

Universidad de las Ciencias Informáticas.
Facultad 10.



Trabajo de diploma para optar por el título de Ingeniero Informático.

Propuesta de Diseño para: Observatorio de Tecnologías Libres.

Autor: Maylú Martínez González.

Tutor: Ing. Abel Meneses Abad.

Ciudad de la Habana, Cuba, 2007.

“Año 49 de la Revolución.”

“ La libertad no es poder elegir entre unas pocas opciones impuestas, sino tener el control de tu propia vida. La libertad no es elegir quién será tu amo, es no tener amo ”.

Richard Stallman

Declaración de Autoría.

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

<nombre autor>

<nombre tutor>

Firma del Autor

Firma del Tutor

Tutor: Ing. Abel Meneses Abad: Profesor graduado de Ingeniería en Telecomunicaciones y Electrónica en el año 2004. Ha impartido asignaturas como Sistemas Operativos, Teleinformática II, Práctica Profesional; de las asignaturas del segundo perfil: GNU/Linux Básico, Propiedad Intelectual, Programación en Perl; e imparte postgrados de GNU/Linux nivel Básico, Programación Web. Posee categoría docente de Instructor recién graduado; ha cursado postgrados como: Ciencia, Tecnología y Sociedad, Gnu/Linux, Ideología y Política de la Revolución Cubana. Ha presentado ponencias en eventos y forma parte del grupo de investigaciones de Software Libre de la UCI (Universidad de las Ciencias Informáticas) y del Grupo Técnico Nacional. Es líder del Proyecto Unicornios (Servicios Especializados para la Migración a SWL (Software Libre) de la UCI).

Primero agradezco a nuestra inmensa **Revolución Cubana**, que me dio la posibilidad de estudiar y realizar mis sueños.

Agradezco a mis padres y hermano: **María Luz González Rojas, Mauricio Martínez Molina y Maury Martínez González**, sin los cuales no hubiese podido lograr mis metas.

A mi abuelo **Chacho**, que la vida no dio la oportunidad de verme crecer.

Un lugar especial para **Isis Rodríguez Pérez**, que ha sido mi soporte durante estos cinco años.

A Pepe, por ser mi compañero de lucha en los momentos más impensados.

A María Antonia Montesino Menéndez por ser mi otra mamá en la UCI, y a Vela por confiar en mí cuando muchas cosas fallaron.

A mis compañeras de apartamento: Dayana, Zuyén, Yadira, Sahily, Yahilín, Yenisey, Ariamna, Katia, Ilio y Karelia, por ser mi familia en la universidad.

A mi entrañable amigo **Abel Meneses Abad**, por contribuir a mi formación como ingeniera y persona.

A todos los que cooperaron en la realización del presente trabajo: Raciél Garrido Torres, Ramón Paumier Samón, Yanedi Abreu Bartomeo, Luis Domínguez Cruz, Yusliel, Alejandro, Yeisel, Dunia María, Xory, Karel, Derick.

A Juan Antonio Plasencia Soler (**Toño**), por ser mi amigo incondicional y enseñarme más de lo que cree.

A Héctor Rodríguez Figueredo por no permitir que el tiempo borrara las huellas y darme otra oportunidad.

A Yenisleidy Cariaga Cristo, por su apoyo incondicional.

A José Francisco Bobadilla Díaz por haber sido una guía en mi vida.

A Reinier Bonne Sharon, por mantener nuestra amistad a pesar de la distancia.

A mis compañeros de teatro y mi profesor César, donde pasé momentos de tensión.

A Frank, que aún en la distancia, siempre me ha tenido presente.

A todos mis compañeros de aula y cadetes de quinto año, con quienes compartí mi vida estos cinco cursos.

Y a todos los que de una forma u otra, hicieron posible la realización de este trabajo.

A todos: Muchas Gracias.

Quiero dedicar este trabajo que me definirá como profesional a dos personas muy importantes en mi vida. En mi primer lugar: A nuestro invicto Comandante en Jefe **Fidel Castro Ruz**, por ser luz y guía de mi formación revolucionaria, humana, social y personal, y porque sé que sería también el deseo de mi segunda persona importante.

A mi abuela **Caridad Enriqueta Molina Triana**, por no haberme dejado sola ni un minuto, y por no haber tenido la oportunidad de decirle cuan importante fue en mi vida, cuanto la quise y la quiero, y por haberse perdido, por cosas del destino, los momentos más importantes de mi vida.

En Cuba donde el bloqueo impuesto limita, entorpece, y encarece la adquisición del software legal necesario para el desarrollo de la sociedad, se manifiesta la perspectiva del Software Libre como solución a esta problemática para el desarrollo de la informática en la isla. Ante la tarea de enfrentar un proceso de migración de Software Privativo a Software Libre, se hace difícil la toma de decisiones sobre que aplicaciones usar en cada caso, primero por la gran cantidad de software existente en el mundo para un mismo propósito, y segundo porque la información publicada en Internet no siempre puede ser considerada como fiable. Se hace entonces necesario contar con un mecanismo confiable para orientar a los organismos del estado que se encuentran en el proceso de migración sobre que tecnologías y que aplicaciones usar en cada momento. El objetivo de este trabajo de diploma ha sido obtener una propuesta de diseño de un servicio Web para la UCI, que permita dar asesoría a empresas o instituciones del estado que se encuentran en procesos de migración a plataformas libres. Este servicio permitirá a los clientes informarse sobre las tecnologías y aplicaciones existentes, así como compararlas. Pueden además obtener asesoría personalizada para la migración de sus empresas y dará facilidades de edición a los miembros del proyecto para mantener actualizada la información.

Palabras Claves:

- ✓ Observatorio de Tecnologías Libres.
- ✓ Software Libre.
- ✓ Información.
- ✓ Tecnología.
- ✓ Aplicación.
- ✓ Diseño.
- ✓ Migración.
- ✓ Servicio Web.
- ✓ Asesoría.

Índice.

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 INTRODUCCIÓN.....	5
1.2 DEFINICIÓN DE OBSERVATORIO TECNOLÓGICO.....	6
1.3 OBSERVATORIOS TECNOLÓGICOS A NIVEL INTERNACIONAL.....	7
1.3.1 Tigris.....	7
1.3.2 CMSmatrix.....	8
1.3.3 Netcraft.....	8
1.3.4 IDC.....	8
1.4 PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE (RUP).....	9
1.5 TECNOLOGÍAS UTILIZADAS.....	10
1.5.1 PHP.....	11
1.5.2 MySQL.....	11
1.5.3 Visual Paradigm.....	13
1.6 CLASIFICACIÓN GLOBAL DE LAS TECNOLOGÍAS LIBRES.....	13
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA PROPUESTO: OBSERVATORIO DE TECNOLOGÍAS LIBRES.....	18
2.1 INTRODUCCIÓN.....	18
2.2 DESCRIPCIÓN DE PLANILLA TECNOLÓGICA.....	18
2.3 MODELO DEL DOMINIO.....	18
2.4 PROPUESTA DE SISTEMA.....	21
2.4.1 Gestionar Información de Tecnologías.....	22
2.4.2 Comparar las aplicaciones.....	23
2.4.3 Definición de Usuarios y permisos.....	24
2.4.4 Solicitud de asesoría personalizada.....	25
2.5 ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE.....	25
2.5.1 Requerimientos Funcionales.....	25
2.5.2 Requerimientos no funcionales.....	28
2.6 DEFINICIÓN DE LOS CASOS DE USO.....	30
2.6.1 Definición de actores del sistema.....	30

