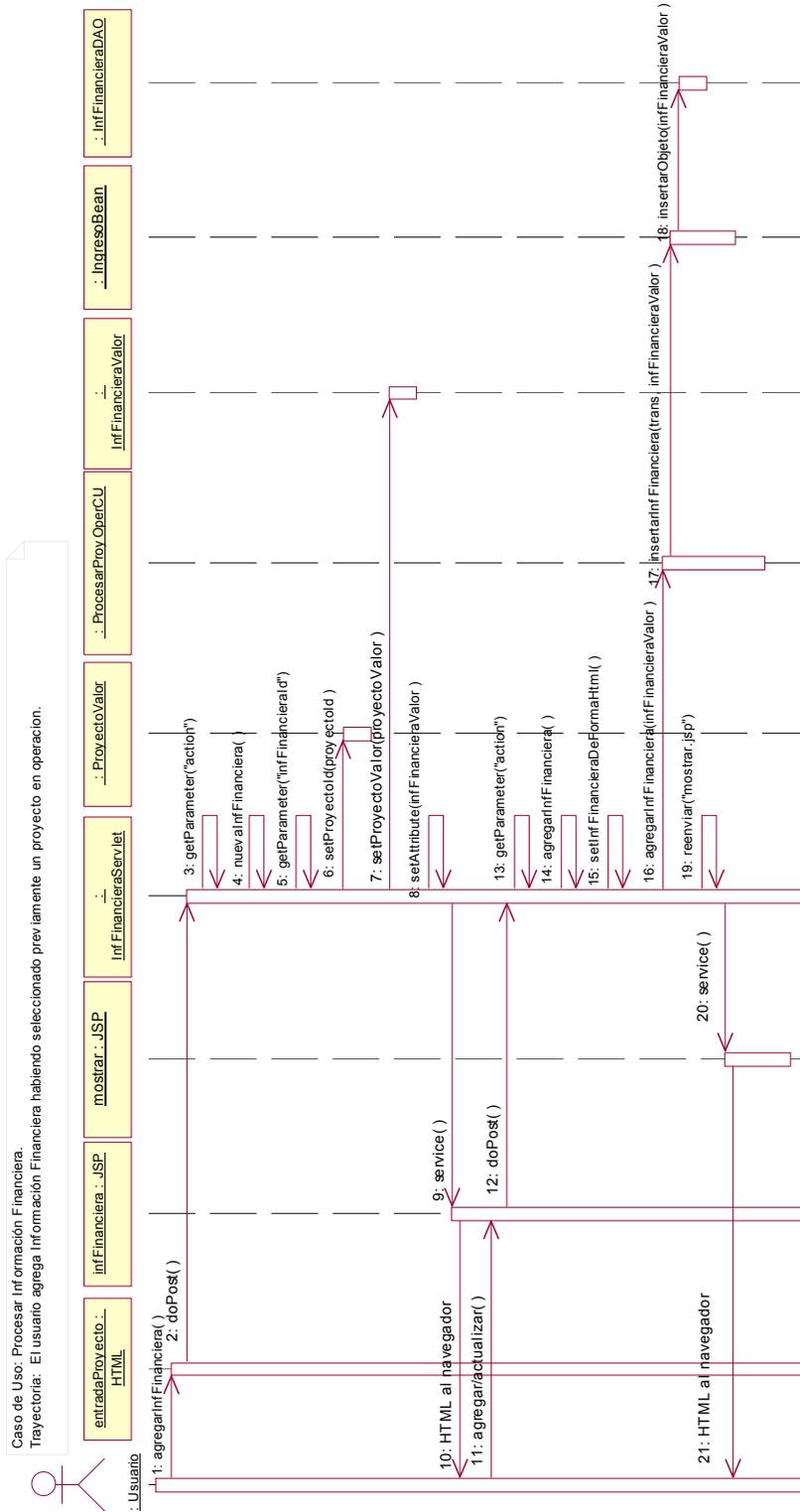


Figura 6.29 El usuario agrega Información Financiera para un proyecto en operación

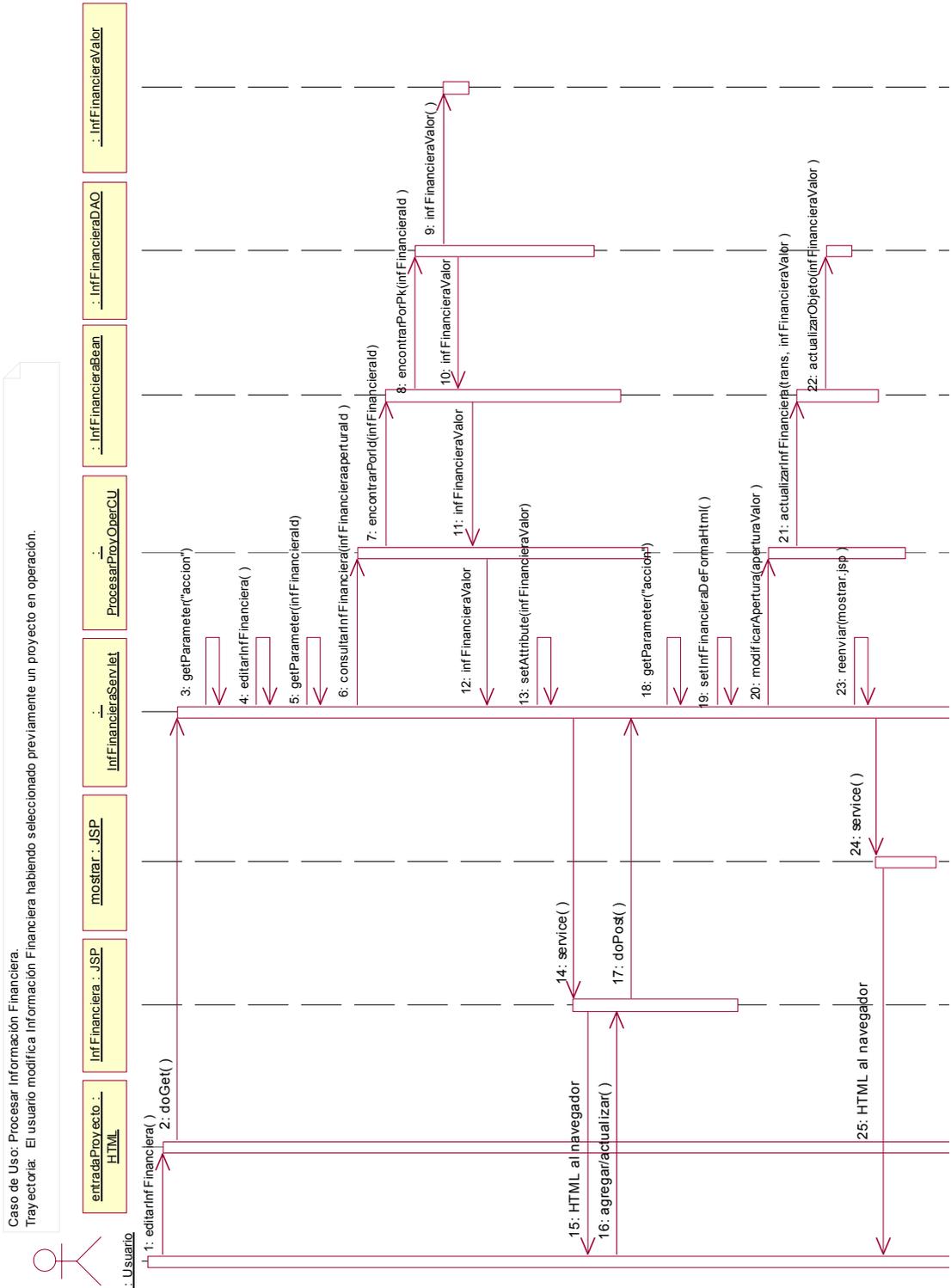


Figura 6.30 El usuario modifica Información Financiera para un proyecto en operación

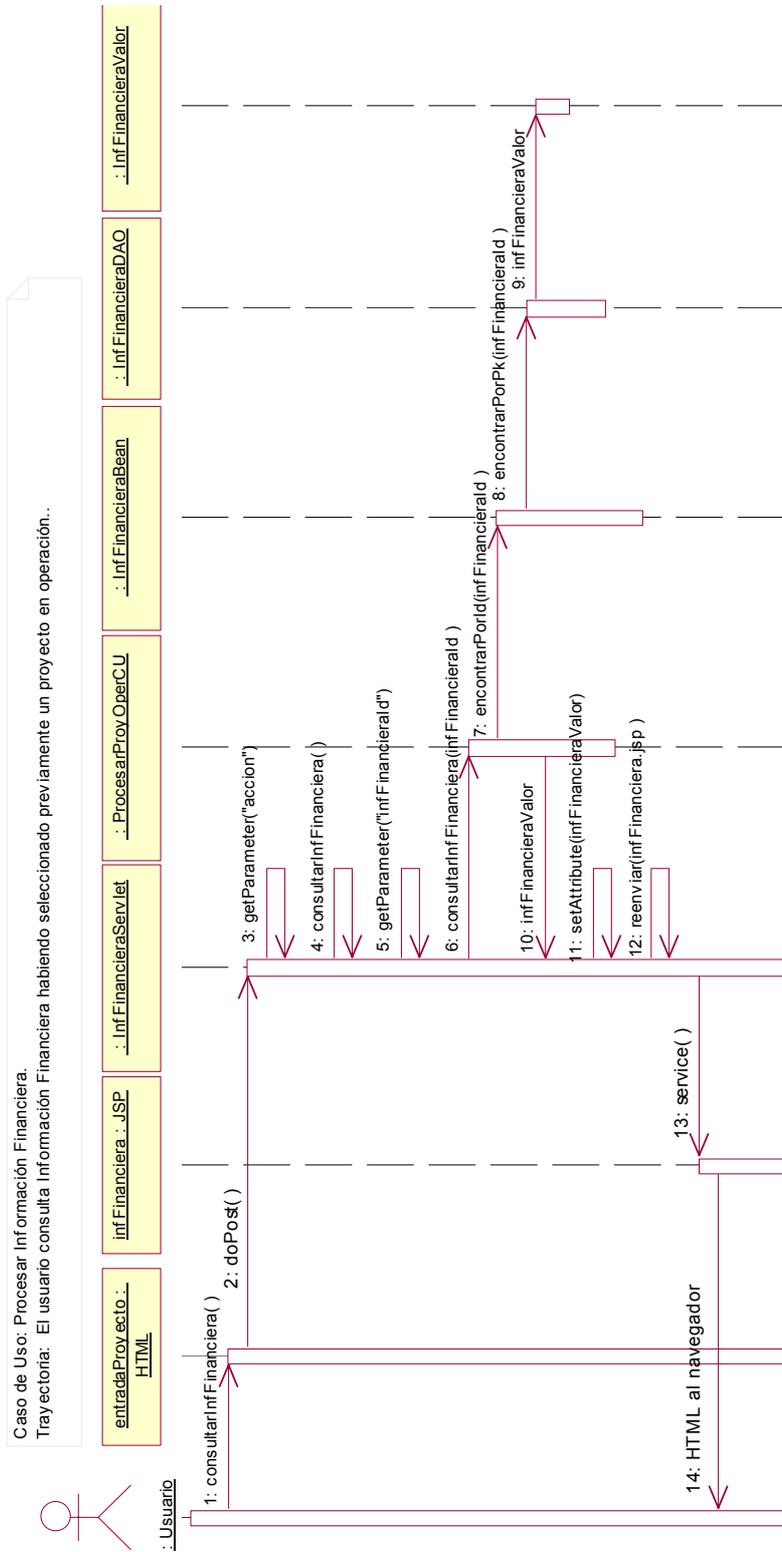


Figura 6.31 El usuario consulta Información Financiera para un proyecto en operación

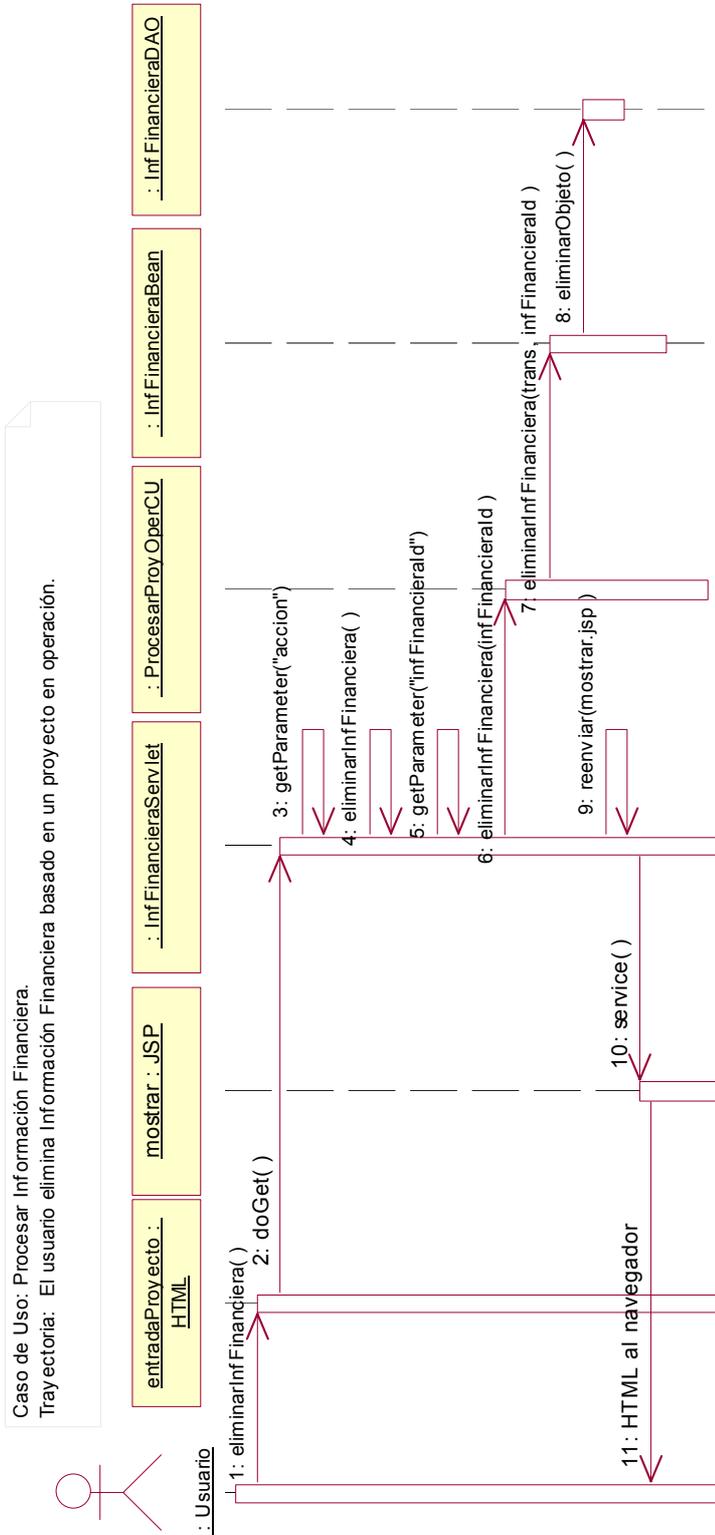


Figura 6.32 El usuario elimina Información Financiera para un proyecto en operación