2.6.2 Definición de casos de uso del sistema.	32
2.6.3 Descripción de los Casos de Uso del Sistema.	36
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.	46
3.1 INTRODUCCIÓN.	46
3.2 ANÁLISIS.	46
3.2.1 Diagrama de clases del análisis de Caso de Uso: Autenticar usuario.	47
3.2.2 Diagrama de clases del análisis de Caso de Uso: Administrar usuario.	47
3.2.3 Diagrama de clases del análisis de Caso de Uso: Comparar aplicaciones.	48
3.2.4 Diagrama de clases del análisis de Caso de Uso: Gestionar tecnología.	48
3.2.5 Diagrama de clases del análisis de Caso de Uso: Visualizar información.	49
3.2.6 Diagrama de clases del análisis de Caso de Uso: Visualizar servicios.	49
3.2.7 Diagrama de clases del análisis de Caso de Uso: Modificar datos usuario.	49
3.2.8 Diagrama de clases del análisis de Caso de Uso: Solicitar asesoría.	50
3.2.9 Diagrama de clases del análisis de Caso de Uso: Gestionar solicitud.	50
3.2.10 Diagrama de clases del análisis de Caso de Uso: Gestionar aplicación.	51
3.2.11 Diagrama de clases del análisis de Caso de Uso: Actualizar aplicación.	51
3.2.12 Diagrama de clases del análisis de Caso de Uso: Registrar Usuario.	52
3.2.13 Diagrama de clases del análisis de Caso de Uso: Configurar reporte.	52
3.3 DIAGRAMAS DE INTERACCIÓN.	53
3.3.1 Diagrama de colaboración del caso de uso: Autenticar usuario.	53
3.3.2 Diagrama de colaboración del caso de uso: Administrar usuario.	53
3.3.3 Diagrama de colaboración del caso de uso: Comparar aplicaciones.	54
3.3.4 Diagrama de colaboración del caso de uso: Gestionar tecnología.	55
3.3.5 Diagrama de colaboración del caso de uso: Visualizar información.	56
3.3.6 Diagrama de colaboración del caso de uso: Visualizar servicios.	56
3.3.7 Diagrama de colaboración del caso de uso: Modificar datos de usuario.	57
3.3.8 Diagrama de colaboración del caso de uso: Solicitar asesoría.	57
3.3.9 Diagrama de colaboración del caso de uso: Gestionar solicitud.	58
3.3.10 Diagrama de colaboración del caso de uso: Gestionar aplicación.	59
3.3.11 Diagrama de colaboración del caso de uso: Actualizar aplicación.	60
3.3.12 Diagrama de colaboración del caso de uso: Registrar usuario.	61
3.3.13 Diagrama de colaboración del caso de uso: Configurar reporte.	62
3.4 DISEÑO.	63

3.4.1	Diagramas de Clases del Análisis.....	64
3.4.1.1	Diagrama de Clases del Diseño del Caso de uso: Autenticar usuario.	64
3.4.1.2	Diagrama de Clases del Diseño del Caso de uso: Administrar usuario.	64
3.4.1.3	Diagrama de Clases del Diseño del Caso de uso: Comparar aplicaciones.	65
3.4.1.4	Diagrama de Clases del Diseño del Caso de uso: Gestionar tecnología.	66
3.4.1.5	Diagrama de Clases del Diseño del Caso de uso: Modificar datos de Usuario.	66
3.4.1.6	Diagrama de Clases del Diseño del Caso de uso: Solicitar Asesoría.	67
3.4.1.7	Diagrama de Clases del Diseño del Caso de uso: Gestionar Solicitud.	67
3.4.1.8	Diagrama de Clases del Diseño del Caso de uso: Gestionar Aplicación.	68
3.4.1.9	Diagrama de Clases del Diseño del Caso de uso: Actualizar Aplicación.	69
3.4.1.10	Diagrama de Clases del Diseño del Caso de uso: Registrar Usuario.	69
3.4.1.11	Diagrama de Clases del Diseño del Caso de uso: Configurar reporte.	70
3.4.2	Descripción de las clases del diseño.	70
3.5	DISEÑO DE LA BASE DE DATOS.	75
3.5.1	Diagrama de Clases Persistentes.	76
3.5.2	Modelo de la Base de datos.	77
3.5.3	Descripción de las tablas de la Base de Datos.	78
	CONCLUSIONES.....	80
	RECOMENDACIONES.	81
	BIBLIOGRAFÍA.....	84
	GLOSARIO DE TÉRMINOS.....	87
	ANEXOS.....	91

Introducción.

El Software Libre se abre paso, cada vez con mayor fuerza en empresas, instituciones y administraciones públicas a nivel mundial, impulsado por las ventajas que representan la liberación de las tecnologías informáticas. Partiendo de que el conocimiento, como el Software Libre, es un bien cuya universalización permite el desarrollo de la sociedad en todos sus ámbitos, la libertad e independencia tecnológica que proporciona el uso de software y estándares tecnológicos libres son aspectos fundamentales que se deben tener en cuenta sobre todo como los responsables del desarrollo de una sociedad verdaderamente solidaria para todos. El Software Libre se basa de cierta forma en compartir el saber, por lo que se convierte en un impulsor para el desarrollo, favoreciendo y agilizando la innovación.

En Cuba, donde el bloqueo impuesto limita, entorpece, y encarece la adquisición del software legal necesario para el desarrollo de la sociedad, y además, como defensores mundiales de la cooperación internacional en función de una Sociedad Mundial de la Información justa, equitativa y solidaria, se manifiesta la perspectiva del Software Libre como solución a esta problemática para el desarrollo de la informática en la isla. Se puede decir que se han dado ya los primeros pasos en cuestiones de migración en este país, citando como ejemplo fehacientes a Infomed, que usa GNU/Linux en todos sus servidores, la Aduana de la República de Cuba que hacen uso del Software Libre en su desempeño diario, y la Empresa de Tabacos Partagás.

Cuando se enfrenta un proceso de migración, se hace necesario asumir un grupo de decisiones importantes corriendo los riesgos típicos de lo que pueda implicar la falta de conocimiento científico sobre el tema. Decisiones sobre distribuciones de sistema operativo, gestores de bases de datos o herramientas de desarrollo se imponen en las entidades del estado, y se hace complicado definir qué sería una solución correcta sin tener previamente estipulado qué parámetros especifican las características que distinguen lo más fiable tecnológicamente en cada caso. Para efectuar un proceso de migración con calidad y minimizar los posibles errores a cometer, se hace necesario tener una base documentada y fundamentada de las clasificaciones de las tecnologías libres que brinde información legal y confiable.

En medio de la contienda de migrar a un sistema más justo para la libertad de usuario, es importante tener una guía que facilite tomar las decisiones correctas y lograr la eficiencia en un proceso de este tipo. O sea, se propone un modelo que permita a una organización o entidad, tener una base fundamentada en una clasificación de las tecnologías libres, dando la posibilidad de decidir que usar en cada caso, más en estos momentos donde crece vertiginosamente la cantidad y diversidad de software disponible para diferentes tecnologías informáticas que se desarrollan desde diversos lugares y proyectos. A esto se debe incluir el exceso de información en la Internet, donde no toda es confiable y se hace fácil la desorientación de usuarios en cuanto a tecnologías y aplicaciones.

Por solo citar algunos ejemplos, existen más de quinientas herramientas CASE reconocidas y documentadas para ser utilizadas en el desarrollo del proceso de Ingeniería del software. Esto hace necesario tener controladores virtuales de tráfico para poderse orientar en un mundo de tanta diversidad. El Observatorio Tecnológico tendría esa función, es parte de sus necesidades. El mismo consiste fundamentalmente en un sistema de información-difusión continuo y sistematizado, que describe la situación actual y las tendencias de la Ciencia y la Tecnología.

En cuanto a este tipo de trabajo, existe una experiencia previa que marca de alguna forma un punto de partida para el propósito de este trabajo. Es precisamente la experiencia del Portal de Software Libre de la facultad número diez. Los primeros pasos de asesoría en temas de Software Libre, se dieron hace dos años, donde se crean en el portal los Foros donde la comunidad decidía que tecnologías usar.

Siguiendo este esquema, surge más tarde el servidor de documentación, asumiendo parte de lo propuesto en el Foro y enriqueciendo una clasificación que queda a disposición de los usuarios, aunque esto todavía no satisfacía las necesidades de la comunidad. Nace entonces foro.prod.uci.cu, el foro de la comunidad de desarrollo, auspiciado por la Federación Estudiantil Universitaria de la Universidad de las Ciencias Informáticas creando un espacio con características similares a los anteriores, donde se pone a disposición de la comunidad un conjunto un poco más amplio de tecnologías, todavía de una forma informal y hasta cierto punto inocuo.

Por tanto el **problema** a resolver consiste en:

No existe un servicio de asesoría legal y certificado para orientar a los organismos del estado que se encuentran en el proceso de migración. Tampoco de manera central existe un servicio para orientar a los polos de desarrollo sobre que tecnologías usar a medida que transcurre el proceso de migración en Cuba.

Se persigue como **Objetivo General**: Analizar y diseñar un servicio Web para la UCI, que permita dar asesoría a empresas o instituciones del estado que se encuentran en procesos de migración a plataformas libres.

Para dar solución a la problemática se plantean los siguientes **Objetivos Específicos**:

- ✓ Investigar todas las clasificaciones posibles para las tecnologías libres.
- ✓ Diseñar un sistema de reportes, clasificación comparación para cada tecnología bien definida.
- ✓ Diseñar el servicio del observatorio tecnológico, de manera funcional y organizacional.