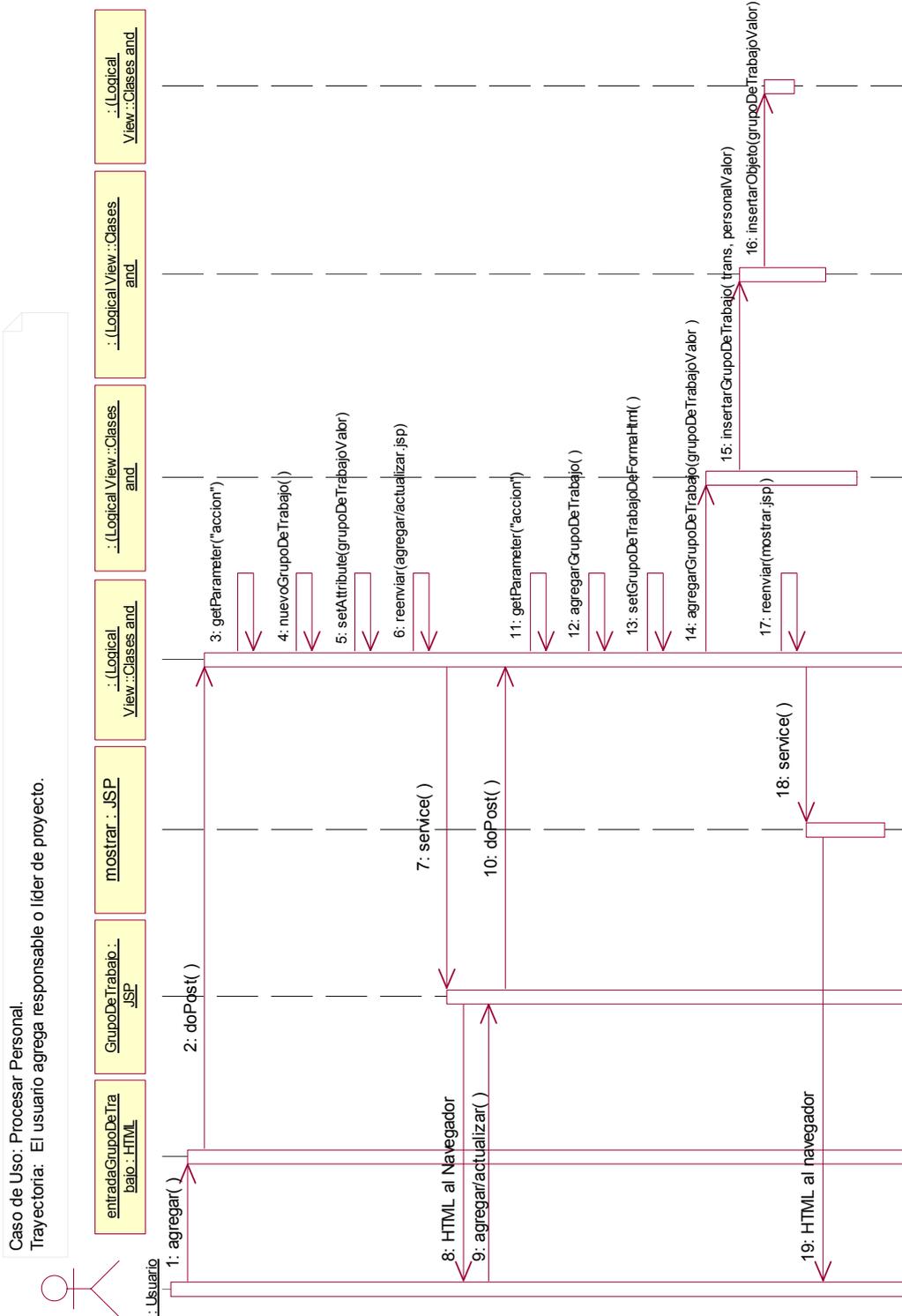


Figura 6.33 El usuario agrega Responsable o Líder de Proyecto

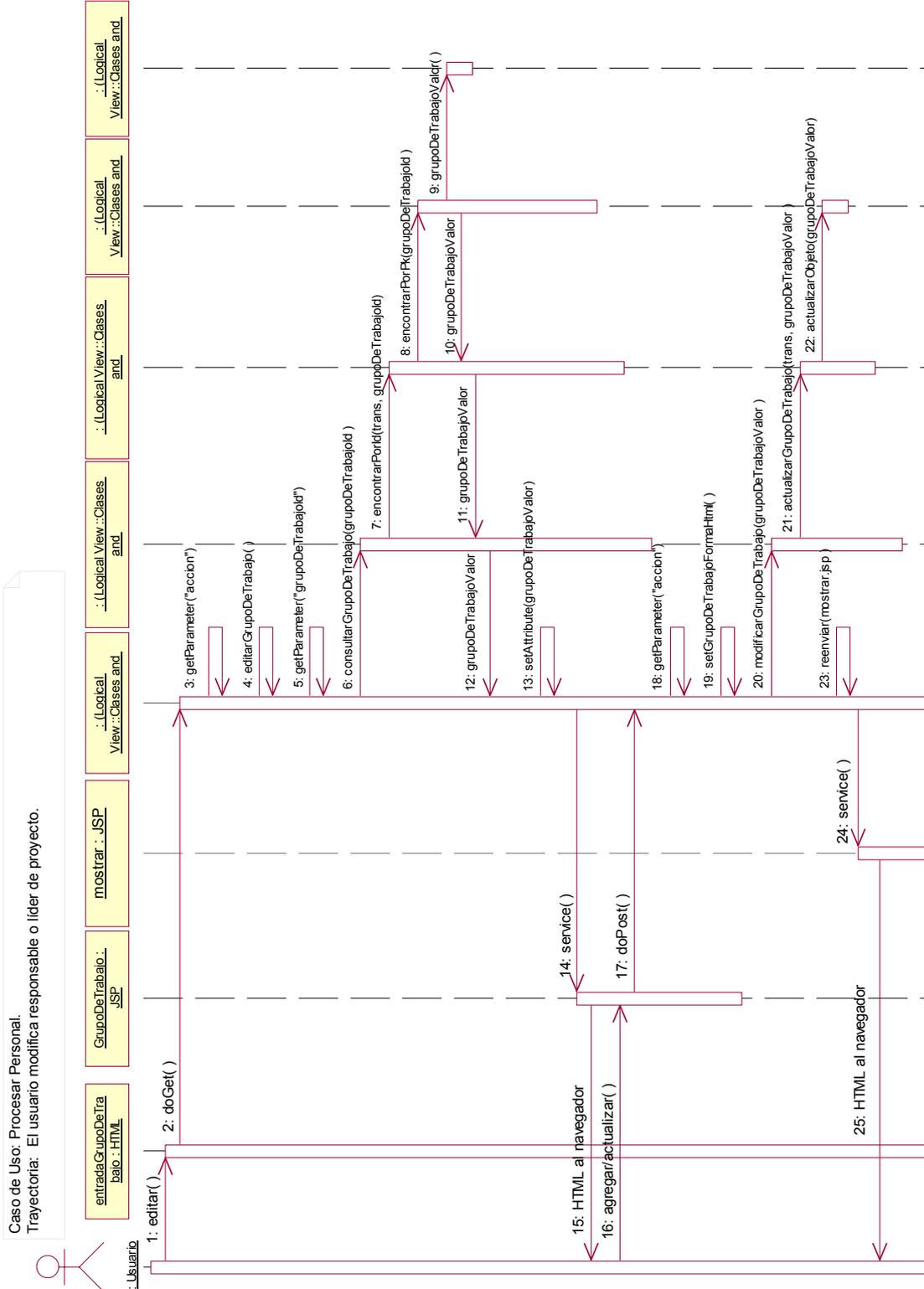


Figura 6.34 El usuario modifica Responsable o Líder de Proyecto

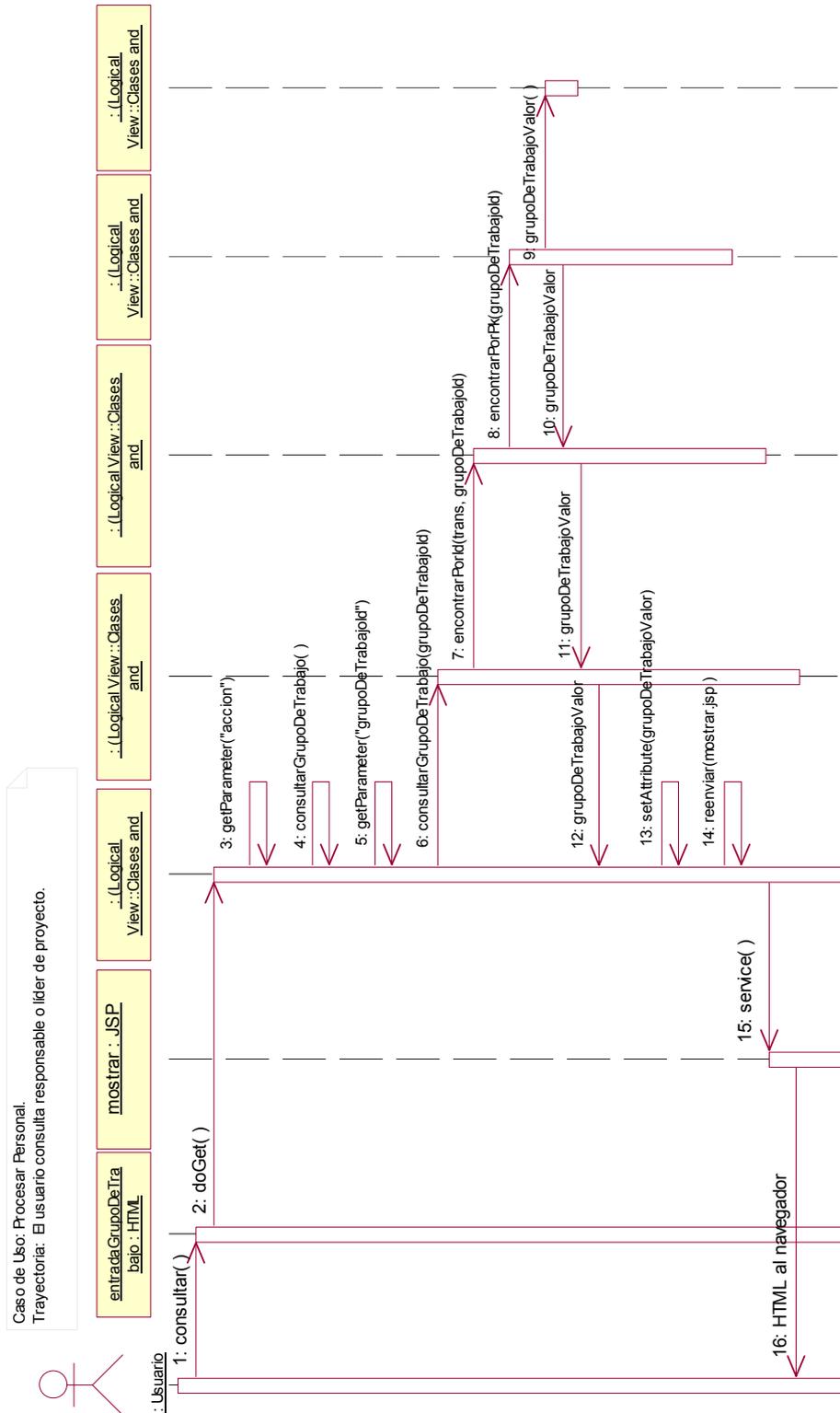


Figura 6.35 El usuario consulta Responsable o Líder de Proyecto

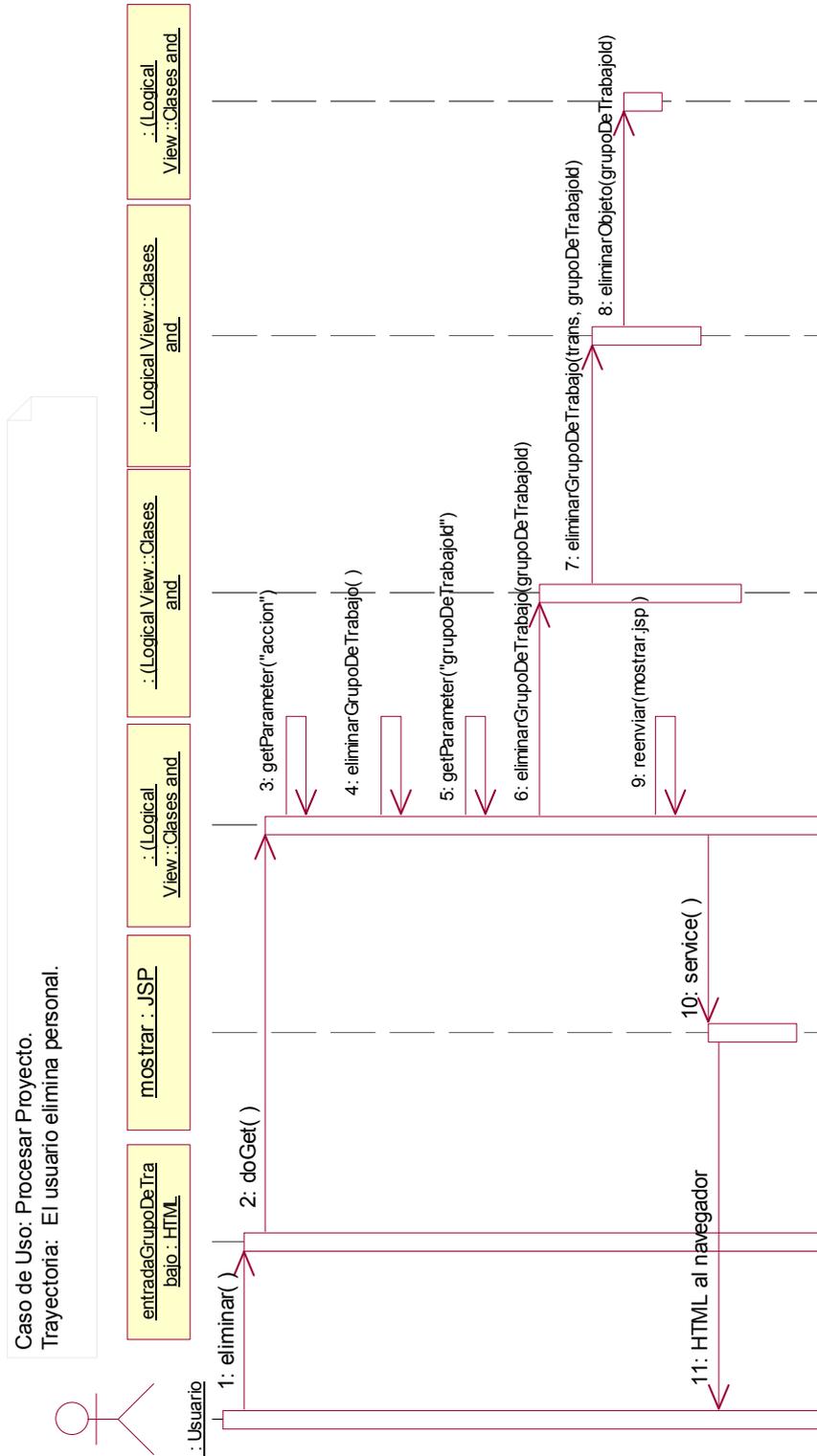


Figura 6.36 El usuario elimina Responsable o Líder de Proyecto

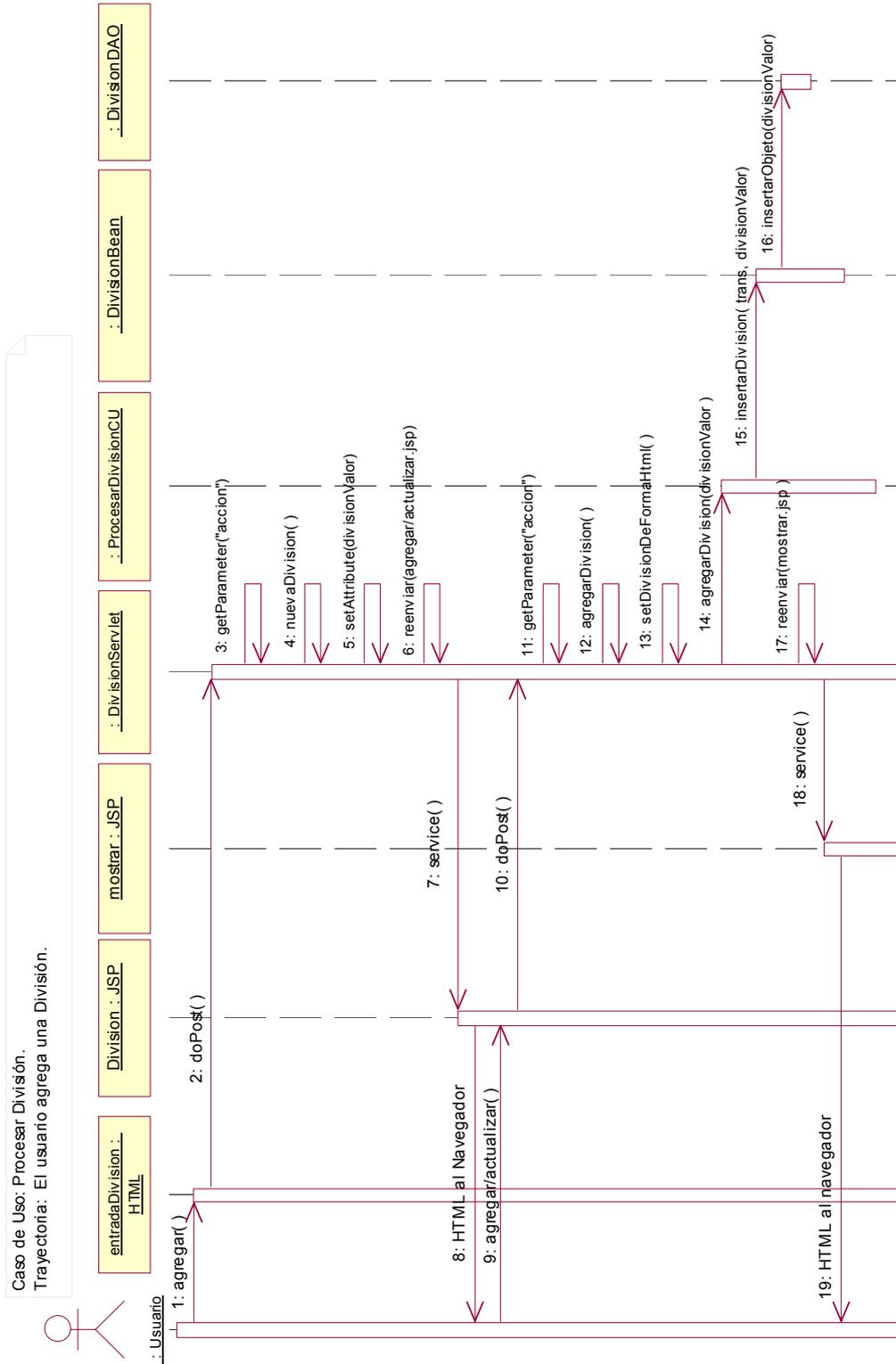


Figura 6.37 El usuario agrega una División

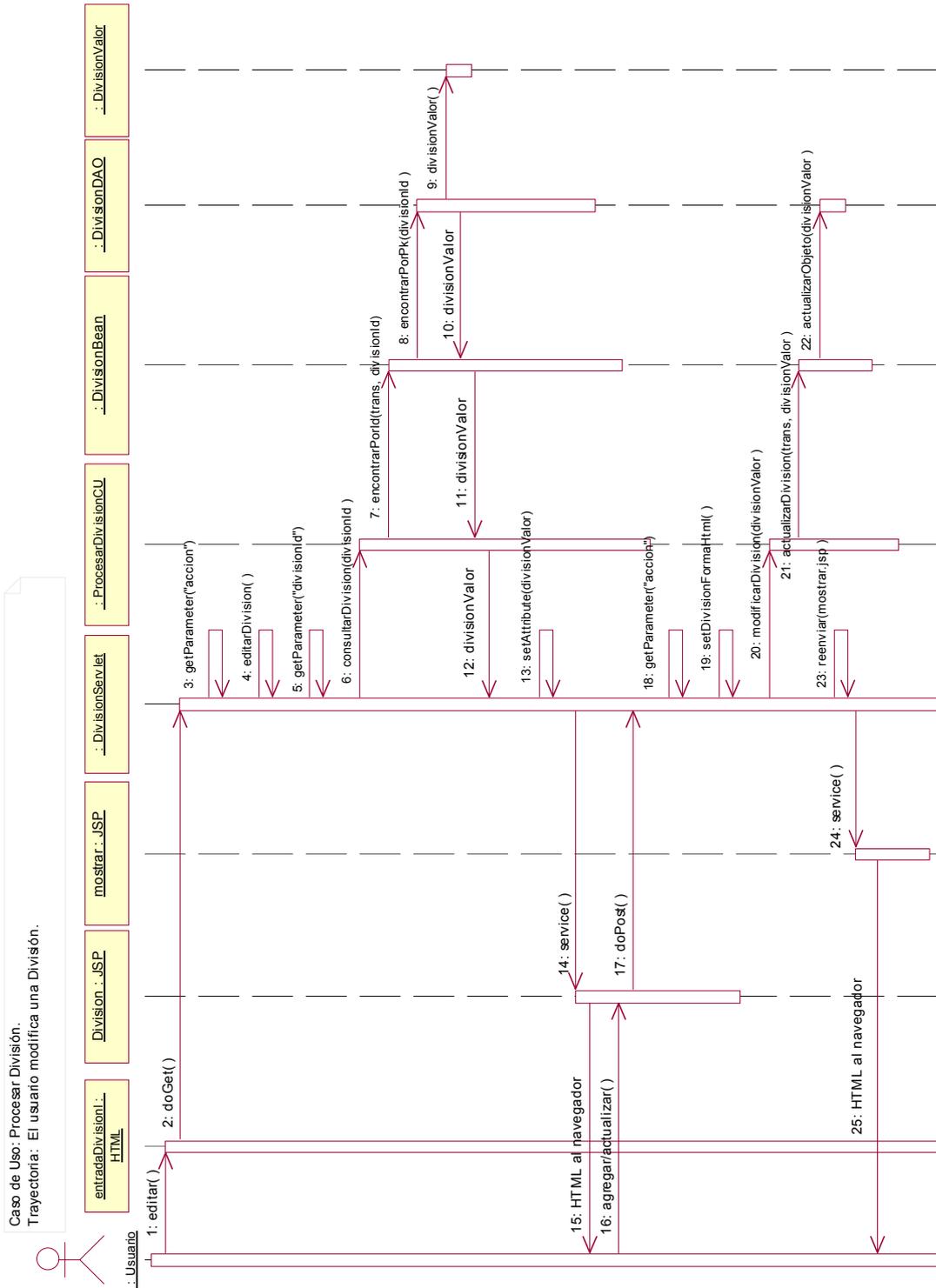


Figura 6.38 El usuario modifica una División

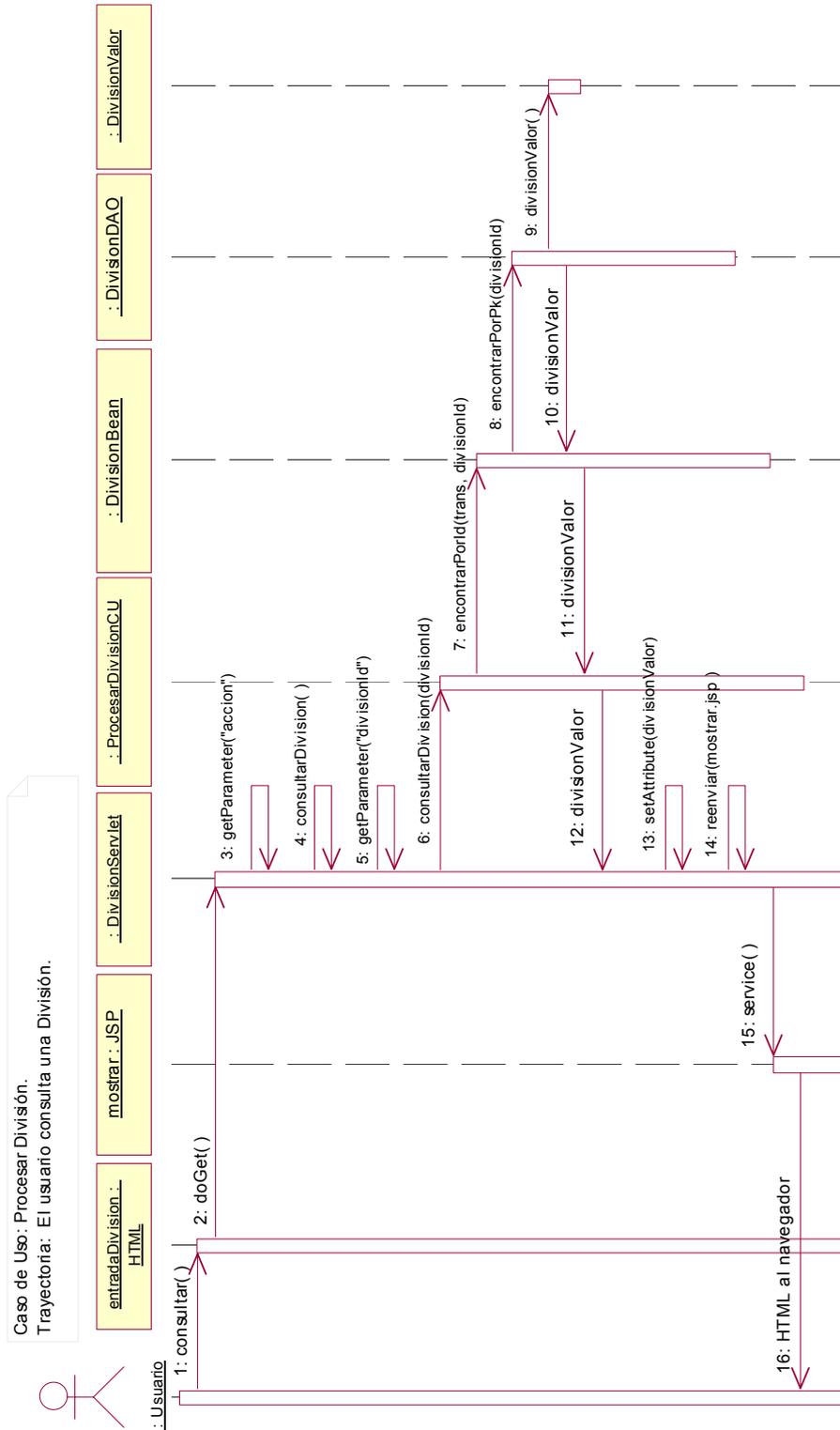


Figura 6.39 El usuario consulta una División

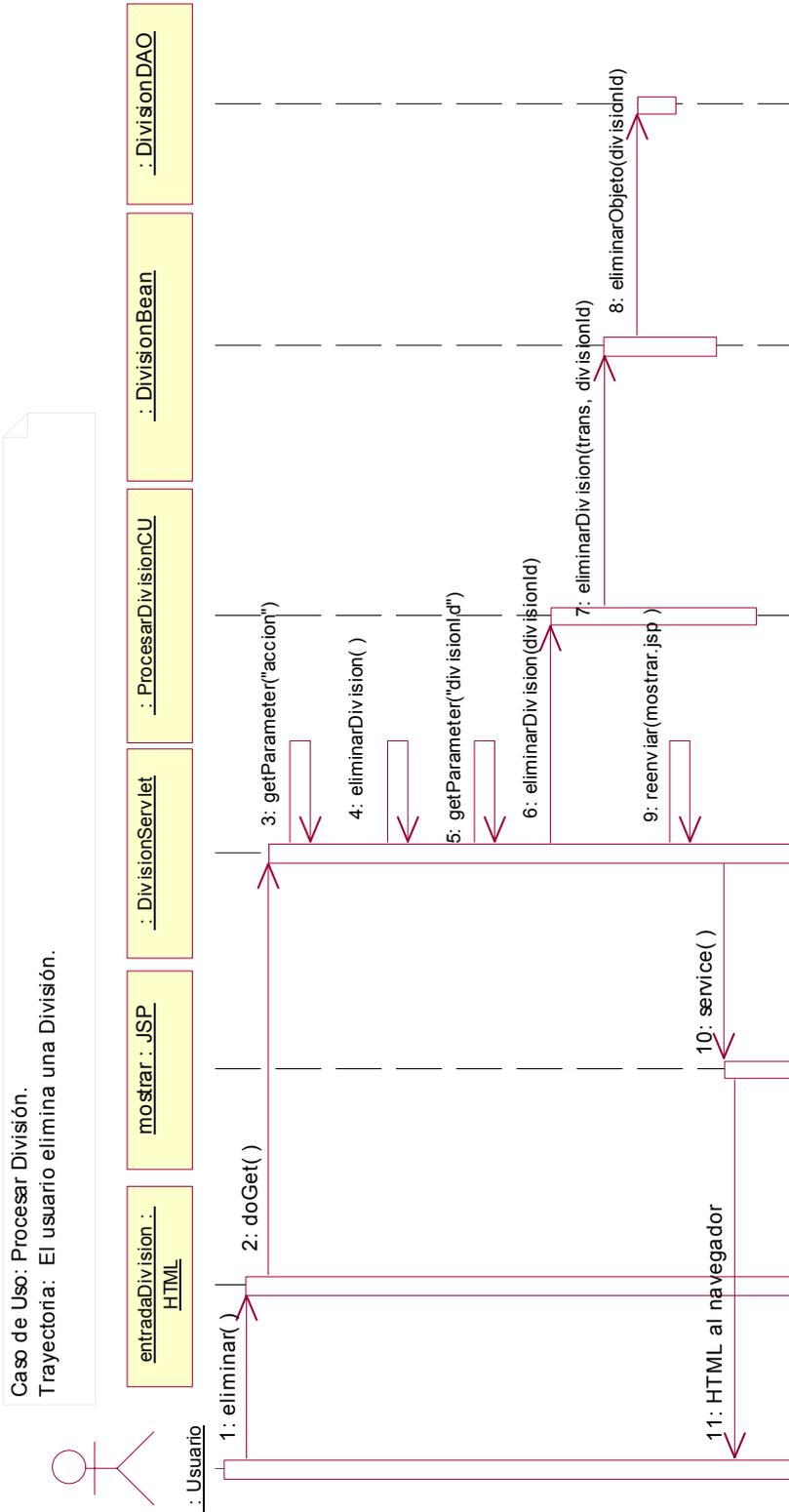


Figura 6.40 El usuario elimina una División

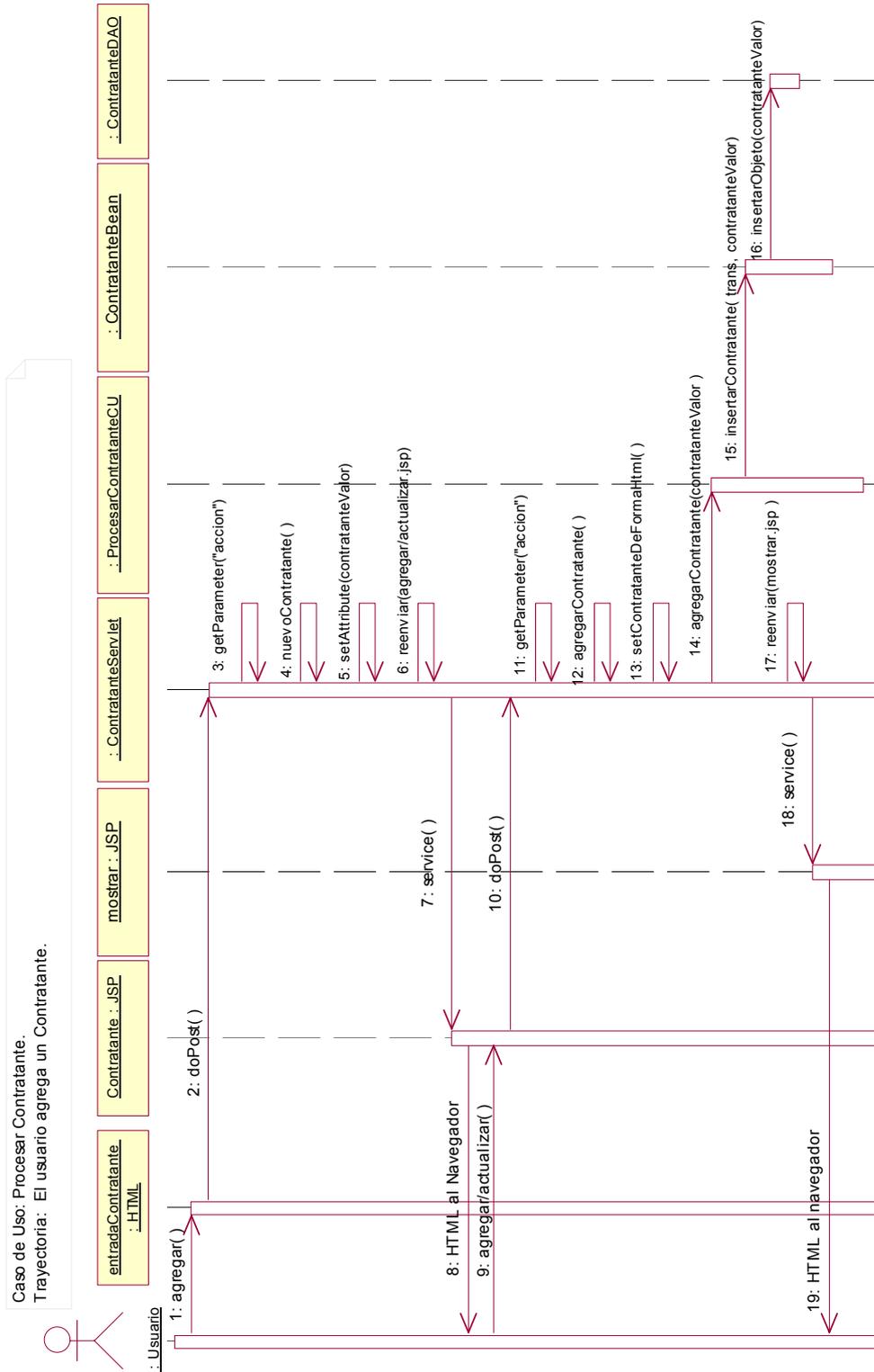


Figura 6.41 El usuario agrega un Contratante

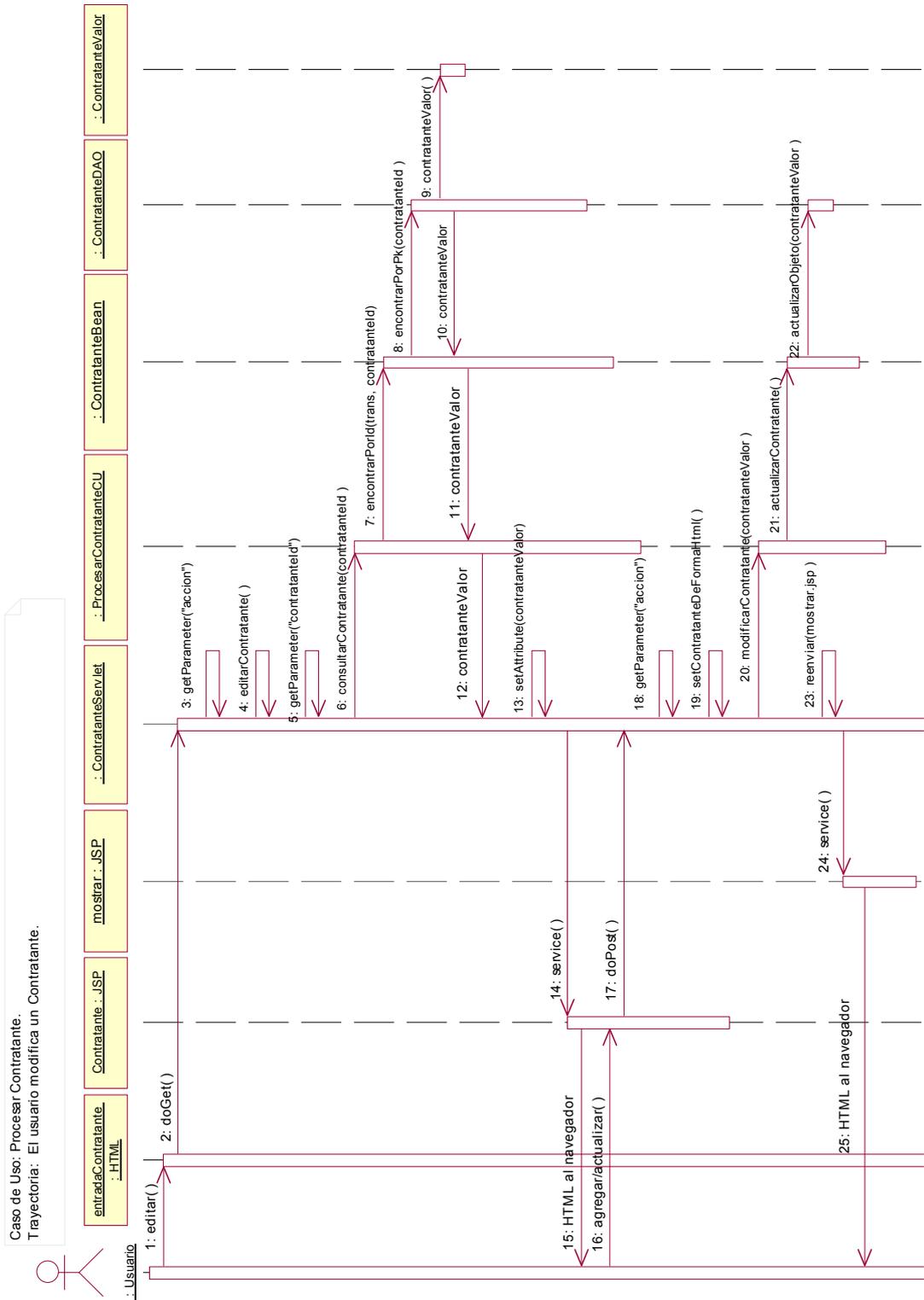


Figura 6.42 El usuario modifica un Contratante

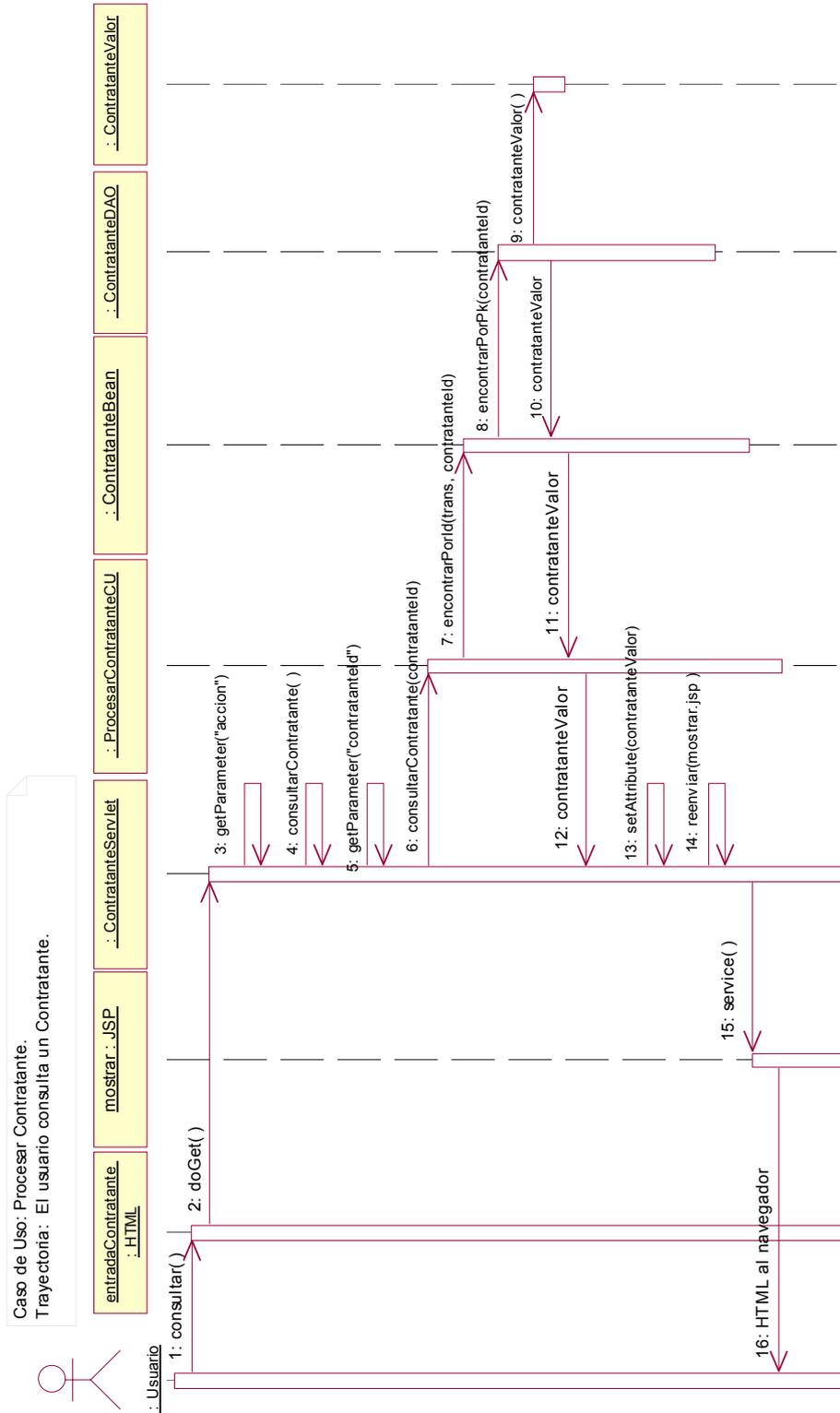


Figura 6.43 El usuario consulta un Contratante

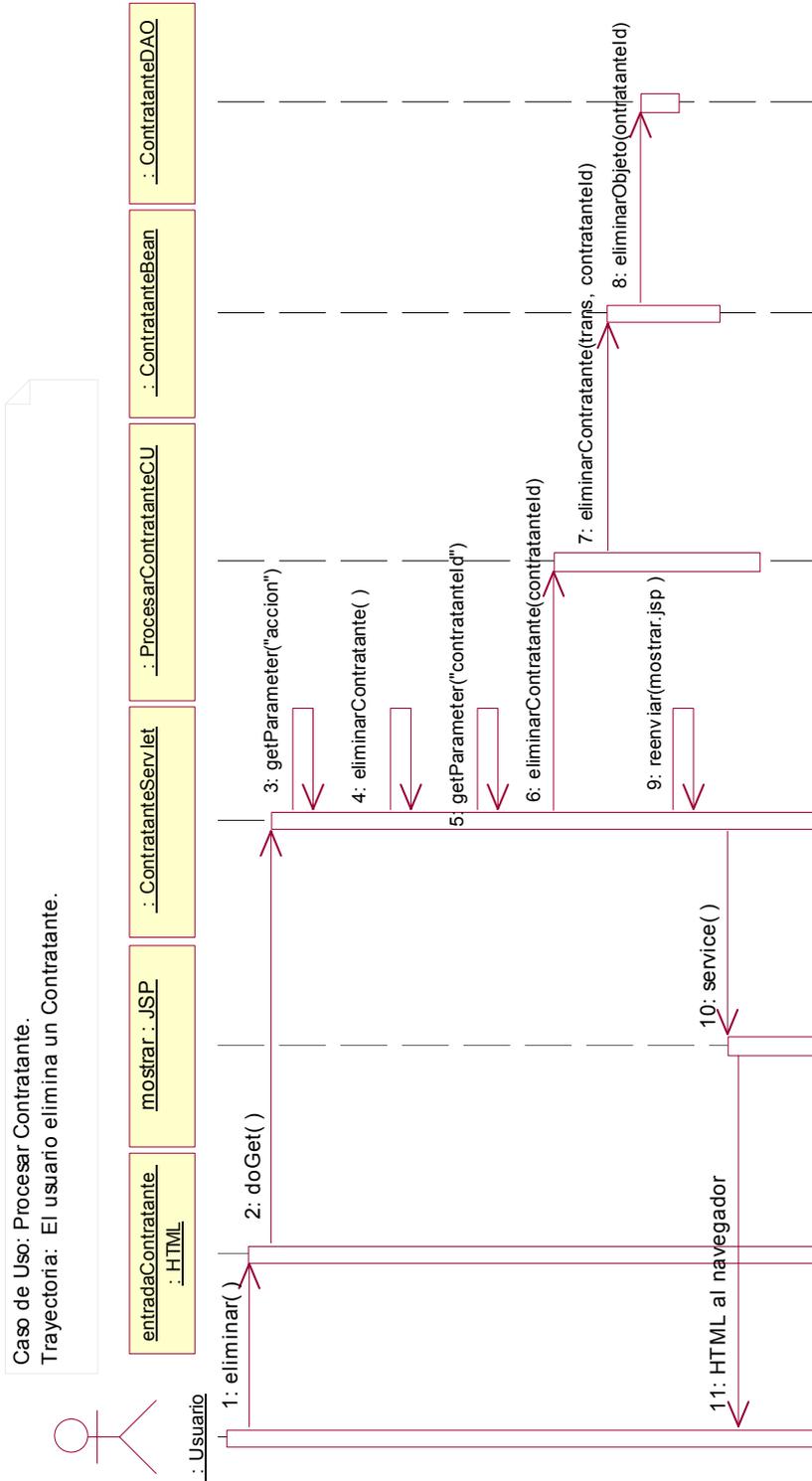


Figura 6.44 El usuario elimina un Contratante

6.3.3 Modelo entidad-relación

En este apartado analizaremos el diseño de la base de datos, el cual se obtuvo a partir del diagrama de clases.

A continuación vamos a realizar un mapeo de las clases de entidad, identificadas para el SIPIEX, a las tablas físicas en la base de datos.

El tener las relaciones de *clase-asociación* en el diagrama de clases nos facilitó el mapeo de las clases a tablas de la base de datos debido a que eliminaron las relaciones muchos a muchos obteniendo un modelo normalizado.

6.3.3.1 Mapeo entre clases y tablas de la base de datos

Para comenzar con el mapeo de clases necesitamos estar seguros de que el contenido de la clase debe ser persistente. Por ejemplo las clases de límite y las de control nunca serán almacenadas en tablas de la base de datos, pero las clases de entidad sí.

Debido a que la clase *proyPromocion* contiene pocos atributos para las relaciones de herencia, crearemos una tabla de la superclase (*proyecto*) y colocaremos cada columna de información de las subclases (*proyPromocion* y *proyOperacion*) en la tabla creada para la superclase, esto se conoce como roll-up. Lo anterior nos asegura que se genere el menor número de objetos en la base de datos (solo una tabla) por lo que obtendremos un mejor performance; la desventaja se presenta cuando se agreguen más subclases ya que se requerirán cambios en la tabla creada para la superclase.

A continuación mostramos el diagrama resultante del mapeo entre las clases de entidad mostradas anteriormente y las tablas de la base de datos del sistema, Figura 6.45.

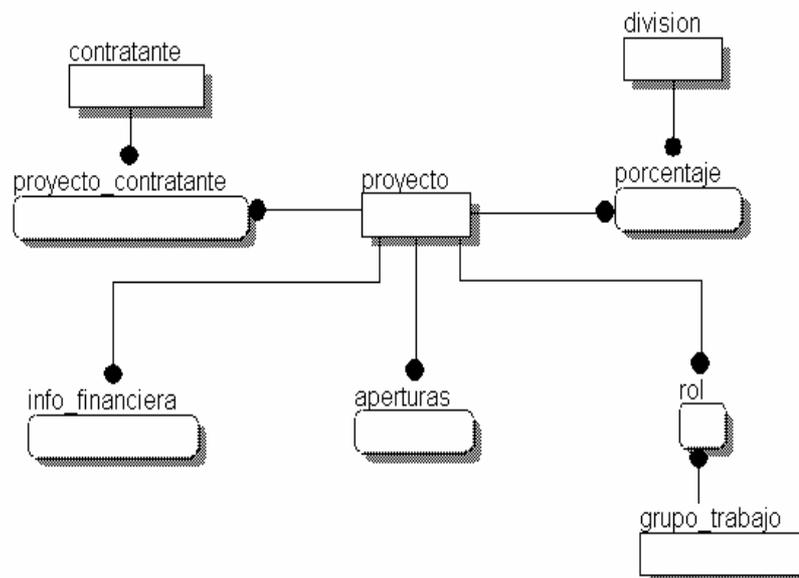


Figura 6. 45 Diagrama entidad relación

6.3.3.2 Diagrama entidad relación

Cada objeto en el sistema debe ser único, aunque esto no tiene por qué preocuparnos, ya que la máquina virtual de java genera identificadores únicos para cada objeto durante todo su tiempo de vida en el sistema. Sin embargo, esto no ocurre cuando la información se inserta en la base de datos, por lo que tenemos que asegurarnos que cada registro en una tabla sea único.

Con base en el diagrama de clases mostrado en la Figura 6.46 obtuvimos el diagrama de entidad-relación completo.

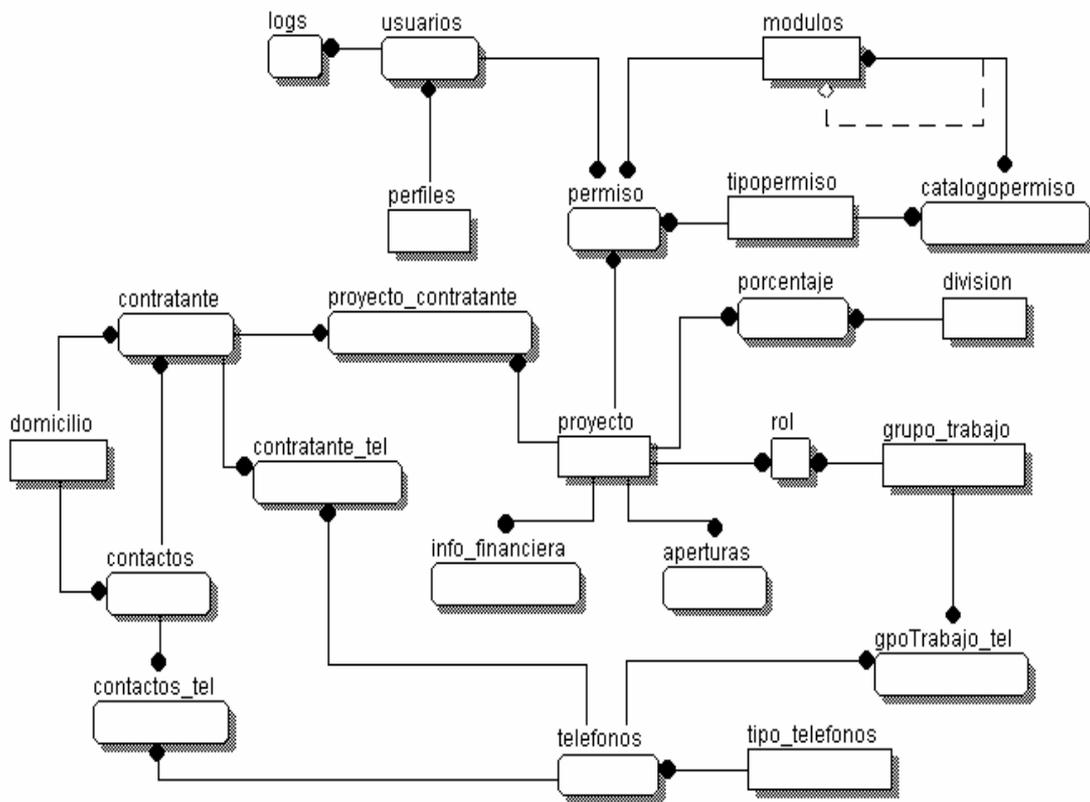


Figura 6. 46 Diagrama de entidad-relación para el SIPIEX

6.3.3.3 Diccionario de datos

El diccionario de datos es una explicación detallada de cada tabla y de cada campo de nuestra base de datos. La Tabla 6.8 muestra el diccionario de datos para el SIPIEX (para más detalles de los grupos y subgrupos de gasto ver Anexo A).

Tabla	Columna	Tipo Dato	Índice	Comentario
proyecto Tabla que permite	proyectoId	int	PK	identificador de proyecto
	clave	text		clave de proyecto
	importe	int		importe de proyecto

Tabla	Columna	Tipo Dato	Índice	Comentario
guardar la información general de un proyecto, ya sea en promoción o en operación.	nombre	text		nombre de proyecto
	objetivo	text		objetivo de proyecto
	duracion	text		duración de proyecto
	estatusConv	text		estatus del convenio del proyecto
	avanceFisico	text		porcentaje de avance del proyecto
	ligaContrato	text		link de la pagina web del contrato
	fIniConv	date		fecha de inicio del convenio
	fFinConv	date		fecha de término del convenio
grupo_trabajo	noConvenio	text		número del convenio del proyecto
	estatusProy	text		indica si el proyecto está en promoción o de operación
	grupo_trabajoId	int	PK	identificador de Responsable o Líder de Proyecto
	nombre	text		nombre de la persona responsable o líder del proyecto
Tabla que permite almacenar las personas que participan en un proyecto.	ubicacion	text		ubicación de la persona responsable o líder del proyecto
	telefono_id	int	FK	identificador del teléfono donde se puede localizar a la persona
	email	text		email de la persona
Rol	rolId	int	PK	identificador del rol de una persona
	proyectoId	int	FK	identificador del proyecto
	grupo_trabajoId	int	FK	identificador de la persona responsable o líder del proyecto
	nombreRol	text		nombre del rol de la persona (responsable, líder de proyecto)
contratante	contratanteId	int	PK	identificador de la entidad contratante
	domicilioId	Int	FK	identificador de la ubicación de la entidad contratante
	contactoId	int	FK	encargado por parte de la entidad contratante
	telefonoId	int	FK	identificador de teléfono de la entidad contratante
	nombre abreviatura	text text		nombre de la entidad contratante abreviatura de la entidad contratante
proyecto_contratante	proyContId	int	PK	identificador de proyecto-contratante
	proyectoId	int	FK	identificador proyecto
	contratanteId	int	FK	identificador contratante
division	divisionId	int	PK	identificador de la división
	personaId	int	FK	identificador de la persona a cargo de la división
	nombre	text		nombre de la división
	abreviatura	text		abreviatura de la división
porcentaje	porcentajeId	int	PK	identificador de porcentaje
	proyectoId	int	FK	identificador proyecto
	divisionId	int	FK	identificador división
	porcentaje	int		porcentaje monetario de una división en un proyecto
info_financiera	inffinancieraId	int	PK	identificador de información financiera
	proyectoId	int	FK	identificador de proyecto
	tipoTramite	text		clasificación del trámite (depósito)

Tabla	Columna	Tipo Dato	Índice	Comentario
información financiera	concepto	text		causa o nombre del gasto
	Fecha	date		fecha de la operación financiera
	importe	int		importe de la operación financiera
	Tipo	text		indica si es un ingreso o un egreso (grupo 100 al 600)
Apertura	aperturaId	int	PK	identificador apertura de proyecto
Tabla que permite registrar las aperturas de los proyectos	proyectoId	int	FK	identificador proyecto
	benfAcademicos	text		beneficios académicos del proyecto
	benfEconomicos	text		beneficios económicos del proyecto
	futProyectos	text		futuros proyectos
usuarios_proyectos	usuarioId	int	FK	identificador del usuario