Dentro de las **tareas** que se proponen para dar solución a los objetivos planteados están:

- ✓ Revisión y búsqueda de observatorios internacionales de tecnologías informáticas.
- ✓ Estudio de las clasificaciones globales existentes de las tecnologías libres.
- ✓ Identificar aplicaciones específicas para cada tecnología.
- ✓ Definir parámetros a medir de esas tecnologías y aplicaciones.

Se espera que una vez concluido el trabajo se pueda contar con:

- ✓ Estudio y clasificación de nivel medio para todas las tecnologías libres.
- ✓ Plantillas estándares para emitir un servicio de asesoría, a partir del observatorio.
- ✓ Análisis y diseño de la Aplicación Web: Observatorio de Tecnologías libres.

El **Objeto de estudio** son los Procesos de migración a plataformas libres.

El **Campo de acción** se refiere a Procesos de migración a plataformas libres de los organismos del estado en Cuba.

Hipótesis: Si se desarrolla un servicio online semi-presencial para la UCI, que de asesoría para las empresas o instituciones del estado que se encuentran en procesos de migración a plataformas libres, entonces se podrá efectuar el proceso de migración de forma eficiente.

Estructuración del contenido:

Capítulo 1: Fundamentación del tema, situación nacional e internacional de las tendencias y tecnologías actuales: En este capítulo se explican las principales clasificaciones de tecnologías libres, la metodología empleada, las herramientas utilizadas para el desarrollo de la aplicación y el estado del arte de los observatorios de tecnologías.

Capítulo 2: Características del sistema. En este capítulo se describen los procesos automatizados. Además, se presentan los documentos específicos que se procesan, acompañados de una descripción general de la propuesta de sistema que se presenta, especificando el funcionamiento global del mismo. Se muestra además la modelación del negocio del observatorio mediante un modelo del dominio, así como todos los requerimientos funcionales y no funcionales que se deben cumplir.

Capítulo 3: Análisis y diseño del sistema. Este capítulo aborda cuestiones fundamentales sobre el análisis y diseño del sistema, utilizando para su modelado el Lenguaje Unificado de Modelación (UML). Se presentan los resultados de estas etapas, incluyéndose el modelo de clases de análisis, los diagramas de secuencia y los diagramas de clases del diseño, la descripción detallada de las clases empleadas en el diseño del sistema y el modelo de datos que soportará el mismo.



Capítulo 1: Fundamentación Teórica.

1.1 Introducción.

Partiendo desde el punto de vista de que existen más de mil distribuciones, si en Cuba, se tuviera la tarea de migrar completamente a Software Libre: ¿Qué distribución de sistema Operativo libre se utilizaría? Es por eso que se hace necesario contar con un servicio de asesoría que permita tomar decisiones acertadas en momentos precisos.

¿Qué es asesoría? Asesoría no es más que la acción de brindar consejo o ayuda mayormente especializada, sobre un tema en particular, apoyando la toma de decisiones correctas y procurando un desempeño fiable de la tarea en cuestión. Y este es el objetivo fundamental del observatorio, o sea, asesorar, brindar sugerencias prácticas a polémicas reales, en este caso: la migración.

Para este tipo de trabajo se necesitaría contratar un especialista que guiara en los pasos fundamentales a seguir como parte del procedimiento de migración, incluso se podría necesitar más de un especialista, a la hora de seleccionar no solo sistema operativo, sino gestor de bases de datos o herramientas de desarrollo entre otras aplicaciones. Pues lo que se propone es aunar esfuerzos en la elaboración de un sitio con la posibilidad de asesorar de forma guiada y sustentada los procesos de migración en el país, un Observatorio de Tecnologías Libres. Este capítulo muestra una panorámica de los Observatorios de Tecnologías que existen, las herramientas y tecnologías que se utilizarán para darle solución a la problemática y concluye con la presentación de la clasificación de tecnologías libres, aplicaciones y parámetros a medir que soportará el observatorio.



1.2 Definición de Observatorio Tecnológico.

El contenido de información que ha generado en el último siglo la humanidad es extraordinario. Su existencia en múltiples plataformas: libros, revistas, objetos, obras de arte, han impulsado el desarrollo y el bienestar del hombre en el planeta. En los comienzos del renacimiento y con la vida de Leonardo Da Vinci se considera por algunos especialistas como el fin de los grandes sabios, aquellos hombres que dominaban toda la ciencia escrita hasta su época. Desde entonces fue imposible para un hombre saber todo lo que sus predecesores en la humanidad habían escrito. Desde la aparición de Internet, el paradigma se ha multiplicado, se escribe cada año – entre obras valiosas y otras de menor valor – tanto como la historia de la humanidad antes del mismo.

Con la llegada de la especialización, la humanidad consiguió organizar el conocimiento en materias bien definidas, y en campos donde muchas personas representaron el papel de los grandes sabios (hombres que dominaron todo el conocimiento de la época), aunque por poco tiempo. Así por ejemplo en la rama de la biología, hubo que construir nuevas esferas y hacer especialistas dedicados a la parte animal, vegetal, microbiológica entre otras.

En el entorno informático la generación de ideas, nuevos conocimientos, aún bajo el esquema de la especialización, ha tornado la gestión de la información muy compleja. Es por ello que empresas, instituciones, estados en general, han elaborado sitios informáticos – basados en los lenguajes y tecnologías Web conocidos como los CMS, – dedicados a brindar asesoría en materia de información. Es también muy conocido el fenómeno de la información redundante o falseada que obstaculiza el desarrollo de determinadas investigaciones sobre Internet. Estos sitios son conocidos como Observatorios Tecnológicos.

Los Observatorios Tecnológicos en Internet, o desarrollados en ambiente Web tienen diferentes características y funciones. Algunos brindan desde servicios de noticias, hasta servicios de artículos científicos relacionados con el tema al cual está dedicado dicho observatorio. Ofrecen ayuda, calendario de eventos, información en aspectos legales siempre relacionados con el tema tecnológico al que están dedicados. No existen modelos predefinidos, aunque si algunos conceptos inviolables.



Los observatorios tecnológicos deben tener bien definidos su estructura, objetivos y la cobertura del mismo, además de sus criterios globales de gestión, acceso, servicios y niveles de información. Su función de manera general está destinada a brindar el estado del arte de las tecnologías a las que está dedicada. Es por mucho un servicio de gran actividad y constante mantenimiento. Asociado a los observatorios se mueve un enorme conjunto de fuentes de información y capital humano especializado, que se desarrolla mientras ejecuta el servicio.

1.3 Observatorios Tecnológicos a nivel internacional.

A medida que el Software Libre se desarrolla, surgen nuevas empresas y/o organizaciones que se dedican al control global de la evolución de las tecnologías. En un mundo informatizado, las tecnologías crecen vertiginosamente, y resulta fácil perder la dirección, por lo complicado que se hace seguir el desarrollo de las mismas, sobre todo en cuestiones de Software Libre, donde existen tantas comunidades trabajando de forma paralela y enriqueciendo en incremento cada una de las aplicaciones. En la actualidad existen un conjunto de Observatorios Tecnológicos que sirven como punto de partida para lograr el objetivo trazado. En este capítulo se tomarán como referencia cuatro de estos fundamentalmente: Tigris.org, CMSmatrix.org, IDC.com y NETCRAFT.com.

1.3.1 Tigris.

Uno de los más visitados es **Tigris.org**, dedicado a las herramientas CASE, que provee información de recursos informáticos para profesionales ingenieros del software y estudiantes, y un lugar para las herramientas utilizadas en proyectos de código abierto. La ingeniería del software constituye una parte fundamental en cualquier proceso de desarrollo de magnitud considerable. Desafortunadamente las herramientas de ingeniería de software y métodos no son plenamente utilizados en la actualidad. El desarrollo del código abierto ha producido un número importante y útil de herramientas de desarrollo de software. En este sitio se puede encontrar una valoración de más de quinientas aplicaciones agrupadas en catorce categorías fundamentales referentes a los principales flujos de trabajo que comprende la Ingeniería del Software.



1.3.2 CMSmatrix.

CMSmatrix.org se presenta como servicio comunitario a cualquier interesado que busque una forma de administrar la información en el contexto de un sitio Web. Basado en un conjunto amplio de parámetros definidos, provee una comparación entre los diferentes sistemas más usados en la actualidad, permitiendo un rango de comparación entre diez sistemas como máximo. Cuenta con cerca de quinientos cuarenta CMS clasificados y caracterizados.

1.3.3 Netcraft.

En el caso de **Netcraft** es una compañía de servicio de Internet con base en Inglaterra que sitúa sus prestaciones fundamentalmente en brindar servicios de seguridad en Internet, incluyendo anti-fraud y anti-phishing, pruebas de aplicación, revisión del código, y comprobación de penetración automatizada. Además proporciona investigaciones informativas y de análisis en muchos aspectos de Internet. Netcraft ha explorado Internet desde mil novecientos noventa y cinco y es considerado una autoridad respetada en lo referente a los servicios Web, los sistemas operativos, hosting providers, ISPs, transacciones encriptadas, comercio electrónico. Este sitio cuenta con un conjunto de servicios agrupados en categorías generales que definen las principales direcciones de los mismos. Un primer grupo serían los servicios de seguridad (Anti-Phishing Toolbar, Phishing Site Feed, Bank Fraud Detection, Phishing Site Countermeasures, Audited by Netcraft, Open Redirect Detection, Web Application Security Testing, Web Application Security Course). Existen otras clasificaciones como: Minería de datos, Exploración de Internet, Performace.

1.3.4 IDC.

IDC, es el primer proveedor global de inteligencia de mercado, servicios asesores, y eventos para la tecnología de información, telecomunicaciones, y mercados de tecnología de consumidor. IDC ayuda a profesionales, ejecutivos comerciales, y a la comunidad de inversión en la toma decisiones basadas en hechos respecto a la compra de tecnología y la estrategia comercial. Alrededor de ochocientos cincuenta analistas de IDC en cincuenta países, proporcionan asesoría global, regional, y local sobre tecnologías y oportunidades de industria y tendencias. Por más de cuarenta y dos años, IDC ha proporcionado las visiones estratégicas para ayudar a sus clientes a lograr objetivos



comerciales importantes. Se puede acceder a sus prestaciones a través de los documentos de investigación individuales, además de la existencia de más de quinientos servicios de suscripción de investigación. Incluso cuentan con relaciones directas con los expertos de industria, consultoría personalizadas y conferencias de calidad mundial y eventos.

1.4 Proceso unificado de desarrollo de software (RUP).

Con el desarrollo que está teniendo en la actualidad la industria del software, el avance y complejidad que alcanzan los nuevos sistemas informáticos, debido principalmente al auge de las computadoras, el aumento del rigor del usuario y el rápido crecimiento en el uso de Internet para el intercambio de todo tipo de información, James Rumbaugh, Grady Booch e Ivar Jacobson, autores de “El proceso unificado de desarrollo de software”, opinan que “El problema del software se reduce a la dificultad que afrontan los desarrolladores para coordinar las múltiples cadenas de trabajo de un gran proyecto de software”. Para desarrollar un software se necesita una forma coordinada de trabajo, un proceso que integre las múltiples facetas del desarrollo, cuyo objetivo será producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecido, cubriendo el ciclo de vida y desarrollo de software [1]

El proceso unificado de desarrollo, RUP, es el resultado de la evolución e integración de diferentes metodologías de desarrollo de software. RUP permite sacar el máximo provecho de los conceptos asociados a la orientación a objetos y al modelado visual. Esto permite a los grupos de desarrollo producir aplicaciones informáticas más robustas y flexibles que se adaptan a las necesidades de los usuarios. La correcta aplicación de RUP permite reducir los tiempos de desarrollo, aumentar la calidad de las aplicaciones y disminuir los costes de mantenimiento. Está basado en componentes, lo cuál quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. RUP es un proceso de desarrollo de software que contiene un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software de forma eficiente. Es el resultado de la experiencia de más de treinta años de trabajo y los autores [James Rumbaugh, Grady Booch e Ivar Jacobson] confirman que es la solución al problema del software.



Características del Proceso Unificado de Software.

Los aspectos definitorios y a la vez que convierten en único al Proceso Unificado, se resumen en tres fases: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental.

✓ **Dirigido por casos de uso:** La razón de ser de un sistema es brindar servicios a los usuarios, RUP define caso de uso como el conjunto de acciones que debe realizar un sistema para dar un resultado de valor a un determinado actor y los utiliza tanto para especificar los requisitos funcionales del sistema, como para guiar todos los demás pasos de su desarrollo, dígame diseño, implementación y prueba.

✓ **Estar centrado en la arquitectura:** La arquitectura es una vista del diseño completo con las características más importantes. Esta no sólo incluye las necesidades de los usuarios e inversores, sino también otros aspectos técnicos como el hardware, sistema operativo, sistema de gestión de base de datos, protocolos de red, con los que debe coexistir el sistema. La arquitectura representa la forma del sistema, la cual va madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas.

✓ **Ser iterativo e incremental:** El alto nivel de complejidad de los sistemas actuales, hace que sea factible dividir el proceso de desarrollo en varios mini-proyectos. Cada uno de estos se les denomina iteración y pueden o no representar un incremento en el grado de terminación del producto completo. En cada iteración los desarrolladores seleccionan un grupo de casos de uso, los cuales se diseñan, implementan y prueban. La planificación de iteraciones hace que se reduzcan los riesgos de los costes de un solo incremento, no sacar al mercado un producto en el tiempo previsto, mantener la motivación del equipo pues puede ver avances claros a corto plazo y que el desarrollo pueda adaptarse a los cambios en los requisitos.

1.5 Tecnologías utilizadas.

El Observatorio Tecnológico será implementado utilizando PHP y MYSQL. Además, se utilizará Visual Paradigm en la modelación de la Ingeniería del Software del mismo.



1.5.1 PHP.

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios web. PHP es un acrónimo recurrente que significa "**PHP Hypertext Pre-processor**" (inicialmente PHP Tools, o, *Personal Home Page Tools*), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web.

Se toma la decisión de utilizarlo a partir de que es un lenguaje multiplataforma. Con una capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL. Lee y manipula datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML. Presenta la capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones). Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Una de las características más importantes es que es libre, por lo que se presenta como una alternativa de fácil acceso para todos. Permite las técnicas de Programación Orientada a Objetos, así como crear formularios para la Web.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de Aplicaciones Web muy robustas.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como GNU/Linux), Windows y Mac OS X, y puede interactuar con los servidores de Web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI. Su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU. [2]

1.5.2 MySQL.

MySQL es un sistema de gestión de base de datos, multihilo y multiusuario con más de seis millones de descargas. MySQL es muy utilizado en aplicaciones Web como MediaWiki o Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento



de errores como Bugzilla. Su popularidad como aplicación Web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. En aplicaciones Web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

La licencia GNU GPL de MySQL obliga a distribuir cualquier producto derivado (aplicación) bajo esa misma licencia, otro de los factores por los que se tomó la decisión de utilizar este producto. Entre las características disponibles en las últimas versiones se puede destacar:

- ✓ Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- ✓ Disponibilidad en gran cantidad de plataformas y sistemas.
- ✓ Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- ✓ Transacciones y claves foráneas.
- ✓ Conectividad segura.
- ✓ Replicación.
- ✓ Búsqueda e indexación de campos de texto.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder a y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido. [3]



1.5.3 Visual Paradigm.

Esta herramienta es seleccionada para el modelado del Observatorio Tecnológico, primero por ser una herramienta libre, además, el Visual Paradigm es una herramienta CASE que sustenta UML. La misma ha sido diseñada para usuarios que están interesados en sistemas de software de gran escala basados en tecnología orientada a objetos. [4] La herramienta apoya los estándares más recientes de las notaciones de Java y de UML. [5] Visual Paradigm, es un producto que facilita la organización para el diseño visual de diagramas. Esta herramienta ayuda al equipo de desarrollo de software, a documentar y cubrir el proceso de modelado, construcción y desarrollo de software, acelerando la contribución individual y a nivel de proyecto de cada equipo de desarrollo, que además soporta lenguajes como: Java, C++, CORBA IDL, PHP, XML y Schema. [6]

1.6 Clasificación global de las tecnologías libres.

En este epígrafe se propone una clasificación global de las tecnologías libres que se soportarán, con sus subcategorías, aplicaciones seleccionadas por categorías y parámetros de comparación para dichas aplicaciones. Para realizar esta clasificación, se tuvo en cuenta un grupo de fuentes importantes que avalan la veracidad de la misma. Se partió de foro.prod.uci.cu, el foro de la comunidad de desarrollo, auspiciado por la FEU, donde se agrupan de cierta manera un grupo de tecnologías libres que proporcionan una guía. Esto aunado con lo publicado en FSF/UNESCO Free Software Directory (<http://directory.fsf.org/>) y teniendo en cuenta también lo expuesto en el estudio: Economic impact of open source software on innovation and the competitiveness of the Information and Communication Technologies (ICT) sector in the EU, publicado el veinte de noviembre del dos mil seis, arrojan como resultado la siguiente clasificación. La misma no es definitiva ni estática, es la propuesta base para el desarrollo del Observatorio que a medida que transcurra el tiempo y bajo el proceso de control de cambios respectivo, puede ser modificada a interés del proyecto. [Ver anexo 1.]