tabla transitiva entre la tabla de proyecto y la tabla de usuarios	proyectoId	int	FK	identificador de proyecto
	permiso	text		tipo de permiso que el usuario tiene para el acceso a la información siendo este de lectura o escritura
usuarios	usuarioId	int	PK	identificador del usuario
Tabla que permite almacenar los usuarios que acceden al sistema	perfilId	int	FK	identificador del perfil del usuario
	usuario	text		usuario
	password	text		password del usuario
	nombre	text		nombre del usuario
	apellido	text		apellido del usuario
permiso	usuarioId	int	FK	identificador del usuario
Tabla que permite almacenar los permisos de cada usuario	proyectoId	int	FK	identificador del proyecto
	tipopermisoId	int	FK	identificador del tipo de permiso
	moduloId	int	FK	identificador del módulo
perfiles	perfilId	int	PK	identificador del perfil
Tabla donde se almacenan los perfiles predeterminados	perfil	tex		listado de los perfiles predeterminados
catalogopermiso	catalogopermisoId	int	PK	identificador del catálogo permiso
Tabla que almacena los permisos para los perfiles predeterminados	perfilId	int	FK	identificador del perfil
	tipopermisoId	int	FK	identificador del tipo de permiso
	moduloId	int	FK	identificador del modulo
tipopermiso	tipopermisoId	int	PK	identificador del tipo de permiso
Tabla que almacena los tipos de permiso	permiso	text		nombre del permiso
modulos	moduloId	int	PK	identificador del módulo
Tabla que almacena los módulos y submódulos con los que cuenta el sistema.	modulo	text		nombre del módulo
	modulo_id_padre	int	FK	identificador del módulo al que pertenece
logs	logId	int	PK	identificador del log del usuario
Tabla que permite mantener el control de las acciones que realizan los usuarios	usuario	text	FK	nombre del usuario
	accion	text		acción que realiza el usuario
	Fecha	date		fecha en que se realizo la acción
contactos	contactoId	int	PK	identificador de contacto
Tabla que permite almacenar los	domicilioId	int	FK	identificador de domicilio
	telefonoId	int	FK	identificador de teléfono
	nombre	text		nombre de de la contacto

Tabla	Columna	Tipo Dato	Índice	Comentario
responsables de la entidad contratante o de la división.	apellido	text		Apellido paterno del contacto
	cargo	text		Cargo o puesto que desempeña
	email	text		email de la persona
domicilio	domicilioId	int	PK	identificador de domicilio
Tabla que permite almacenar direcciones	domicilio	text		domicilio
	colonia	text		colonia
	codigoPostal	text		código postal de la zona
	delegacion	text		delegación a la que corresponde
	ciudad	text		ciudad
contactos_tel	estado	text		estado
	contactoId	int	FK	identificador de persona
Tabla transitiva entre la tabla de persona y teléfono	telefonoId	int	FK	identificador de teléfono
	contratante_tel	contratanteId	int	FK
Tabla transitiva entre la tabla de contratante y la tabla teléfono	telefonoId	int	FK	identificador de teléfono
	gpoTrabajo_tel	grupo_trabajoId	int	FK
Tabla transitiva entre la tabla grupo_trabajo y la tabla telefonos	telefonoId	int	FK	identificador de teléfono
	tipoId	int	FK	identificador de tipo de teléfono
Tabla que permite almacenar los telefonos de las personas	Telefono	text		número telefónico
	tipo_telefonos	tipoId	int	PK
Tabla que permite almacenar los tipos de teléfonos	tipotelefono	Text		tipo de teléfono

Tabla 6. 8 Diccionario de Datos

6.4 Desarrollo del sistema

En este apartado nos vamos a enfocar en construir el sistema con base en todos los puntos anteriores, obteniendo la versión operativa del mismo la cual corresponde a la fase de construcción de la metodología empleada.

Vamos a presentar la construcción del sistema de la siguiente manera: presentación de pantallas y la descripción de las mismas.

6.4.1 Presentación de las pantallas

Teniendo en cuenta que las pantallas son la representación del sistema y que su esquema facilitará su uso, se tomaron en consideración los siguientes lineamientos para el diseño de las mismas:

- Mantener una pantalla sencilla.
- Mantener consistente la presentación de las pantallas.

- Facilitar al usuario la navegación entre pantallas.
- Crear una pantalla atractiva.

6.4.2 Sitio Web de la UAPIEX

El sitio Web fue creado con el propósito de mostrar información a la comunidad en general, acerca de los proyectos que se desarrollan en la Universidad por medio de la *Memoria Tecnológica*, sin embargo con el fin de ofrecer una solución más completa, el sitio maneja un área de *Avisos* y de *Descargas*, sin perder de vista que la función primordial del sitio es contener el SIPIEX.

6.4.2.1 Página principal

La página principal es la entrada al Sitio Web, su finalidad es dar una visión general del contenido del sitio por medio del menú y proporcionar el marco general de las funciones del CISI, la UAPIEX y el SIPIEX (Figura 6.47).



Figura 6. 47 Página principal

6.4.2.2 Entrar al sistema

Su función es reconocer si el usuario esta registrado en el SIPIEX para ingresar al mismo (Figura 6.48).

Centro de Innovación y Servicios de Ingeniería

Unidad de Administración
de Proyectos de Ingresos Extraordinarios

Página Principal **Entrar al Sistema** Memoria Tecnológica Descargas Avisos

Sistema de Información para Proyectos que Generan Ingresos Extraordinarios
"SIPIEX"

Por favor introduce tu nombre de usuario y contraseña:

Nombre de Usuario:

Contraseña:

Figura 6.48 Entrar al sistema

6.4.2.3 Memoria Tecnológica

La pantalla de la Figura 6.49 ingresa a la *Memoria Tecnológica* mostrando como primera instancia una breve explicación sobre la misma.



Figura 6. 49 Memoria Tecnológica

La Figura 6.50 muestra el listado de los proyectos que se encuentran en operación o finalizados.

 Centro de Innovación y Servicios de Ingeniería 						
Unidad de Administración de Proyectos de Ingresos Extraordinarios						
Página Principal	Entrar al Sistema	Memoria Tecnológica	Descargas	Avisos		
Memoria Tecnológica						
 Regresar						
Listado de proyectos desarrollados en la Facultad de Ingeniería de la UNAM						
No	Proyecto	Contratante	Responsable	Líder	División	Vigencia
1	Cursos presenciales para la capacitación técnica	SCT	M. en C. Roy Sanchez Herrera	M. en I. Raquel Quiroz Sandoval	DIE, DICTyG	09/02/2004 al 09/12/2004
2	Determinación de propiedades de rocas fracturadas en el Gruta Bernal	PEMEX	M. en I. Raquel Quiroz Sandoval	M. en C. Roy Sanchez Herrera	COORD. PROY. PEMEX	01/12/2002 al 12/01/2003
3	Proyecto integral de voz y datos de la delegación Cuauhtémoc	IMSS	M. en C. Víctor Manuel Vital Cruz	M. en C. Roy Sanchez Herrera, Dr. Salvador García Méndez	DEP	09/07/2004 al 08/09/2004
4	Levantamiento topográfico y estudio de mecánica de suelos en los polígonos	PEMEX	Dr. Salvador García Méndez	Dr. Salvador García Méndez	DICTyG	05/01/2002 al 12/09/2002
5	Evaluación del plan de estudio de la Universidad Teologica del Estado de México	CFE	M. en C. Roy Sanchez Herrera	Dr. Ivan Ruvalcaba Soto	DICT	10/09/2004 al 04/09/2004
Total de Proyectos: 5						

Figura 6. 50 Memoria Tecnológica listado de proyecto

La Figura 6.51 muestra información más detallada sobre los proyectos.

The screenshot displays the 'Memoria Tecnológica' web application. At the top, there is a header with the logo of the 'Centro de Innovación y Servicios de Ingeniería' and the 'Unidad de Administración de Proyectos de Ingresos Extraordinarios'. Below the header is a navigation menu with links for 'Página Principal', 'Entrar al Sistema', 'Memoria Tecnológica', 'Descargas', and 'Avisos'. The main content area is titled 'Memoria Tecnológica' and includes a 'Regresar' button with a left-pointing arrow. Below this is a section titled 'Detalle del Proyecto' which lists the following information:

- Nombre:** Cursos presenciales para la capacitación técnica
- Objetivo:** Apoyar en la generación de soluciones tecnológicas a los problemas reales de SCT Exploración y Producción, a través de la aplicación de las tecnologías más actualizadas para la exploración, explotación, distribución y suministro de los hidrocarburos
- Contratante(s):** Secretaría de Comunicaciones y Transportes (*Responsable: Arq. Laura Pacheco Solis, Tel: 55 98 63 44 ext. 1542*)
- División(es):** División de Ingeniería Eléctrica, División de Ingeniería Civil, Topográfica y Geodésica
- Responsable(s):** M. en C. Roy Sanchez Herrera (*Tel: 56 84 39 14*)
- Líder(es):** M. en I. Raquel Quiroz Sandoval (*Tel: 55 97 34 76*)
- Vigencia del Proyecto:** 09/02/2004 al 09/12/2004
- Metodología:**
- Entregables:**
 - Preparación de muestras.
 - Porosidad medida con helio a temperatura de laboratorio.
 - Comportamiento de porosidad efectiva de matriz y/o fractura, en función del esfuerzo efectivo, medida con helio a temperatura de laboratorio (25 mediciones).
 - Comportamiento de porosidad efectiva de matriz y/o fractura, en función del esfuerzo efectivo, medida con helio a temperatura de yacimiento (25 mediciones).
 - Compresibilidad de matriz determinada a temperatura de laboratorio.
- Resultados:**
- Publicaciones:**
- Equipo de Trabajo:**

Figura 6. 51 Memoria Tecnológica, datos de un proyecto

6.4.2.4 Descargas

La pantalla de la Figura 6.52 tiene el objetivo de poner al alcance del usuario archivos relacionados con el sitio, referentes a la normatividad manejada en los proyectos, formatos de convenios de los proyectos, aplicaciones referentes a los proyectos, entre otros.



Nombre	Descripción	Descargar
Apertura de proyecto	Programa para generar la apertura de proyecto. Facilita el cálculo de totales en subgrupos, y permite generar el formato de apertura de proyecto de manera automática	apertura.exe
Manual de asesoría	Manual de asesoría de procedimientos para proyectos que generan ingresos extraordinarios	manual.html
Formato de alta preliminar	Formato de alta preliminar de un proyecto	formato alta preliminar.doc
Formato de apertura de proyecto	Formato de la apertura de proyecto. Éste es utilizado cuando no se quiere obtener por medio del programa "Apertura de Proyecto" mencionado arriba	formato apertura.doc
Formatos de convenios	Formatos de convenios generales y de colaboración manejados en la Facultad de Ingeniería de la UNAM	formatos de convenios.zip

Total de Archivos a Descargar: 5

Figura 6. 52 Descargas

6.4.2.5 Avisos

La pantalla de la Figura 6.53 mostrará cualquier tipo de avisos relacionados con la UAPIEX.



Figura 6. 53 Avisos

6.4.3 SIPIEX

El SIPIEX es de acceso restringido a los involucrados en los procesos relacionados con los proyectos que generan ingresos extraordinarios, su función es la gestión de dichos proyectos, lo cual se lleva a cabo por medio de varios módulos, sin embargo, depende de los permisos con los que cuenta cada usuario para ingresar a cada uno de ellos.

Las pantallas que a continuación se muestran son las vistas de un usuario con acceso total como un administrador.

6.4.3.1 Menú principal

El menú principal muestra los módulos de los que consta el sistema (Figura 6.54) los cuales son:

- Proyectos en Promoción
- Proyectos en Operación
- Usuarios
- Permisos
- Responsables o Líderes de Proyecto
- Contratantes
- Divisiones



Figura 6. 54 Menú principal del SIPIEX

6.4.3.2 Proyectos en Promoción

La siguiente pantalla muestra el Resumen de Proyectos en Promoción cuyo objetivo es informar el total de los proyectos y un resumen financiero de los mismos. Es un marco general de los ingresos que se podrían obtener al desarrollar estos proyectos (Figura 6.55).

The screenshot displays the 'Resumen Ejecutivo' (Executive Summary) for 'Proyectos en Promoción' (Projects in Promotion) within the SIPIEX system. The interface features a blue header with the university's name and logo, a navigation menu with options like 'Menú Principal', 'Ayuda', 'Salir', and 'Acerca', and a sidebar with 'Resumen Ejecutivo' and 'Proyectos en Promoción'. The main content area shows a summary table with the following data:

Resumen Ejecutivo	
Total de Proyectos:	2
Total de Divisiones:	3
Monto Total:	\$3,621,918.00
Retención UNAM:	\$724,383.60
Retención FI:	\$543,287.70
Monto Operativo:	\$2,354,246.70

Navigation buttons for 'Regresar', 'Avanzar', and 'Imprimir' are visible on the right side of the summary table.

Figura 6. 55 Resumen ejecutivo de Proyectos en Promoción

La Figura 6.56 muestra los datos de los proyectos con sus principales datos. En esta pantalla se permite *Agregar*, *Editar*, *Enviar a Operación*, *Eliminar* e *Imprimir* un proyecto en promoción.

Centro de Innovación y Servicios de Ingeniería
 Unidad de Administración de Proyectos de Ingresos Extraordinarios

SIPIEX
 Sistema de Información para Proyectos que Generan Ingresos Extraordinarios

Menú Principal Ayuda Salir Acerca

Proyectos en Promoción - Listado de Proyectos

Regresar Agregar Editar Enviar a Operación Eliminar Imprimir

<input type="checkbox"/>	No	Proyecto	Contratante	Responsable	Líder	División	Importe
<input type="checkbox"/>	1	Construcción general para la vivienda, Manuel	SCT, SE		M. en I. Raquel Quiroz Sandoval, Dr. Ivan Ruvalcaba Soto	DEP, DIMEI	\$1,769,844.00
<input type="checkbox"/>	2	Evaluación de seguridad de edificios en el Centro Histórico de la Ciudad de México	IMSS, INE, SE, SSA	M. en I. Raquel Quiroz Sandoval	M. en C. Víctor Manuel Vital Cruz	DICTyG	\$822,564.00

Total de Proyectos: 2

Figura 6. 56 Listado de Proyectos en Promoción

La Figura 6.57 muestra la información de un proyecto desglosada en distintos rubros como son: información técnica, financiera y jurídica.

The screenshot displays the SIPIEX web interface. At the top, it identifies the 'Centro de Innovación y Servicios de Ingeniería' and the 'Unidad de Administración de Proyectos de Ingresos Extraordinarios'. The main header includes navigation links: 'Menú Principal', 'Ayuda', 'Salir', and 'Acerca'. The left sidebar contains 'Resumen Ejecutivo' and 'Proyectos en Promoción'. The main content area is titled 'Proyectos en Promoción - Ver Proyecto' and features three action buttons: 'Regresar', 'Editar', and 'Imprimir'. The data is organized into three sections: 'Información Técnica', 'Información Financiera', and 'Información Jurídica'.

Información Técnica	
Nombre:	Construcción general para la vivienda, Manual
Objetivo:	Actualización manual CEMEX, Mejoramiento de la Vivienda de Autoconstrucción
Duración:	6 meses
Avance:	40%
Contratante(s):	Secretaría de Comunicaciones y Transportes (<i>Responsable: Arq. Laura Pacheco Solis, Tel: 55 98 63 44 ext. 1542</i>), Secretaría de Economía (<i>Responsable: Ing. Omar Gutierrez Bocanegra, Tel: 52 50 48 99</i>)
División(es):	División de Estudios de Posgrado de la Facultad de Ingeniería, División de Ingeniería Mecánica e Industrial
Responsable(s):	Dr. Rodrigo Cervantes Saavedra (<i>Tel: 56 87 99 41</i>)
Líder(es):	M. en I. Raquel Quiroz Sandoval (<i>Tel: 55 97 34 76</i>), Dr. Ivan Ruvalcaba Soto (<i>Tel: 56 11 66 85</i>)
Información Financiera	
Importe:	\$1,769,844.00
Retención UNAM:	\$353,968.80
Retención FI:	\$265,476.60
Monto Operativo:	\$1,150,398.60
Información Jurídica	
Estatus Convenio:	Sin versión de Convenio

Figura 6. 57 Ver Proyecto en Promoción

Las Figuras 6.58 y 6.59 tienen como objetivo agregar un nuevo proyecto en promoción.

The screenshot shows the SIPIEX web application interface. At the top, there is a header with the logo of the Centro de Innovación y Servicios de Ingeniería and the text "Unidad de Administración de Proyectos de Ingresos Extraordinarios". Below the header, there is a navigation menu with "Menú Principal", "Ayuda", "Salir", and "Acerca". The main content area is titled "Proyectos en Promoción - Agregar Proyecto". On the left, there is a sidebar with "Resumen Ejecutivo" and "Proyectos en Promoción". The main form has several sections: "Datos Generales del Proyecto" with fields for "Nombre:", "Objetivo:", "Duración:", "Importe:", "Estatus Convenio:", and "Avance:" (set to 0%). Below this is the "Contratantes" section, which includes a list of "Contratantes a Seleccionar" (SCT, SE, INE, SSA, IMSS) and a "Contratantes Seleccionados" list. There are "Regresar" and "Agregar" buttons, and navigation arrows between the contractor lists.

Figura 6. 58 Agregar un Proyecto en Promoción

The screenshot shows a web-based interface for adding a project in promotion. It is organized into three main sections, each with a blue header bar:

- Divisiones:** Contains a list of divisions to select (DIE, DEP, DIMEI, DCSyH, DICTyG) and a list of selected divisions. An 'Agregar' button is in the top right.
- Responsables de Proyecto:** Contains a list of project managers to select (M. en I. Raquel Quiroz Sandoval, M. en C. Roy Sanchez Herrera, Dr. Salvador Garcia Méndez, Dr. Ivan Ruvalcaba Soto, M. en C. Víctor Manuel Vital Cruz) and a list of selected managers. An 'Agregar' button is in the top right.
- Líderes de Proyecto:** Contains a list of project leaders to select (M. en I. Raquel Quiroz Sandoval, M. en C. Roy Sanchez Herrera, Dr. Salvador Garcia Méndez, Dr. Ivan Ruvalcaba Soto, M. en C. Víctor Manuel Vital Cruz) and a list of selected leaders. An 'Agregar' button is in the top right.

Below the sections, a red asterisk note states: *** Los campos en rojo son requeridos**. At the bottom of the interface are three buttons: 'Guardar', 'Deshacer', and 'Cancelar'.

Figura 6. 59 Agregar un Proyecto en Promoción

Las Figuras 6.60 y 6.61 corresponden a las pantallas para editar un proyecto en promoción.