Clasificación para tecnologías libres.

1. Sistema Operativo.



Un sistema operativo (SO) es un conjunto de programas destinados a permitir la comunicación del usuario con un ordenador y gestionar sus recursos de manera eficiente. Comienza a trabajar cuando se enciende el ordenador, y gestiona el hardware de la máquina desde los niveles más básicos.

Los sistemas operativos, motivados por su condición de capa software que posibilita y simplifica el manejo de la computadora, desempeñan una serie de funciones básicas esenciales para la gestión del equipo. Entre las más destacables, cada una ejercida por un componente interno (módulo en núcleos monolíticos y servidor en microkernels), podemos reseñar las siguientes:

- ✓ Gestionar los recursos del equipo ejecutando servicios para los procesos (programas.).
- ✓ Brindar una interfaz al usuario, ejecutando instrucciones (comandos). [7]

2. Audio.

El audio digital es la codificación digital de una señal eléctrica que representa una onda sonora. Consiste en una secuencia de números binarios y se obtiene del muestreo y cuantificación digital de la señal eléctrica (que en este tema se llama señal analógica, para contraponerla a la señal digital) posteriormente se puede codificar o comprimir, dando lugar a formatos más compactos (compresión de audio). [8]

En esta categoría, se propone una subclasificación que de cierta manera agrupe todas las herramientas necesarias para la manipulación de audio en su computadora.

3. Base de Datos.

Una base o banco de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En la actualidad, y gracias al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos. [9] Esta categoría está compuesta por dos subcategorías.



4. Gráfico.

En Informática, el término gráfico expresa que usa elementos no hechos de texto. Por extensión, gráficos (siempre en plural) son cualquier imagen generada por ordenador, pero usualmente se les llama gráficos a los diseños de personajes, murallas, texturas. [10]

Todas las aplicaciones que permitan manipular gráfico en un entorno libre, estarán agrupadas en esta categoría de la clasificación de tecnologías libres.

5. Aplicaciones Internet.

Internet es una red de redes a escala mundial de millones de computadoras interconectadas con el conjunto de protocolos TCP/IP. También se usa este nombre como sustantivo común y por tanto en minúsculas para designar a cualquier red de redes que use las mismas tecnologías que la Internet, independientemente de su extensión o de que sea pública o privada. [11]

Esta categoría agrupa un conjunto de aplicaciones necesarias para utilizar este recurso. Desde navegadores Web hasta clientes de mensajería instantánea.

6. Desarrollo de software.

Se denomina software, programática, equipamiento lógico o soporte lógico a todos los componentes intangibles de una computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware). Esto incluye aplicaciones informáticas tales como un procesador de textos, que permite al usuario realizar una tarea, y software de sistema como un sistema operativo, que permite al resto de programas funcionar adecuadamente, facilitando la interacción con los componentes físicos y el resto de aplicaciones. Este software necesita ser desarrollado. El proceso de creación de software es materia de la ingeniería del software. Es un proceso complejo que involucra diversas tareas de gestión y desarrollo. [12]



Aquí se pone a disposición de los usuarios, las aplicaciones necesarias para el desarrollo de software.

7. Ofimática.

Ofimática es un término utilizado para expresar “Organización” y “Tecnología”. Es la perspectiva que estudia el uso de tecnologías de la información en el trabajo de las oficinas.

El rápido crecimiento del sector servicios dentro de la economía, iniciado a mediados de la década de los setenta, ha creado un nuevo mercado de equipos de oficina de tecnología avanzada. La creciente incorporación de microchips y microcircuitos a los equipos ha difuminado la frontera entre el ordenador o computadora y el resto de la maquinaria. Todas las máquinas de oficina modernas (máquinas de escribir, dictáfonos, fotocopiadoras, equipos telefónicos y calculadoras, entre otras) contienen un microprocesador.

En la actualidad, los ordenadores (independientes o conectados a una red) equipados con programas especializados realizan tareas de telecopia, fax, correo de voz o telecomunicaciones, que anteriormente eran realizadas por otros equipos. Las computadoras, además de haber sustituido prácticamente a las máquinas de escribir, a las calculadoras y a los equipos de contabilidad manual, realizan también tareas de diseño gráfico, de planificación, de producción y de diseño de ingeniería.

Los documentos de oficina son, generalmente, registros producidos mecánica o electrónicamente (por ejemplo, cartas, hojas de cálculo, informes y facturas). Se realizan en equipos tales como máquinas de escribir, procesadores de texto y computadoras, y se pueden guardar en papel o en soporte electrónico. En este caso, la categoría Ofimática se refiere a todas las aplicaciones que permitirán este tipo de trabajo.

8. Video.

La palabra **vídeo** o **video** en Latino América, hace referencia a la señal de imagen de televisión. Etimológicamente la palabra Video proviene del verbo latino **Video - Videre**, que significa "Ver". La



señal de vídeo esta formada por un número de líneas agrupadas en varios cuadros y estos a la vez divididos en dos campos porta la información de luz y color de la imagen. [13]

En dependencia de los objetivos de la organización, puede ser necesaria la manipulación de este recurso, y las herramientas necesarias se proponen en la misma.

9. Administración de servicios.

En esta categoría, se presenta algunas de las herramientas necesarias para sustentar todo lo relacionado con las prestaciones de servicios telemáticos, garantizando el correcto funcionamiento de la empresa o entidad



Capítulo 2: Características del Sistema propuesto: Observatorio de Tecnologías Libres.

2.1 Introducción.

En este capítulo se analizará exhaustivamente el funcionamiento del Observatorio. Esto permitirá un mejor entendimiento del entorno organizacional que se presenta, lo que dará las bases necesarias para una mejor comprensión de la propuesta de sistema. Debido al carácter difuso de los Flujos de información, la imposibilidad de determinar subsistemas, difícil establecimiento de reglas de funcionamiento, y a la poca estructuración de los procesos de negocio, es que se toma la decisión de plantear un modelo de dominio. Se presentan además los requerimientos funcionales y no funcionales obtenidos a partir de la opinión y análisis del equipo de desarrollo.

2.2 Descripción de planilla tecnológica.

La planilla tecnológica es el documento base del funcionamiento del Observatorio. La misma cuenta con una descripción general de una aplicación en específico. La capa más importante de información, son las aplicaciones, que están agrupadas en categorías y subcategorías definidas por el Observatorio. La información referente a cada una de las aplicaciones, es lo que se almacena en la planilla tecnológica. La planilla se desglosa en un grupo de parámetros que serán el punto de partida para la posterior comparación.

2.3 Modelo del dominio.

Debido al bajo nivel de estructuración que presenta el negocio que se está estudiando y que está altamente centrado en tecnologías informáticas, se propone un modelo del dominio, ya que permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, clientes, desarrolladores e interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema. Para capturar satisfactoriamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.



Se realiza la descripción del modelo de dominio mediante un diagrama de clases UML donde se especifican las principales clases conceptuales que pueden intervenir en el sistema, estos representarán los objetos que existen o eventos que suceden en el entorno en el que trabajará el mismo. Se procederá a explicar los principales conceptos que se identifican en este modelo.

Departamento de asesoría: esta clase se refiere al departamento de la Universidad de las Ciencias Informáticas, donde residen las informaciones y el personal al frente de la migración, que en este caso los que se denominan responsables.

Responsable: esta clase agrupa a todas las personas que forman parte del departamento de asesoría. Las principales responsabilidades de estas personas consisten en gestionar las informaciones y sustentar los servicios.

Tecnologías: esta clase se refiere a una categoría dentro del observatorio, una carpeta donde aparece la clasificación global de donde se desglosan todas las aplicaciones de las que se tiene información.

Subcategorías: esta clase se refiere a categorías que pueden o no aparecer dentro de la categoría de tecnología. O sea, lo que se pudiera llamar sub-tecnología. Ejemplo: Dentro de la tecnología Base de datos, está la subcategoría Gestor de Base de Datos.

Aplicaciones: en esta clase están las aplicaciones como tal. O sea, lo que compone a las tecnologías y a las subcategorías. Una aplicación tiene una descripción y una planilla tecnológica, donde aparecen todos los datos de la misma. Ejemplo: Dentro de la Tecnología SO (Sistema Operativo), está la aplicación Debian.