The screenshot shows the 'Proyectos en Promoción - Editar Proyecto' screen. The header includes the logo of the 'Centro de Innovación y Servicios de Ingeniería' and the 'Unidad de Administración de Proyectos de Ingresos Extraordinarios'. The navigation menu contains 'Menú Principal', 'Ayuda', 'Salir', and 'Acerca'. The left sidebar has 'Resumen Ejecutivo' and 'Proyectos en Promoción'. The main content area is divided into two sections: 'Datos Generales del Proyecto' and 'Contratantes'. The 'Datos Generales del Proyecto' section contains the following fields:

Nombre:	Manual de autoconstrucción y mejoramiento de la vivienda
Objetivo:	Actualización manual CEMEX, Mejoramiento de la Vivienda de Autoconstrucción
Duración:	6 meses
Importe:	2999312
Estatus Convenio:	Sin versión de convenio
Avance:	40%

The 'Contratantes' section features an 'Agregar' button and two columns: 'Contratantes a Seleccionar' and 'Contratantes Seleccionados'. The 'Contratantes a Seleccionar' column contains a list box with the following items: INE, SSA, IMSS, STPS, and PEMEX. The 'Contratantes Seleccionados' column contains a list box with the following items: SCT and SE. There are also '<<' and '>>' buttons between the two columns.

Figura 6. 60 Editar un Proyecto en Promoción

Divisiones

 **Agregar**

Divisiones a Seleccionar

- DIE
- DCSyH
- DICTyG
- DCB
- DICT

<< >>

Divisiones Seleccionadas

- DEP
- DIMEI

Responsables de Proyecto

 **Agregar**

Responsables a Seleccionar

- M. en I. Raquel Quiroz Sandoval
- M. en C. Roy Sanchez Herrera
- Dr. Salvador García Méndez
- Dr. Ivan Ruvalcaba Soto
- M. en C. Víctor Manuel Vital Cruz

<< >>

Responsables Seleccionados

- Dr. Rodrigo Cervantes Saavedra

Líderes de Proyecto

 **Agregar**

Líderes a Seleccionar

- M. en C. Roy Sanchez Herrera
- Dr. Salvador García Méndez
- M. en C. Víctor Manuel Vital Cruz
- Ing. Margarita Rueda Alfonso
- Ing. Sergio Oswaldo Ríos

<< >>

Líderes Seleccionados

- M. en I. Raquel Quiroz Sandoval
- Dr. Ivan Ruvalcaba Soto

*** Los campos en rojo son requeridos**

Guardar Deshacer Cancelar

Figura 6. 61 Editar un Proyecto en Promoción

6.4.3.3 Proyectos en Operación

La primera página de este módulo (Figura 6.62) muestra el Resumen de Proyectos en Operación cuyo objetivo es informar el total de los proyectos que existen registrados en esta etapa así como un resumen financiero de los mismos. Esta pantalla es de vital importancia, ya que indica los ingresos generados por los proyectos.

The screenshot displays the SIPIEX system interface. At the top, it identifies the 'Centro de Innovación y Servicios de Ingeniería' and the 'Unidad de Administración de Proyectos de Ingresos Extraordinarios'. The main content area is titled 'Proyectos en Operación - Resumen Ejecutivo'. A summary table provides the following data:

Resumen Ejecutivo	
Total de Proyectos:	3
Total de Divisiones:	4
Monto Total:	\$3,209,692.00
Ingreso Actual:	\$2,000.00
Retención UNAM:	\$400.00
Retención FI:	\$300.00
Monto Operativo:	\$1,300.00
Ejercido:	\$400.00
Saldo Actual:	\$900.00

Navigation options include 'Regresar', 'Avanzar', and 'Imprimir'.

Figura 6. 62 Resumen Ejecutivo de Proyectos en Operación

La Figura 6.63 muestra el listado de los proyectos en operación con sus principales datos. En esta pantalla se permite *Editar*, *Concluir*, *Eliminar* e *Imprimir* un proyecto en operación.

Centro de Innovación y Servicios de Ingeniería

Unidad de Administración
de Proyectos de Ingresos Extraordinarios

SIPIEX
Sistema de Información para Proyectos
que Generan Ingresos Extraordinarios

Menú Principal Ayuda Salir Acerca

Proyectos en Operación - Listado de Proyectos

Regresar
 Editar
 Concluir Proyecto
 Eliminar
 Imprimir

<input type="checkbox"/>	No	Proyecto	Contratante	Responsable	Líder	División	Vigencia	Importe
<input type="checkbox"/>	1	Cursos presenciales para la capacitación técnica	SCT	M. en C. Roy Sanchez Herrera	M. en I. Raquel Quiroz Sandoval	DIE, DICTyG	09/02/2004 al 09/12/2004	\$2,000.00
<input type="checkbox"/>	2	Determinación de propiedades de rocas fracturadas en el Gruta Bernal	PEMEX	M. en I. Raquel Quiroz Sandoval	M. en C. Roy Sanchez Herrera	COORD. PROY. PEMEX	01/12/2002 al 12/01/2003	\$2,287,692.00
<input type="checkbox"/>	3	Proyecto integral de voz y datos de la delegación Cuauhtémoc	IMSS	M. en C. Víctor Manuel Vital Cruz	M. en C. Roy Sanchez Herrera, Dr. Salvador García Méndez	DEP	09/07/2004 al 08/09/2004	\$920,000.00

Total de Proyectos: 3

Figura 6. 63 Listado de Proyectos en Operación

La Figura 6.64 muestra la información de un proyecto desglosada en información técnica, financiera y jurídica.

Centro de Innovación y Servicios de Ingeniería

Unidad de Administración de Proyectos de Ingresos Extraordinarios

SIPIEX
Sistema de Información para Proyectos que Generan Ingresos Extraordinarios

[Menú Principal](#) [Ayuda](#) [Salir](#) [Acerca](#)

- ▶ Resumen Ejecutivo
- ▶ **Proyectos en Operación**

Proyectos en Operación - Ver Proyecto

← Regresar
📄 Apertura
📡 Convenio
✍ Editar
🖨 Imprimir

Información Técnica

Nombre: Cursos presenciales para la capacitación técnica

Objetivo: Apoyar en la generación de soluciones tecnológicas a los problemas reales de SCT Exploración y Producción, a través de la aplicación de las tecnologías más actualizadas para la exploración, explotación, distribución y suministro de los hidrocarburos

Avance: 20%

Contratante(s): Secretaría de Comunicaciones y Transportes (*Responsable: Arq. Laura Pacheco Solis, Tel: 55 98 63 44 ext. 1542*)

División(es): División de Ingeniería Eléctrica, División de Ingeniería Civil, Topográfica y Geodésica

Responsable(s): M. en C. Roy Sanchez Herrera (*Tel: 56 84 39 14*)

Líder(es): M. en I. Raquel Quiroz Sandoval (*Tel: 55 97 34 76*)

Información Financiera

Importe: \$2,000.00

Resumen Financiero:		Resumen del Ejercicio:	
Ingreso:	\$2,000.00	Grupo100:	\$100.00
Retención UNAM:	\$400.00	Grupo200:	\$200.00
Fondo FI:	\$300.00	Grupo300:	\$10.00
Monto Operativo:	\$1,300.00	Grupo400:	\$20.00
Ejercido:	\$400.00	Grupo500:	\$30.00
Saldo Actual:	\$900.00	Grupo600:	\$40.00

Información Jurídica

Estatus Convenio: Convenio Firmado

Vigencia del Proyecto: 09/02/2004 al 09/12/2004

Número de Convenio: 9766-721-9-XI-00

Figura 6. 64 Ver Proyectos en Operación

167

El objetivo de las Figuras 6.65 y 6.66 es editar un proyecto en operación.

Centro de Innovación y Servicios de Ingeniería
Unidad de Administración de Proyectos de Ingresos Extraordinarios

SIPIEX
 Sistema de Información para Proyectos que Generan Ingresos Extraordinarios

Menú Principal Ayuda Salir Acerca

Proyectos en Operación - Editar Proyecto Regresar

Información Técnica

Nombre: Cursos presenciales para la capacitación técnica

Objetivo: Apoyar en la generación de soluciones tecnológicas a los problemas reales de SCT Exploración y Producción, a través de la aplicación de las tecnologías más actualizadas para la exploración, explotación, distribución y suministro de los hidrocarburos

Avance: 20%

Contratantes Agregar

Contratantes a Seleccionar: CFE, IMSS, INE, PEMEX, SE

Contratantes Seleccionados: SCT

Divisiones Agregar

Divisiones a Seleccionar: DEP, DIMEI, DCSyH, DCB, DICT

Divisiones Seleccionadas: DIE, DICTyG

Responsables de Proyecto Agregar

Responsables a Seleccionar: M. en I. Raquel Quiroz Sandoval, Dr. Salvador García Méndez, Dr. Ivan Ruvalcaba Soto, M. en C. Víctor Manuel Vital Cruz, Ing. Margarita Rueda Alfonso

Responsables Seleccionados: M. en C. Roy Sanchez Herrera

Figura 6. 65 Editar un Proyecto en Operación

Líderes de Proyecto



Agregar

Líderes a Seleccionar

M. en C. Roy Sanchez Herrera

Dr. Salvador García Méndez

Dr. Ivan Ruvalcaba Soto

M. en C. Víctor Manuel Vital Cruz

Ing. Margarita Rueda Alfonso

Líderes Seleccionados

M. en I. Raquel Quiroz Sandoval

Información Financiera

Importe:	<input type="text" value="\$ 2000"/>	Ingresos:	<input type="text" value="\$ 2000"/>
Grupo100:	<input type="text" value="\$ 100"/>	Grupo200:	<input type="text" value="\$ 200"/>
Grupo300:	<input type="text" value="\$ 10"/>	Grupo400:	<input type="text" value="\$ 20"/>
Grupo500:	<input type="text" value="\$ 30"/>	Grupo600:	<input type="text" value="\$ 40"/>

Información Jurídica

Estatus Convenio:

Número de Convenio:

Fecha de Inicio del Convenio:  (dd/MM/YYYY)

Fecha de Término del Convenio:  (dd/MM/YYYY)

* Los campos en rojo son requeridos

Figura 6. 66 Editar un Proyecto en Promoción

6.4.3.4 Usuarios

En esta opción el administrador del sistema registra a los usuarios del sistema. La Figura 6.67 muestra la primera pantalla de este módulo cuyo objetivo es listar a los usuarios del sistema y además permite *Agregar*, *Editar*, *Eliminar* e *Imprimir* información de un usuario.

The screenshot displays the 'Usuarios - Listado de Usuarios' interface. At the top, it identifies the system as 'SIPIEX Sistema de Información para Proyectos que Generan Ingresos Extraordinarios' under the 'Centro de Innovación y Servicios de Ingeniería' and 'Unidad de Administración de Proyectos de Ingresos Extraordinarios'. A navigation bar includes 'Menú Principal', 'Ayuda', 'Salir', and 'Acerca'. A sidebar on the left shows 'Usuarios' as the active menu item. The main content area features a table with the following data:

<input type="checkbox"/>	No	Usuario	Perfil	Nombre	Apellido
<input type="checkbox"/>	1	marcela	DBA	Marcela	Gamiño
<input type="checkbox"/>	2	ubaldo	DBA	Ubaldo	Acosta
<input type="checkbox"/>	3	vicky	LIDER DE PROYECTO	Victoria	Llanes

Below the table, it states 'Total de Usuarios: 3'. Above the table, there are five action buttons: 'Regresar' (back arrow), 'Agregar' (plus icon), 'Editar' (pencil icon), 'Eliminar' (trash icon with red X), and 'Imprimir' (printer icon).

Figura 6. 67 Listado de Usuarios

La siguiente Figura 6.68 corresponde a la pantalla de *Agregar* un usuario, cuyo propósito es registrar un nuevo usuario con sus datos personales.

The screenshot shows a web application interface for adding a user. At the top, there is a header with the logo of the 'Centro de Innovación y Servicios de Ingeniería' and the text 'Unidad de Administración de Proyectos de Ingresos Extraordinarios'. Below the header is a navigation bar with 'Menú Principal', 'Ayuda', 'Salir', and 'Acerca'. The main content area is titled 'Usuarios - Agregar Usuario' and contains a form titled 'Datos Generales del Usuario'. The form has the following fields:

- Usuario:** Text input field.
- Password:** Text input field.
- Reescribe Password:** Text input field.
- Perfil:** Dropdown menu with 'OTRO' selected.
- Nombre:** Text input field.
- Apellido:** Text input field.

A red asterisk and the text '* Los campos en rojo son requeridos' are displayed below the form fields. At the bottom of the form, there are three buttons: 'Guardar', 'Deshacer', and 'Cancelar'. A 'Regresar' button with a left-pointing arrow is located in the top right corner of the form area.

Figura 6. 68 Agregar Usuarios

La Figura 6.69 corresponde a la opción de *Editar* y tiene el objetivo de modificar los datos de un usuario.

The screenshot shows the 'Editar Usuario' (Edit User) form within the SIPIEX system. The header includes the logo of the 'Centro de Innovación y Servicios de Ingeniería' and the 'Unidad de Administración de Proyectos de Ingresos Extraordinarios'. The system name 'SIPIEX' is displayed as 'Sistema de Información para Proyectos que Generan Ingresos Extraordinarios'. A navigation menu contains 'Menú Principal', 'Ayuda', 'Salir', and 'Acerca'. The left sidebar shows 'Usuarios' as the active menu item. The main form area is titled 'Usuarios - Editar Usuario' and includes a 'Regresar' button with a left-pointing arrow. Below the title is a section header 'Datos Generales del Usuario'. The form fields are as follows:

Usuario:	<input type="text" value="vicky"/>
Password:	<input type="password" value="•••"/>
Reescribe Password:	<input type="password" value="•••"/>
Perfil:	<input type="text" value="LIDER DE PROYECTO"/>
Nombre:	<input type="text" value="Victoria"/>
Apellido:	<input type="text" value="Llanes"/>

* Los campos en rojo son requeridos

Buttons:

Figura 6. 69 Editar Usuario

6.4.3.5 Responsables o Líderes de Proyecto

Este módulo tiene como finalidad registrar los datos personales de los diferentes responsables y/o líderes de proyectos.

La Figura 6.70 muestra la primera página de este módulo, donde es posible *Agregar*, *Editar* o *Eliminar* e *Imprimir* información de algún responsable o líder de proyecto.

Centro de Innovación y Servicios de Ingeniería
Unidad de Administración de Proyectos de Ingresos Extraordinarios

SIPIEX
Sistema de Información para Proyectos que Generan Ingresos Extraordinarios

Menú Principal Ayuda Salir Acerca

Responsables o Líderes de Proyecto

Responsables o Líderes de Proyecto - Listado

Regresar Agregar Editar Eliminar Imprimir

<input type="checkbox"/>	No	Nombre	Ubicación	Teléfono	Email
<input type="checkbox"/>	1	M. en I. Raquel Quiroz Sandoval	Edificio A Posgrado 1er. Piso	55 97 34 76	raqua@yahoo.com
<input type="checkbox"/>	2	M. en C. Roy Sanchez Herrera	Av. de los 100 metros 246. Despacho 304. Col. San Martin	56 84 39 14	roy@hotmail.com
<input type="checkbox"/>	3	Dr. Salvador Garcia Méndez	Ricardo Flores Magon No.54, 9º Piso, C.P.06300 México , D.F.	56 85 66 11	garciam@hotmail.com
<input type="checkbox"/>	4	Dr. Ivan Ruvalcaba Soto	Edificio A Posgrado 3er. Piso	56 11 66 85	ruvalcabas@hotmail.com
<input type="checkbox"/>	5	M. en C. Víctor Manuel Vital Cruz	Edificio A Posgrado 1er. Piso	56 97 89 44	vitalc@hotmail.com
<input type="checkbox"/>	6	Ing. Margarita Rueda Alfonso	Edificio B Posgrado 2do. Piso	55 34 97 76	ruedaa@servidor.unam.mx
<input type="checkbox"/>	7	Ing. Sergio Oswaldo Rios	División de Ingeniería Eléctrica 3er. Piso	58 98 44 63	oswaldor@servidor.unam.mx

Total de Responsables o Líderes de Proyecto: 7

Figura 6. 70 Listado de los Responsables o Líderes de proyecto

En la Figura 6.71 es posible registrar un nuevo responsable o líder de proyecto con sus principales datos personales.

Centro de Innovación y Servicios de Ingeniería
Unidad de Administración de Proyectos de Ingresos Extraordinarios

SIPIEX
Sistema de Información para Proyectos que Generan Ingresos Extraordinarios

Menú Principal Ayuda Salir Acerca

Responsables o Líderes de Proyecto - Agregar

Responsables o Líderes de Proyecto

Regresar

Datos Generales del Responsable o Líder de Proyecto

Nombre:

Ubicación:

Teléfono:

Celular:

Email:

* Los campos en rojo son requeridos

Guardar Deshacer Cancelar

Figura 6. 71 Agregar un Responsable o Líder de proyecto

La Figura 6.72 muestra la opción de *Editar* los datos de un responsable o líder de proyecto.

Centro de Innovación y Servicios de Ingeniería
Unidad de Administración de Proyectos de Ingresos Extraordinarios

SIPIEX
Sistema de Información para Proyectos que Generan Ingresos Extraordinarios

Menú Principal Ayuda Salir Acerca

Responsables o Líderes de Proyecto - Editar

Responsables o Líderes de Proyecto

Regresar

Datos Generales del Responsable o Líder de Proyecto

Nombre: M. en I. Raquel Quiroz Sandoval

Ubicación: Edificio A Posgrado 1er. Piso

Teléfono: 55 97 34 76

Celular: 04455 10034399

Email: raqua@yahoo.com

* Los campos en rojo son requeridos

Guardar Deshacer Cancelar

Figura 6. 72 Editar un Responsable o Líder de proyecto

6.4.3.6 Contratantes

El módulo de contratantes tiene la finalidad de registrar las entidades contratantes. La primera pantalla se muestra en la Figura 6.73, en ella es posible *Agregar*, *Editar*, *Eliminar* e *Imprimir* información de algún contratante.

Centro de Innovación y Servicios de Ingeniería
Unidad de Administración de Proyectos de Ingresos Extraordinarios

SIPIEX
Sistema de Información para Proyectos que Generan Ingresos Extraordinarios

Menú Principal Ayuda Salir Acerca

Contratantes - Listado

Regresar Agregar Editar Eliminar Imprimir

<input type="checkbox"/>	No	Nombre	Abreviatura	Contacto	Ubicación	Teléfono
<input type="checkbox"/>	1	Comisión Federal de Electricidad	CFE	M. C. Fernando del Prado Monroy	Av. Río Churubusco No. 1005. Col. Santo Tomas . Delegación Iztacalco	56 66 85 12
<input type="checkbox"/>	2	Instituto Mexicano del Seguro Social	IMSS	Ing. Gerardo Gonzalez de Alva	Avenida Insurgentes norte, Colonia Grijalva, C.P. 11850, México, Distrito Federal	54 78 96 78
<input type="checkbox"/>	3	Instituto Nacional de Ecología	INE	Lic. Pablo Morales Gil	Tacuba No. 17, Col. Centro, C.P. 04009, Ciudad de México, Distrito Federal.	58 96 47 12 y 56 84 12 34
<input type="checkbox"/>	4	Petróleos Mexicanos	PEMEX	Ing. Gloria Martinez Flores	Sady Carnot No 10 Col. San Rafael, 01780., México D.F	58 49 22 65
<input type="checkbox"/>	5	Secretaría de Comunicaciones y Transportes	SCT	Arq. Laura Pacheco Solis	Londres No. 102. Piso 7°. Col. Juárez, Delegación Cuauhtemoc. C.P. 08600	55 98 63 44 ext. 1542
<input type="checkbox"/>	6	Secretaría de Economía	SE	Ing. Omar Gutierrez Bocanegra	Octavio Paz No. 18 Tercer Piso, Col Coyoacan, México DF	52 50 48 99
<input type="checkbox"/>	7	Secretaría de Salud	SSA	Lic. Marlen Villa Sanchez	Dirección Comercial, Avenida Constitución de 1945 No. 4 Código Postal 64000, Monterrey, Nuevo León. México.	59 87 45 63
<input type="checkbox"/>	8	Secretaría del Trabajo y Previsión Social	STPS	Lic. Sofía de la Luz Soto	Plaza de la constitución No.1, Colonia Tlalpan Centro. C.P. 14000, Delegación Tlalpan	56 55 60 72

Total de Contratantes: 8

Figura 6. 73 Listado de Entidades Contratantes

En la Figura 6.74 muestra la página que permite *Agregar* los datos principales de una entidad contratante.

The screenshot displays the 'Contratantes - Agregar' page in the SIPIEX system. The header includes the logo of the Centro de Innovación y Servicios de Ingeniería and the Unidad de Administración de Proyectos de Ingresos Extraordinarios. The main content area features a sidebar with 'Contratantes' and a central form titled 'Datos Generales del Contratante'. The form contains the following fields:

- Nombre:** (Required field)
- Abreviatura:** (Required field)
- Contacto:** (Required field)
- Ubicación:** (Required field)
- Teléfono:** (Required field)
- Celular:** (Required field)
- Email:** (Required field)

A red asterisk below the fields states: *** Los campos en rojo son requeridos**. At the bottom of the form, there are three buttons: 'Guardar', 'Deshacer', and 'Cancelar', and a 'Regresar' button with a left-pointing arrow.

Figura 6. 74 Agregar información de la entidad Contratante

La siguiente pantalla tiene el objetivo de modificar los datos de alguna entidad contratante (Figura 6.75).

Centro de Innovación y Servicios de Ingeniería
 Unidad de Administración de Proyectos de Ingresos Extraordinarios

SIPIEX
 Sistema de Información para Proyectos que Generan Ingresos Extraordinarios

[Menú Principal](#) [Ayuda](#) [Salir](#) [Acerca](#)

Contratantes - Editar [Regresar](#)

Datos Generales del Contratante

Nombre: Comisión Federal de Electricidad

Abreviatura: CFE

Contacto: M. C. Fernando del Prado Monroy

Ubicación: Av. Río Churubusco No. 1005. Col. Santo Tomas . Delegación Iztacalco

Teléfono: 56 66 85 12

Celular: 04455 86 45 98 14

Email: prado@cfe.mx

*** Los campos en rojo son requeridos**

Figura 6. 75 Editar una entidad Contratante

6.4.3.7 Divisiones

El módulo de divisiones es el encargado de registrar a las diferentes divisiones o campos disciplinarios que intervienen en el proceso de los proyectos que generan ingresos extraordinarios.

La Figura 6.76 muestra la primera página que aparece al elegir dicho módulo, en ella es posible *Agregar, Editar, Eliminar* e *Imprimir* los datos de las divisiones.

Centro de Innovación y Servicios de Ingeniería
 Unidad de Administración de Proyectos de Ingresos Extraordinarios

SIPIEX
 Sistema de Información para Proyectos que Generan Ingresos Extraordinarios

Menú Principal Ayuda Salir Acerca

Divisiones - Listado

Regresar Agregar Editar Eliminar Imprimir

<input type="checkbox"/>	No	Nombre	Abreviatura
<input type="checkbox"/>	1	Coordinación de Planeación y Desarrollo	CPYD
<input type="checkbox"/>	2	Coordinación de Proyectos Pemex de la Facultad de Ingeniería	COORD. PROY. PEMEX
<input type="checkbox"/>	3	Dirección General de Servicios de Cómputo Académico	DGSCA
<input type="checkbox"/>	4	División Ciencias Básicas	DCB
<input type="checkbox"/>	5	División de Ciencias Sociales y Humanidades	DCSyH
<input type="checkbox"/>	6	División de Educación Continua Facultad de Ingeniería	DEC
<input type="checkbox"/>	7	División de Estudios de Posgrado de la Facultad de Ingeniería	DEP
<input type="checkbox"/>	8	División de Ingeniería Civil, Topográfica y Geodésica	DICTyG
<input type="checkbox"/>	9	División de Ingeniería Eléctrica	DIE
<input type="checkbox"/>	10	División de Ingeniería Mecánica e Industrial	DIMEI
<input type="checkbox"/>	11	División de Ingeniería en Ciencias de la Tierra	DICT
<input type="checkbox"/>	12	Instituto de Ingeniería	INST. ING.

Total de Divisiones: 12

Figura 6. 76 Listado de las Divisiones o Campos disciplinarios

La Figura 6.77 es la pantalla de la opción *Agregar* y su finalidad es registrar una nueva división o campo disciplinario.

The screenshot shows the 'Agregar' (Add) screen for Divisions. The header includes the logo of the 'Centro de Innovación y Servicios de Ingeniería' and the 'Unidad de Administración de Proyectos de Ingresos Extraordinarios'. The system name 'SIPIEX' is displayed as 'Sistema de Información para Proyectos que Generan Ingresos Extraordinarios'. A navigation menu contains 'Menú Principal', 'Ayuda', 'Salir', and 'Acerca'. The main title is 'Divisiones - Agregar'. A 'Regresar' button is located in the top right. The form section, titled 'Datos Generales de la División', contains two input fields: 'Nombre:' and 'Abreviatura:'. A red asterisk note states '* Los campos en rojo son requeridos'. At the bottom, there are three buttons: 'Guardar', 'Deshacer', and 'Cancelar'.

Figura 6. 77 Agregar una nueva División o Campo disciplinario

La pantalla de la Figura 6.78 tiene el propósito de *Editar* una división existente en el sistema.

The screenshot shows the 'Editar' (Edit) screen for Divisions. The header and navigation menu are identical to the 'Agregar' screen. The main title is 'Divisiones - Editar'. A 'Regresar' button is located in the top right. The form section, titled 'Datos Generales de la División', contains two input fields: 'Nombre:' with the value 'Coordinación de Planeación y Desarrollo' and 'Abreviatura:' with the value 'CPYD'. A red asterisk note states '* Los campos en rojo son requeridos'. At the bottom, there are three buttons: 'Guardar', 'Deshacer', and 'Cancelar'.

Figura 6. 78 Editar una División o Campo disciplinario

6.4.3.8 Ayuda.

En esta opción de la barra del menú se muestra un manual para el usuario, con la finalidad de guiarlo en el uso correcto del sistema.

6.4.3.9 Salir

Esta opción permite cerrar la sesión en el SIPIEX regresando a la pantalla de *Entrar al Sistema*.

6.4.3.10 Acerca

Muestra los datos generales del sistema, Figura 6.79.

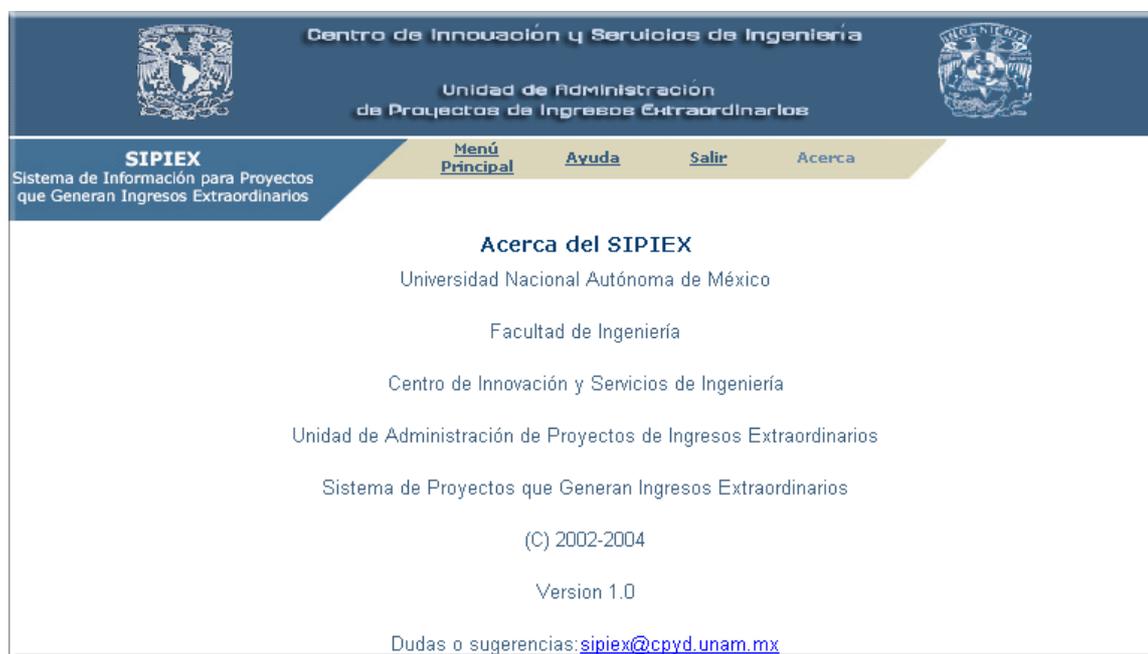


Figura 6. 79 Acerca

6.5 Pruebas y resultados en el sistema

En este rubro vamos a enfocarnos a asegurar que los requerimientos de usuario se cumplan de manera completa, correcta y consistente, además de encontrar fallas funcionales, lógicas o de diseño por medio de la realización de pruebas al sistema, lo cual corresponde a la fase de transición de la metodología empleada.

Para dicha fase, el sistema ya ha alcanzado suficiente capacidad operativa y confianza para ejecutarse en el ambiente del usuario, sin embargo esto no quiere decir que el sistema haya quedado libre de errores; es por esta razón que es de vital importancia delinear adecuadamente el sistema final.

En el sistema utilizamos distintos tipos de pruebas, como son de:

- **Integración:** Son pruebas aplicadas a un grupo de objetos para asegurarnos que los mensajes sean enviados adecuadamente entre ellos. El objetivo es tomar los objetos probados en unidad y construir una estructura de aplicación que esté de acuerdo con lo que dicta nuestro diseño.
- **Caja blanca:** Son pruebas basadas en el conocimiento sobre la lógica y estructura internas. Permiten entender la estructura interna del sistema.
- **Caja negra:** Son pruebas funcionales basadas en los requerimientos del usuario sin conocimiento sobre cómo fue construido un método o un componente.
- **Carga:** Son pruebas realizadas para verificar el comportamiento adecuado y eficiente del sistema bajo condiciones de concurrencia y de alta actividad por parte de los usuarios.
- **Aceptación de usuario:** Son las pruebas finales ejecutadas por el usuario para asegurar que el sistema satisface sus necesidades, de igual manera pretende descubrir errores cometidos en la implantación de los requerimientos.

Debido a que las pruebas se hicieron basándonos en las trayectorias de los casos de usos, principales y alternativas, descritos en las Tablas 6.4 y 6.5, no volveremos a describir cada trayectoria.

Todos los tipos de pruebas se ejecutaron exitosamente. Cabe mencionar que de este tipo de pruebas obtuvimos una retroalimentación y se pudieron corregir errores de diseño y funcionales.

Para las pruebas de carga se utilizó una herramienta llamada Web Application Stress Tool ([HTTP://webtool.rte.microsoft.com](http://webtool.rte.microsoft.com)). Dicha herramienta nos permite simular un número específico de clientes accediendo al sistema de manera concurrente y visitando distintas páginas del sistema.

Cabe mencionar que se debe hacer un plan de trabajo para la aplicación de las pruebas hechas al sistema, el cual se debe revisar exhaustivamente antes de liberarlo. Aunque esto queda fuera del alcance de esta tesis, es importante mencionarlo, ya que para futuras versiones es indispensable seguir paso a paso las pruebas hechas al sistema.

Conclusiones

Al cubrir los requerimientos solicitados por la UAPIEX, podemos concluir que el SIPIEX nos permite registrar proyectos, conocer información relevante, oportuna y confiable, obteniéndose un control eficiente de los mismos. La información puede ser consultada vía Internet por medio del sitio Web de la Unidad, esto hace posible que la información sea accesible a todo usuario del sistema.

El SIPIEX permite dar un seguimiento adecuado a los proyectos y por medio de la información técnica, financiera y jurídica, las instancias se mantienen informadas.

La seguridad fue un aspecto de vital importancia al acceder a la información que proporciona el sistema. Fue por ello que se definieron distintos roles para los usuarios, obteniéndose un óptimo control de la información que se está mostrando.

Por otro lado el sitio Web se diseñó para que la información presentada sea clara, las pantallas consistentes, la navegación del usuario sencilla y el entorno general del sitio sea atractivo.

La metodología para el desarrollo de software en la que nos basamos fue el Proceso Unificado. Éste nos proporcionó un conjunto de procedimientos y técnicas útiles para la captura de requerimientos, el ritmo de trabajo, la administración de las actividades, la reducción de los riesgos y en general nos auxilió a tener una visión más amplia del proyecto.

Además también utilizamos UML como la metodología para especificar, visualizar, construir y documentar nuestro sistema. El Proceso Unificado especifica en uno de sus puntos el uso de UML y la utilización de ambas herramientas nos dio un buen resultado, sobre todo para la administración del proyecto y la documentación.

El software libre fue otra de las tecnologías que elegimos, principalmente por que su uso no tiene costo, sin embargo profundizando en sus ventajas encontramos que este tipo de software es eficiente y robusto, como es el caso de Java, Tomcat y PostgreSQL, los cuales cubrieron ampliamente nuestras necesidades, evitando utilizar software propietario con las desventajas que esto conlleva, principalmente el costo de licencias.

Hoy en día una interfaz Web es muy común, pero cada vez es más común usuarios con necesidades de aplicaciones inalámbricas. Un modelo como el propuesto hace los cambios, hacia nuevas interfaces para el usuario, mucho más sencillas.

No debemos olvidar que lo más importante es entender los requerimientos del cliente, ya que sin importar las tecnologías que utilicemos para desarrollar un proyecto, el verdadero reto es poder ofrecerle una solución que cubra sus necesidades.

ANEXO A

Formatos utilizados en la UAPIEX

Este anexo detalla algunos formatos manejados en la UAPIEX y en general en la Facultad de Ingeniería para la administración de los proyectos que generan ingresos extraordinarios.

El formato de Alta Preliminar se utiliza para poder registrar al proyecto en la etapa de promoción y el formato de la Apertura de Proyecto, el cual es un documento oficial manejado en la Facultad de Ingeniería, se utiliza para poder registrar a un proyecto en estado de operación.

La Apertura de Proyecto cubre los siguientes tópicos:

- Registro de variables generales del proyecto, tales como: nombre, vigencia, división o campo disciplinario al que pertenece, patrocinador o entidad contratante, responsable y líder de proyecto, entre otras.
- Cálculo del presupuesto general del proyecto, el cual involucra el ejercicio o gasto del mismo, subdividido en los siguientes grupos:
 - Grupo 100.- Remuneraciones Personales.
 - Grupo 200.- Servicios.
 - Grupo 300.- Becas, Prestaciones y Estímulos.
 - Grupo 400.- Artículos y Materiales de Consumo.
 - Grupo 500.- Mobiliario y Equipo.
 - Grupo 600.- Inmuebles y Construcciones.
 - Imprevistos.
- Firma por parte del Líder de Proyecto, Responsable de Proyecto y por el Titular de la Dependencia.

Un Líder de proyecto puede justificar el presupuesto de un proyecto empleando cada uno de los subgrupos descritos a continuación:

Grupo 100. Remuneraciones personales

- SubGrupo 154. Tiempo extraordinario al personal administrativo de base.
- SubGrupo 182a. Honorarios por servicios profesionales para apoyo a académicos.
- SubGrupo 182b. Honorarios por servicios profesionales para apoyo a externos.
- SubGrupo 184. Honorarios a conferencistas.

Grupo 200. Servicios

- SubGrupo 211. Viáticos.
- SubGrupo 212. Pasajes aéreos.
- SubGrupo 213. Gastos de alumnos en prácticas escolares.
- SubGrupo 214. Gastos de intercambio.
- SubGrupo 215. Gastos de trabajo de campo.
- SubGrupo 218. Otros pasajes.

- SubGrupo 222. Ediciones.
- SubGrupo 223. Encuadernaciones.
- SubGrupo 226. Impresiones.
- SubGrupo 231. Servicios externos.
- SubGrupo 232. Servicios de mantenimiento por contrato para edificios e instalaciones.
- SubGrupo 256. Cuotas de afiliación e inscripción.

Grupo 300. Becas, Prestaciones y Estímulos

- SubGrupo 311. Becas para estudios o investigaciones de mexicanos en el país.
- SubGrupo 312. Becas para estudios o investigaciones de mexicanos en el extranjero.
- SubGrupo 313. Becas para estudios o investigaciones de extranjeros en el país.
- SubGrupo 322. Cuota institucional de seguros para el personal académico y administrativo.
- SubGrupo 322 A. Cuota institucional de seguros para alumnos.