Planilla: una planilla representa el documento donde se almacenan las características y datos referentes a una aplicación en específico. En cada planilla aparece una descripción de la aplicación y los parámetros de los que se tiene información.

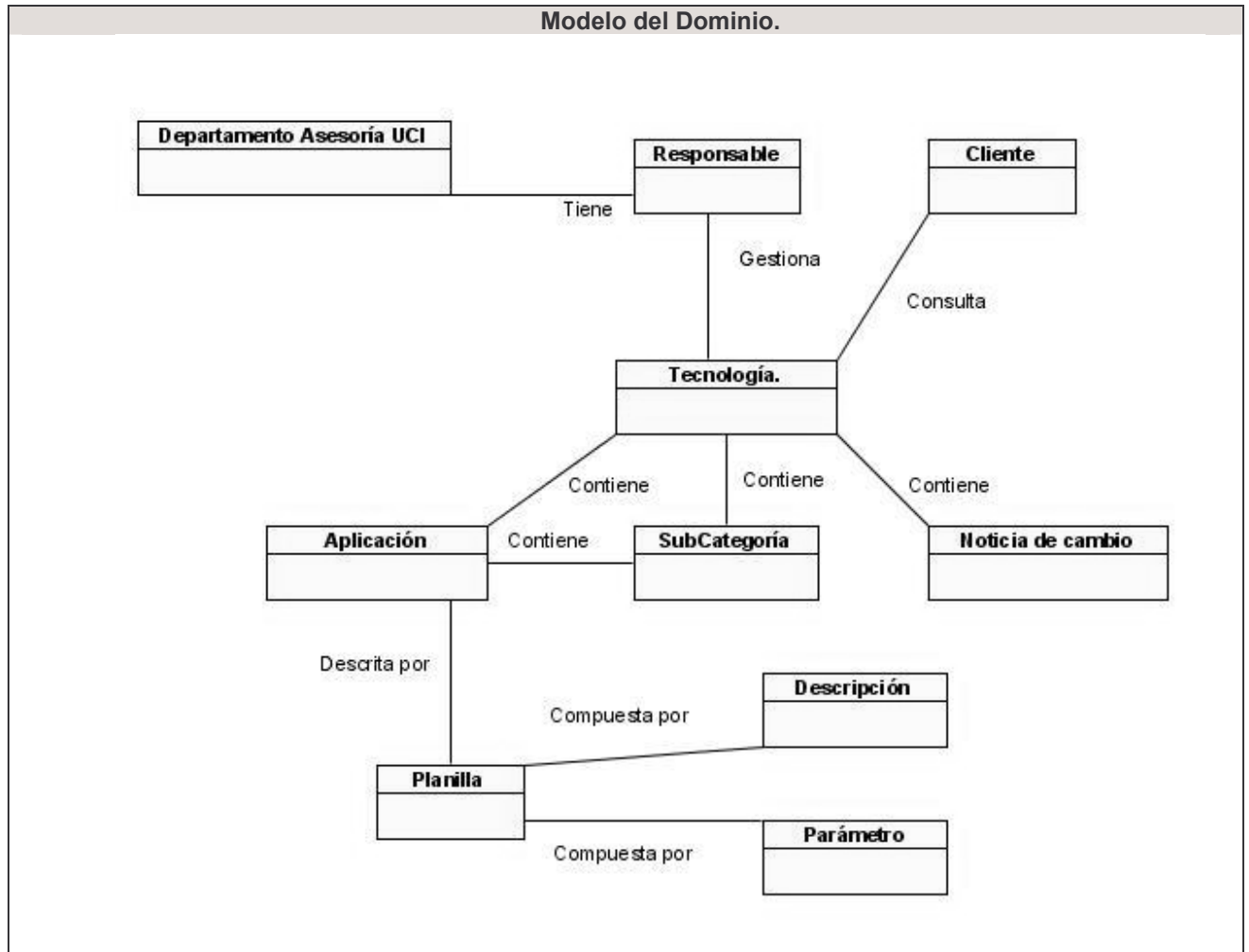
Parámetro: es una característica de la aplicación. Es definido comúnmente para un grupo de aplicaciones relacionadas.



Descripción: es la explicación formal de la aplicación, que ofrece una panorámica general de la misma, sin caer en las especificidades representadas por los parámetros.

Cliente: esta clase se refiere a todos los que acceden al departamento de asesoría en busca de informaciones referentes a la migración.

Noticia de cambio: son todas las informaciones de los cambios que ocurran en el Observatorio de Tecnologías Libres. Cada actualización, eliminación o adición de información, constituirá una noticia.



2.4 Propuesta de Sistema.

El Observatorio de Tecnologías Libres estará implementado de forma colaborativa y escalable. Será fundamentalmente una base sobre la cual un conjunto de usuarios bien definidos, se interrelacionarán, compartirán, publicarán y actualizarán información de forma controlada y sostenida. En el funcionamiento global del sistema, que se refiere a la asesoría de las entidades, se pueden desglosar cuatro procesos fundamentales, que abarcan las principales acciones a realizar. Estos son: Gestión de la información de las Tecnologías, Comparación de las tecnologías, Solicitud



de asesoría personalizada y Definición de usuarios y permisos. A continuación describiremos cada uno de estos procesos.

2.4.1 Gestionar Información de Tecnologías.

Uno de los procesos básicos de este sistema, es la gestión de la información. El sistema cuenta con una plantilla (planilla tecnológica) base, que documenta las especificaciones de cada aplicación. Estas son los parámetros sobre los cuales se va a realizar las comparaciones, y la información que se va a mostrar.

El usuario accede al sistema mediante una página principal desde donde puede:

1. Visualizar información seleccionada.
2. Acceder a los servicios de comparación de aplicaciones.
3. Registrarse como usuario del sitio.
4. En caso de ser colaborador:
 - Actualizar una planilla.
 - Eliminar una planilla.
 - Crear una planilla.
5. En caso de ser mantenedores.
 - Actualizar una planilla.
 - Eliminar una planilla.
 - Crear una planilla.
6. En caso de ser visitantes:
 - Visualizar noticia.
 - Visualizar páginas informativas.
 - Registrarse como usuario.
7. En caso de ser usuarios:
 - Acceder a servicios de asesoría. (Comparación y/o selección de aplicaciones por tecnología).
 - Visualizar noticia.
 - Visualizar páginas informativas.
 - Solicitar asesoría legal.



Para visualizar la información, el usuario solo debe utilizar las opciones mostrar del sitio. Solicita la información que desee (noticias, páginas informativas), y el sistema muestra los resultados deseados.

Para acceder a los servicios de comparación o solicitud de aplicaciones, el usuario debe estar previamente registrado como usuario o cliente del sistema. Solo así podrá acceder a estos servicios. Una vez en la página de asesoría tecnológica, el usuario puede:

1. Visualizar la clasificación global de tecnologías, permitiendo la navegación dentro de las categorías, subcategorías y aplicaciones, mostrando toda la información de las mismas o de una en específico.
2. Comparar aplicaciones, que es otro proceso general que será explicado más adelante.

En caso de los usuarios definidos como colaboradores una vez registrados en el sistema, tienen acceso a las planillas tecnológicas. Estas planillas comprenden la información referente a cada aplicación, los parámetros y características que los definen. Estos colaboradores pueden crear una planilla, cada planilla pertenece a una aplicación. Por tanto, al crear una planilla crean una nueva aplicación dentro de la clasificación existente. Los colaboradores además, tienen acceso a otras planillas, las cuales podrán completar o modificar. Cada modificación será registrada y atribuida al colaborador responsable de la misma.

2.4.2 Comparar las aplicaciones.

Una vez registrados los usuarios, como se ha explicado ya, tendrán acceso a la página de asesoría. En dicha página, los usuarios tienen tres opciones principales:

Comparar aplicaciones seleccionando o no parámetros específicos. En esta opción, el usuario accede a la lista de aplicaciones de una categoría o subcategoría, tiene la posibilidad de seleccionar hasta 5 aplicaciones para ser comparadas. Además, el usuario tiene la posibilidad de seleccionar los parámetros que le interesa, o puede seleccionarlos todos. Al presionar el botón comparar, el sistema mostrará el resultado en base a lo seleccionado.

Solicitar asesoría personalizada: mediante un formulario, envían los datos de la solicitud, que el proyecto puede aceptar o no, y recibe un mensaje informando si su solicitud fue o no aceptada.



Visualizar información sobre una aplicación específica. O sea, el usuario puede seleccionar una aplicación y obtener las informaciones referentes a la misma.