Grupo 400. Artículos y Materiales de Consumo

- SubGrupo 411. Artículos, materiales y útiles diversos.
- SubGrupo 413. Combustibles y lubricantes.
- SubGrupo 415. Herramientas, artículos y materiales para proyectos de investigación.
- SubGrupo 420. Material informativo.
- SubGrupo 421. Documentos y servicios de información.

Grupo 500. Mobiliario y Equipo

- SubGrupo 511. Mobiliario.
- SubGrupo 512. Equipo e instrumental.
- SubGrupo 514. Equipo de cómputo.
- SubGrupo 517. Equipo diverso para proyectos de investigación.
- SubGrupo 521. Libros.
- SubGrupo 523. Revistas técnicas y científicas.
- SubGrupo 599 A. Renta de equipo.

Grupo 600. Inmuebles y Construcciones

- SubGrupo 622. Rehabilitación de inmuebles, equipo e instalaciones.
- SubGrupo 623. Remodelación de inmuebles e instalaciones.

Imprevistos

A continuación mostramos los formatos mencionados anteriormente:

ALTA PRELIMINAR DE PROYECTO

FECHA:

NUMERO DE ALTA PRELIMINAR:	IMPORTE ESTIMADO:				
NOMBRE DEL PROYECTO:					
RESPONSABLE : LÍDER:	TEL: TEL:				
DURACIÓN APROXIMADA:					
DIVISIÓN O CAMPOS DISCIPLINARIOS:					
RAZÓN SOCIAL O DENOMINACIÓN DEL CONTRATANTE:					
DEPENDE DE CONVENIO: GENERAL () COLABORACIÓN () ESPECÍFICO ()					
NUMERO	DE	REGISTRO	Y	VALIDACIÓN	UNAM

OBJETIVO:					
ALCANCE:					
FIRMA DEL RESPONSABLE			FIRMA DEL LÍDER		

APERTURA DE PROYECTO

FECHA:

NOMBRE DEL PROYECTO:

IMPORTE:

DIVISIÓN / CAMPOS DISCIPLINARIOS:

RAZÓN SOCIAL O DENOMINACIÓN DEL CONTRATANTE:

NÚMERO DE CONVENIO

OBJETIVO:

ALCANCE:

BENEFICIOS ACADÉMICOS:

FUTUROS PROYECTOS:

BENEFICIOS ECONÓMICOS:

ESPECIFICACIÓN DEL MONTO OPERATIVO

1.-Remuneraciones Personales (grupo 100)	\$ _____
2.-Servicios (grupo 200)	\$ _____
3.-Becas, Prestaciones y Estímulos (grupo 300)	\$ _____
4.-Artículos y Materiales de Consumo (grupo 400)	\$ _____
5.-Mobiliario y Equipo (grupo 500)	\$ _____
6.-Inmuebles y Construcciones (grupo 600)	\$ _____
7.-Imprevistos	\$ _____
Total:	\$ _____

PRESUPUESTO GENERAL DEL PROYECTO

A) MONTO OPERATIVO	\$ _____
B) FONDO F.I (15%)	\$ _____
C) INDIRECTOS UNAM (20%)	\$ _____
D) IMPORTE TOTAL DE PROYECTO	\$ _____

Elaboró

Revisó

Visto Bueno

Nombre y Firma Líder Proyecto

Nombre y Firma Responsable Proyecto

Nombre y Firma Secretario Administrativo

AUTORIZACIÓN

Nombre y firma del titular de la dependencia

Anexo B

Plan de actividades

Id	Nombre de tarea	Duración
1	Sistema de Información para Proyectos que Generan Ingresos Extraordinarios (SIPIEX)	223 días
2	Definición de objetivos y alcances	5 días
3	Recopilación de especificaciones y requerimientos generales	5 días
4	Fase de Inicio:	28 días
5	Plan para la fase de inicio del proyecto	2 días
6	Evaluación de los alcances y riesgos del proyecto	2 días
7	Desarrollo del plan de trabajo inicial	2 días
8	Planeación del monitoreo y control del proyecto	2 días
9	Revisión de los alcances y riesgos	1 día
10	Requerimientos	3 días
11	Análisis del problema	2 días
12	Entendimiento de las necesidades del cliente	1 día
13	Definir los alcances del sistema	1 día
14	Modelo de Negocios	3 días
15	Reconocimiento de procesos	3 días
16	Proyectos en Promoción	3 días
17	Proyectos en Operación	3 días
18	Usuarios	3 días
19	Responsables o líderes de proyecto	3 días
20	Contratantes	3 días
21	Divisiones	3 días
22	Memoria Tecnológica	3 días
23	Análisis del Sistema	9 días
24	Vision general del sistema	2 días
25	Identificación los actores	3 días
26	Identificación de los eventos	3 días
27	Identificación de los casos de uso	3 días
28	Identificar de las trayectorias	2 días
29	Trayectorias principales	2 días
30	Trayectorias alternativas	2 días
31	Trayectorias de excepción	2 días
32	Definición de las iteraciones	4 días
33	Identificación de los procesos para 1ra iteración	2 días
34	Procesar proyectos en promoción	2 días
35	Procesar grupo de trabajo	2 días
36	Procesar contratante	2 días
37	Procesar división	2 días
38	Identificación de los procesos para 2da iteración	2 días
39	Procesar proyecto en operación	2 días
40	Procesar apertura de proyecto	2 días
41	Procesar información financiera	2 días
42	Procesar información jurídica	2 días
43	Procesar información tecnológica	2 días
44	Aquitectura preliminar	1 día
45	Documentación y diagramas preliminares	10 días

Plan de Actividades

Id	Nombre de tarea	Duración
46	Fase de Elaboración:	101 días?
47	Plan para la fase de Elaboración del proyecto	1 día
48	Plan de desarrollo para las iteraciones	1 día
49	Análisis del problema	1 día
50	Recopilación de requisitos	2 días
51	Análisis del comportamiento del sistema	3 días
52	Revisión de los alcances y riesgos del proyecto	1 día
53	Detalle de los actores	3 días
54	Detalle de los casos de uso	3 días
55	Identificación de clases	1 día?
56	Identificar los atributos y métodos de las clases de entidad	1 día?
57	Clases de entidad	3 días
58	Clases de límite	3 días
59	Clases de control	3 días
60	Diagrama de clases	3 días
61	Documentación y diagramas	5 días
62	Diseño de la base de datos	6 días
63	Diagrama entidad-relación	2 días
64	Diseño de tablas	2 días
65	Diccionario de datos	2 días
66	Diagramas de Secuencia	30 días
67	Módulo Proyectos en Promoción	20 días
68	Módulo Proyectos en Operación	20 días
69	Módulos de Responsables o Líderes de Proyecto	20 días
70	Módulo Contratantes	10 días
71	Módulo Divisiones	10 días
72	Diseño de la interfaz de usuario	35 días
73	Diseño del sitio Web	10 días
74	Diseño de pantallas	20 días
75	Diseño de componentes	5 días
76	Plan de integración de componentes	1 día
77	Pruebas de los componentes diseñados	2 días
78	Documentación y diagramas	15 días

Id	Nombre de tarea	Duración
79	Fase de Construcción:	63 días
80	Recopilación completa de requisitos	2 días
81	Análisis de los requisitos restantes	2 días
82	Completar las clases restantes	2 días
83	Completar los casos de uso restantes	2 días
84	Delinear los casos de uso existentes	2 días
85	Desarrollar los diagramas restantes	5 días
86	Obtención de la arquitectura del sistema actualizada	2 días
87	Redefinición de la arquitectura de la Base de datos	2 días
88	Desarrollar la interfaz del usuario	52 días
89	Construcción de pantallas	47 días
90	Programación de Páginas WEB	47 días
91	Creación de componentes	5 días
92	Creación de páginas Web	5 días
93	Scripts para Páginas Web	2 días
94	Codificación	30 días
95	Pruebas	1 día
96	Corrección de errores	2 días
97	Revisión de la codificación	3 días
98	Plan de Integración de subsistemas	1 día
99	Integración de cada subsistema	1 día
100	Pruebas	1 día
101	Realizar las pruebas de unidad de todos los componentes	1 día
102	pruebas de integración de los componentes	1 día
103	Evaluación de las pruebas	1 día
104	Creación del manual de usuario preliminar	5 días
105	Obtención de plan de proyecto para la siguiente fase	1 día
106	Creación de la versión operativa de sistema	1 día
107	Fase de Transición	34 días
108	Corrección de errores encontrados en la etapa de pruebas de fase anterior	1 día
109	Obtención de la versión completa y actualizada de la arquitectura	1 día
110	Implantación	2 días
111	Plan de integración de todos los componentes	1 día
112	Implantación de los componentes	2 días
113	Integración de cada subsistemas	1 día
114	Pruebas	3 días
115	Plan de pruebas	1 día
116	Prueba de de módulo de proyectos en promoción	2 días
117	Prueba del módulo de Proyectos en Operación	2 días
118	Prueba del módulo de Usuarios	2 días
119	Prueba del módulo de Permisos	2 días
120	Prueba del módulo de Responsables o Líderes de proyecto	2 días
121	Prueba del módulo de Contratantes	2 días
122	Prueba del módulo de Divisiones	2 días
123	Prueba de Memoria Tecnológica	2 días
124	Evaluar las pruebas	10 días
125	Pruebas y Correcciones	10 días
126	Pruebas en Stand Alone	5 días
127	Pruebas en Línea	5 días
128	Manuales del Sistema	7 días
129	Manual Técnico	7 días
130	Manual de Usuario	7 días
131	Liberación de Sistema	12 días
132	Entrega del Sistema	12 días
133	Últimas Correcciones	5 días
134	Entrega del sistema ejecutable	1 día
135	Entrega del sistema de instalación	1 día
136	Entrega de la versión final del Sistema	5 días

Con base en el plan de actividades mencionado anteriormente, vamos a obtener el tiempo de desarrollo del proyecto y el costo del mismo. El costo es únicamente una referencia, ya que los sueldos mencionados son ficticios.

- Total de días laborables = 223
- Total de meses = 10.6

- Número de integrantes = 2
- Sueldos mensual por persona = \$ 10,080.00
- Costo Total del Proyecto = \$ 214,080.00

Referencias

- Ceballos Sierra, Javier F; 2002; Java 2. Curso de Programación; Ed. Alfaomega rama; Primera Edición; México.
- Diplomado de Redes de Computadoras y Telecomunicaciones. Material Didáctico; 1999; DECFI; Palacio de Minería, México.
- Hall, Marty; Servlets y JavaServlet Pages. Guía Práctica; Ed. Prentice Hall; Primera Edición; México
- Johnson, Rod; 2003; Expert One to One J2EE Design and Development; Ed. Wiley Publishing; Primera Edición; E.U.A.
- Reed, Paul R; 2001; Developing applications with Java and UML; Ed. Addison Wesley; Primera Edición; E.U.A.
- Rumbaugh James, Jacobson Ivar, Booch Grady; 2000; El lenguaje unificado de modelado. Manual de referencia; Ed. Addison Wesley; Primera Edición; España.
- Joyanes Aguilar Luis; 1998; Programación orientada a objetos, Ed. Osborne McGraw-Hill Segunda Edición; México.
- <http://advocacy.postgresql.org/advantages/?lang=es>; Ventajas de PostgreSQL
- <http://jakarta.apache.org/tomcat>; Página principal de Apache-Tomcat.
- <http://java.sun.com>; Página principal de Java Sun Microsystems.
- <http://www.linux.org/>; Página principal de Linux.
- <http://www.postgresql.org/>; Página principal de PostgreSQL
- <http://pp.com.mx/legalfaq.html>; Software Libre.
- http://uxmcc1.iimas.unam.mx/~cursos/Objetos/clases3_4.html; Metodología Booch.
- <http://uxmcc1.iimas.unam.mx/~cursos/Objetos/Omt/omt.html>; Metodología OMT
- <http://www.abits.cl/soluciones/rational/sr.htm>; Metodología Proceso Unificado.
- http://www.aditel.org/~dpecos/docs/mysql_postgres/x15.html; Características de Postgres;
- <http://www.cs.ualberta.ca/~pfiguero/soo/uml/>; Diagramas UML
- <http://www.design-nation.net/es/archivos/000153.php>; Patrones de diseño MVC.
- <http://www.gnu.org>; Software Libre.

- http://www.informatizate.net/articulos/model_view_controller_mvc_20040324.html; Diseño Modelo Vista Controlador
- <http://www.linuxware.com.mx/serverdb.php>; Ventajas de utilizar Linux
- <http://www.rational.com/support/usergroups/rose/otug.jsp>; Metodología Proceso Unificado.

Glosario

- **API (Application Programming Interface).** Desde un lenguaje de alto nivel el API es un conjunto de procedimientos y funciones que se ofrecen al programador para realizar una serie de acciones.
- **Byte code.** El código de bytes es el archivo en lenguaje máquina creado por el compilador de Java independiente del sistema operativo, con el fin de ejecutar una aplicación Java basados en la máquina virtual Java.
- **Capas lógicas.** Es el diseño lógico o conceptual para especificar la estructura de la aplicación, tales capas incluyen el orden de procesamiento, mantenimiento y seguimiento, a diferencia del diseño físico que especifica exactamente donde se encontrarán las piezas de la aplicación (como discos, ejecutables, cable de red y computadoras).
- **CGI (Common Gateway Interface).** Es un protocolo genérico que permite extender las capacidades de HTTP, su función principal es la de añadir mayor interacción a los documentos web que por medio del HTML se presentan de forma estática para aportar un contenido dinámico.
- **Cliente/Servidor.** Es un sistema en el que una máquina llamada cliente solicita a una segunda máquina llamada servidor que ejecute una tarea específica. El cliente suele ser una computadora personal común y el servidor es, por lo general, una máquina anfitriona la cual procesa las peticiones de los clientes.
- **Compilador.** Es un programa que toma los datos de un programa escrito en lenguaje de alto nivel y da como resultado el programa escrito en lenguaje máquina. En particular, el lenguaje máquina creado por el compilador Java es conocido como Byte Code (código de bytes).
- **Elementos persistentes de una aplicación.** Desde la perspectiva de la POO (programación orientada a objetos) son aquellos elementos, objetos o entidades que existen más allá del tiempo de vida de la aplicación y por ello necesitan guardarse en una base de datos o cualquier unidad de almacenamiento.
- **Intérprete.** Es un programa que efectúa la traducción y ejecución simultáneamente para cada una de las sentencias de la aplicación. Es el que verifica cada línea del programa cuando se escribe.
- **JavaBean.** Es una clase de Java estándar que cumple con ciertas características de nomenclatura y diseño. Su uso contribuye a migrar la programación a un modelo de componentes, debido a esto son objetos reutilizables.
- **Java Database Connectivity (JDBC).** Es una API de Java para el acceso a bases de datos y consiste en un conjunto de clases e interfaces para proporcionar una base común para la construcción de herramientas y utilidades de alto nivel.

- **Lenguaje de alto nivel.** Es el lenguaje de programación que utiliza una terminología fácilmente comprensible, esto es, se aproxima más al lenguaje humano.
- **Lenguaje de Programación.** Son instrucciones creadas para la comunicación con dispositivos electrónicos por ejemplo con computadoras, con el objetivo de que realicen alguna acción y arrojen un resultado.
- **Manejador de base de datos.** DBMS por sus siglas en inglés Data Base Management System. Es un sistema que permite crear, extraer, almacenar y manipular la información de una base de datos.
- **Máquina virtual Java.** Es un programa que permite que se ejecute una aplicación Java en cualquier computadora que la tenga instalada. Esta es una de las características que hace a Java un lenguaje independiente de plataforma.
- **Modelo Multicapas.** Es un diseño de la arquitectura de sistemas que consiste en dividir los componentes de la aplicación y programarlos por separado para después unirlos. Comúnmente se dividen en tres capas, sin embargo la división puede ser mayor, dependiendo de la visión del diseñador y de las necesidades del sistema. Estas tres capas se denominan: Presentación, Reglas del Negocio y Acceso a la Base de Datos.
- **Normalización de una base de datos.** Es el proceso de simplificar la relación entre los campos de un registro, esto es, se siguen un conjunto de normas que nos aseguran la creación de bases de datos flexibles, además de evitar la redundancia e inconsistencia de datos. Existen varias reglas de normalización y son conocidas como formas normales.
- **Página dinámica.** Es una página Web cuyo contenido se genera a partir de una petición en la página o en un formulario, por ejemplo mostrar o actualizar el contenido de una base de datos.
- **Página estática.** Es una página Web cuyo contenido se encuentra incluido en un archivo HTML, lo que significa que su contenido no cambia.
- **Página Web.** También conocida como página de Internet es un documento electrónico que contiene información con texto, imágenes, audio, video y otros elementos a los que uno puede acceder a través de enlaces sobre un tema en particular y es cargado en Internet para ser accedido por cualquier persona que se conecte a esta red mundial y cuente con los permisos apropiados. La página Web puede ser dinámica o estática.
- **Reglas del Negocio.** Son los métodos creados con el propósito de regular alguna acción del usuario y son definidas en base a las políticas de la compañía. Por ejemplo, en una aplicación bancaria una regla del negocio podría ser que el cliente no debe retirar por taquilla más de \$100,000 pesos y en caso de una petición de este tipo se genere un error.
- **Servidor Web.** Es un programa que escucha las peticiones HTTP que le llegan. Dependiendo del tipo de la petición el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado al cliente o navegador que realizó la petición.

- **HTTP (Hyper Text Transfer Protocol).** Es el protocolo que permite que los servidores de Web y los navegadores interactúen entre sí, transfiriendo páginas HTML desde el servidor hasta las computadoras que lo soliciten.
- **Sistema Operativo (SO).** Es el programa que administra todos los recursos de la computadora, esto es, define qué aplicaciones y en qué orden serán ejecutadas, maneja la memoria del sistema, los dispositivos de entrada, de salida y envía mensajes de error o información necesaria para el trabajo estable.