2.4.3 Definición de Usuarios y permisos.

Para controlar el flujo de información en el sitio, es necesario determinar los permisos que tendrá cada usuario en el mismo. El responsable de definir las categorías de usuarios será el administrador del sistema.

Tipos de usuarios posibles:

Administrador: Este es el usuario supremo, quien define a todos los demás tipos de usuarios. Este usuario posee los permisos necesarios para eliminar o adicionar una tecnología de la clasificación global propuesta.

Colaborador: En la escala de jerarquía, es el usuario que sigue al administrador. Este usuario puede adicionar, eliminar y actualizar la planilla tecnológica de cualquier aplicación. Para eliminar una planilla, el colaborador propone mediante el sistema la eliminación, pero hasta que el resto de los colaboradores no acepten la propuesta, la misma no podrá ser eliminada. Los colaboradores aceptan o no las solicitudes de asesoría legal personalizada.

Mantenedor: Estos usuarios actualizan la planilla tecnológica de la aplicación que les ha sido asignada. Pueden atender una o más aplicaciones en dependencia de los permisos asignados. Otra de las responsabilidades del mantenedor es adicionar, modificar y/o eliminar parámetros de la aplicación en cuestión.

Cliente: El cliente es el usuario que accede a los servicios de asesoría que brinda el sistema. Para poder ser cliente del Observatorio Tecnológico, el requisito fundamental es estar registrado en el mismo. Una vez registrado, podrá utilizar los servicios de asesoría brindados, y además, podrá solicitar asesoría tecnológica personalizada.



Anónimo: El anónimo es el usuario que solo puede acceder a las páginas informativas del sitio. O sea, aquel que no se ha registrado, por tanto no tendrá acceso a los servicios de asesoría.

2.4.4 Solicitud de asesoría personalizada.

Este es un servicio disponible solamente para los clientes del sistema. Consiste en un formulario donde el cliente debe completar un conjunto de datos para realizar una solicitud a la dirección del proyecto de obtener una asesoría legal personalizada por parte del mismo. El proyecto se reserva el derecho de admisión, asegurando que el usuario será informado vía e-mail del estado de su solicitud (aceptado o rechazado). A diferencia de los demás clientes, los que obtengan la asesoría personalizada, tendrán otros tipos de atenciones, y al final del proceso, recibirán una planilla impresa con los resultados de la asesoría.

2.5 Especificación de los requisitos de Software.

Una vez conocidos los conceptos que rodean al objeto de estudio, es preciso analizar: ¿Qué debe hacer el sistema para que se cumplan los objetivos trazados?, para ello se enumeran, a través de requerimientos funcionales, las acciones que el sistema deberá ser capaz de realizar. Dentro de ellos se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema, y las condiciones extremas a determinar por el sistema. De acuerdo con los objetivos planteados el sistema debe ser capaz de:

2.5.1 Requerimientos Funcionales.

R1. Gestionar tecnología:

- 1.1 Crear tecnología.
- 1.2 Eliminar tecnología.

R2. Generar noticia.

R3. Enviar mensaje personalizado de cambio vía e-mail.



R4. Visualizar informaciones.

4.1 Visualizar información sobre funcionamiento del observatorio.

4.2 Visualizar listado de noticias.

4.3 Visualizar una noticia.

R5. Visualizar los servicios del observatorio.

5.1 Visualizar menú con los servicios.

5.2 Visualizar servicio Registrarse como usuario.

5.3 Visualizar servicio Comparar aplicaciones.

5.4 Visualizar servicio Solicitar Asesoría legal personalizada.

R6. Comparar las aplicaciones.

R7. Administrar usuarios y roles de usuarios para la asignación de permisos.

7.1 Crear usuario. (Definir a que se refiere la información).

7.2 Crear rol de usuario. (Administrador, Mantenedor, Colaborador, Cliente, anónimo)

7.3 Asignar rol a usuario.

R8. Autenticarse.

R9. Buscar información en el sitio.

R10. Modificar datos de usuario.



R11. Visualizar información sobre tecnologías y aplicaciones.

11.1 Visualizar menú con la Clasificación global de las tecnologías.

11.2 Visualizar subcategorías por tecnología.

11.3 Visualizar listado de aplicaciones.

11.4 Visualizar información de una aplicación.

R12. Solicitar asesoría personalizada.

12.1 Mostrar formulario de solicitud.

12.2 Enviar solicitud.

R13. Enviar mensaje con estado de solicitud. (Rechazada o aceptada).

R14. Gestionar solicitud.

14.1 Mostrar solicitudes.

14.2 Rechazar solicitud.

14.3 Aceptar solicitud.

R15. Gestionar aplicación.

15.1 Crear aplicación.

15.2 Eliminar aplicación.

R16. Actualizar aplicación.



R17. Registrar usuario.

R18. Modificar reporte de resultados de comparación manualmente.

R19. Imprimir reporte de resultados de comparación.

R20. Darse de baja del servicio de notificaciones vía e-mail.

2.5.2 Requerimientos no funcionales.

Confidencialidad.

Solo se permitirá conexiones seguras mediante usuarios del directorio activo, para lograr la privatización de las preferencias de cada usuario.

Software.

Para la instalación del servidor:

- 1- Servidor Apache versión 2 o superior.
- 2- PHP 4 con módulo GD para el tratamiento de imágenes.
- 3- Servidor de bases de datos MYSQL versión 4 o superior.

Para la interpretación por clientes:

Navegador Web Internet Explorer versión 6 o Navegador Web Firefox o cualquier navegador Libre.

Hardware.

Mínimo Requerido: RAM 128 Mb, Hard Disk 512 Mb.

Mínimo Recomendado: RAM 256 Mb, Hard Disk 1 Gb.



Usabilidad.

Esta interfaz será Web, permitiendo un mejor uso e interacción con los usuarios, y mayor facilidad de intercambio de información en conexiones simultáneas.

Diseño e implementación.

El sistema funcionará sobre una aplicación desarrollada utilizando software libre, y filosofía cliente - servidor.

Como gestor de Base Datos se utilizará Mysql, se desarrollará sobre un sistema GNU/Linux, utilizándose herramientas como el Quanta, Bluefish, Phpmyadmin, Visual Paradigm.

Portabilidad:

El sistema será multiplataformas.

Seguridad:

Identificar al usuario antes de que pueda realizar cualquier acción sobre el contenido del portal.

Garantizar que la información sea editada únicamente por quien tiene derecho a editarla.

Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que este activo.

Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.

Verificación sobre acciones irreversibles (eliminaciones).

Permitir que los usuarios solo se conecte desde una sola dirección física a la vez por usuario, la autenticación se realizara por el dominio con el servicio de un directorio activo **ldap**.

Político – Culturales:



Debido a que se trata de un producto para Cuba, el sistema debe ajustarse a las características político-culturales de este país.

Legales:

La plataforma escogida para el desarrollo de la aplicación, está basada en la licencia GNU/GPL.

Confiabilidad:

La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.

Funcionalidad:

Reducir al mínimo el tiempo en que carga el portal.

Guardar en caché páginas de contenido para agilizar la navegación del portal.

2.6 Definición de los casos de uso.

A continuación se describen los actores del sistema, que no son más que terceros fuera del sistema que interactúan con el mismo, y que en este caso representan un rol jugado por una o varias personas.

2.6.1 Definición de actores del sistema.

Actores	Justificación
Colaborador	Es el usuario que tiene la responsabilidad de agregar, actualizar, modificar y proponer eliminar cualquier aplicación que esté disponible en el observatorio. Este usuario es el encargado de atender las solicitudes de asesoría legal.
Administrador	Es el usuario que define al resto de los usuarios en el observatorio. Además, tiene la responsabilidad de adicionar o eliminar las



	clasificaciones o categorías globales de las tecnologías libres. Tiene control total sobre el sistema.
Mantenedor	Este usuario posee el control de modificar, adicionar o eliminar parámetros dentro de las aplicaciones.
Cliente	Es el usuario que tiene la posibilidad de solicitar asesoría personalizada.
Anónimo	Este usuario es cualquier navegante que accede sin registrarse al sistema. Solo podrá registrarse en el sistema, o sea, convertirse en un usuario autenticado.
Usuario Registrado.	Este usuario es el que puede autenticarse en el observatorio, puesto que ya existe. Además, puede utilizar los servicios de comparación de aplicaciones, puede modificar sus datos y visualizar los servicios.
Usuario 2.	Este usuario podrá visualizar las informaciones publicadas en el observatorio.



2.6.2 Definición de casos de uso del sistema.

Utilizando las facilidades que brinda el UML, se representarán los requisitos funcionales del sistema mediante un diagrama de casos de uso. Para ello hay que definir de acuerdo a lo planteado en los epígrafes anteriores, cuales serían los actores que van a interactuar con el sistema, y los casos de uso que van a representar las funcionalidades.

A continuación se presentan los casos de uso determinados para satisfacer los requerimientos funcionales del sistema:

Listado de casos de uso.

CUS-1	Autenticar usuario.
Actor	Usuario registrado.
Descripción	Un usuario accede al observatorio para utilizar los servicios del mismo, introduciendo usuario y contraseña.
Referencia	R8.

CUS-2	Administrar usuarios.
Actor	Administrador.
Descripción	El administrador accede al portal para editar la seguridad. (roles, usuarios y permisos)
Referencia	R7, CUS-1.

CUS-3	Comparar aplicaciones.
Actor	Usuario registrado.
Descripción	El usuario solicita comparar aplicaciones mediante los servicios brindados en el observatorio.
Referencia	R6, R11, CUS-1.

CUS-4	Gestionar tecnología.
Actor	Administrador.
Descripción	El administrador accede al observatorio para editar (eliminar o adicionar) una o más tecnologías.



Referencia	R1, CUS-1.
-------------------	------------

CUS-5	Visualizar información.
Actor	Usuario2.
Descripción	El usuario solicita ver las informaciones publicadas.
Referencia	R4, R9.

CUS-6	Visualizar servicios.
Actor	Usuario registrado.
Descripción	El usuario solicita ver los servicios que brinda el observatorio.
Referencia	R5, CUS-1.

CUS-7	Modificar datos de usuario.
Actor	Usuario registrado.
Descripción	El usuario solicita modificar los datos de su perfil.
Referencia	R10, R20, CUS-1.

CUS-8	Solicitar asesoría.
Actor	Cliente.
Descripción	El cliente solicita asesoría legal personalizada mediante el formulario correspondiente.
Referencia	R12, CUS-1.

CUS-9	Gestionar Solicitud.
Actor	Colaborador.
Descripción	El colaborador accede al sistema para aceptar o denegar las solicitudes de asesoría.
Referencia	R13, R14, CUS-1, CUS-14, CUS-13.

CUS-10	Gestionar aplicación.
Actor	Colaborador.
Descripción	El colaborador accede al sistema para adicionar o eliminar una aplicación.



Referencia	R15, CUS-1, CUS-14, CUS-13.
-------------------	-----------------------------

CUS-11	Actualizar aplicación.
Actor	Mantenedor.
Descripción	El mantenedor accede al sistema para actualizar los datos de una aplicación.
Referencia	R16, CUS-1, CUS-14, CUS-13.

CUS-12	Registrar usuario.
Actor	Anónimo.
Descripción	El usuario anónimo accede al sistema para registrarse como miembro del mismo.
Referencia	R17.

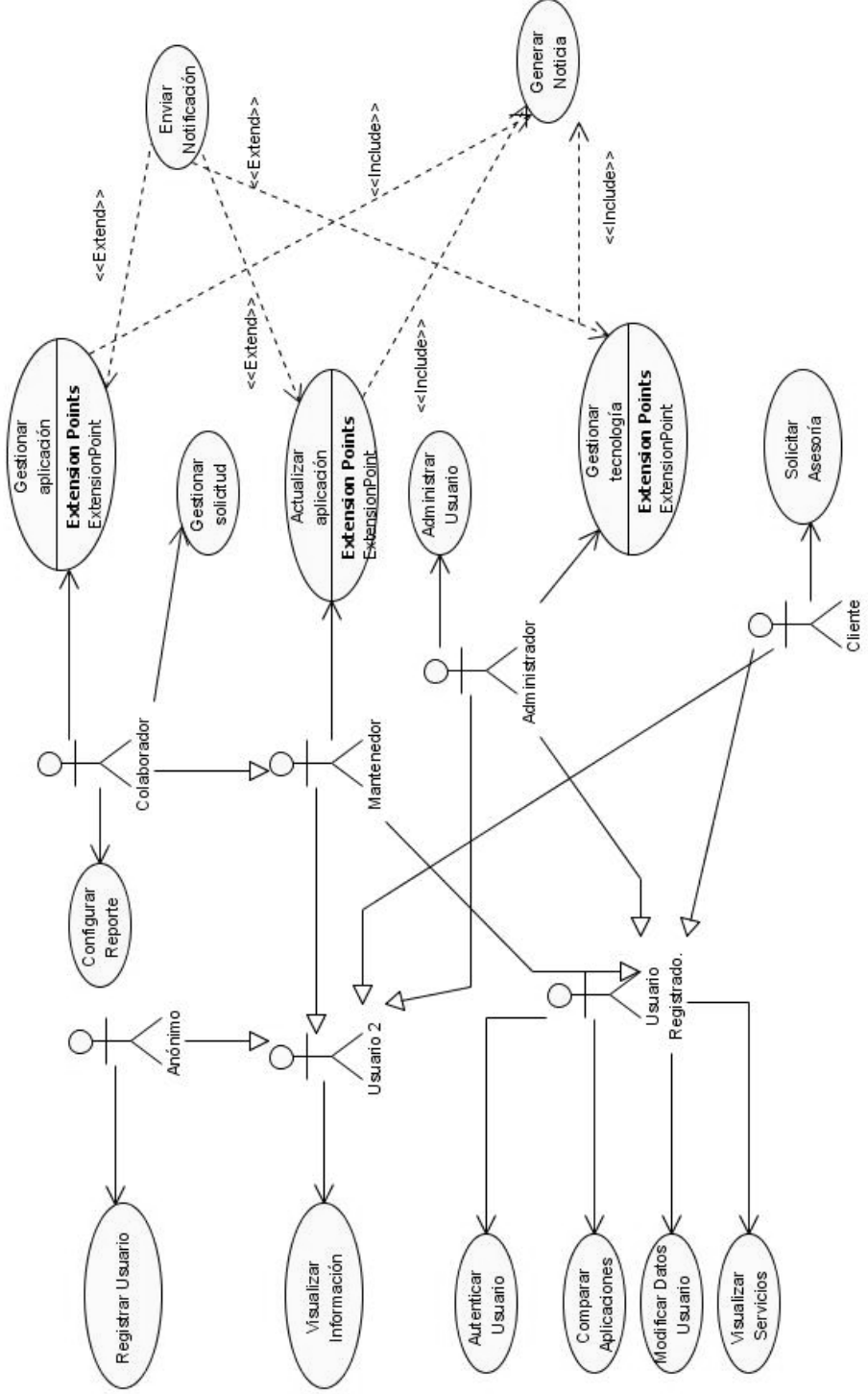
CUS-13	Generar noticia.
Actor	
Descripción	Al ocurrir algún cambio tanto en las tecnologías como en las aplicaciones, el sistema generará automáticamente una noticia de formato estándar en la página de noticia.
Referencia	R2, CUS-10, CUS-4, CUS-11.

CUS-14	Enviar notificación.
Actor	
Descripción	Al ocurrir algún cambio tanto en las tecnologías como en las aplicaciones, el sistema generará automáticamente un mensaje electrónico de formato estándar que será enviado a los usuarios del observatorio suscritos al servicio de notificaciones online.
Referencia	R3, CUS-10, CUS-4, CUS-11.

CUS-15	Configurar reporte.
Actor	Colaborador.
Descripción	Una vez generado el reporte automático, el colaborador podrá modificar el mismo de acuerdo a sus conocimientos o necesidades reales de la empresa. Además este informe se podrá imprimir.
Referencia	R19, R20, CUS-1.



DIAGRAMA DE CASOS DE USO





2.6.3 Descripción de los Casos de Uso del Sistema.

En este subepígrafe se presenta la descripción detallada de los casos de uso.

Caso de uso	
CUS-1	Autenticar usuario.
Propósito	Un usuario se autentica con los permisos correspondientes.
Actores: Usuario registrado.	
Resumen: El caso de uso se inicia cuando un usuario registrado accede al sistema y este solicita usuario y contraseña. El sistema concede al usuario un permiso para poder acceder a las demás funcionalidades del sistema, verificando la validez de los datos.	
Referencias	R8
Acción del actor	Respuesta del sistema
1. Accede a la aplicación.	2. Muestra el formulario con los datos a introducir.
3. Introduce los datos. (Usuario y contraseña)	4. Verifica autenticidad de los datos. 5. Muestra página de inicio de los usuarios autenticados.
Flujo alternativo	
Acción del actor	Respuesta del sistema
3. Introduce los datos. (Usuario y contraseña)	4.1 No son válidos los datos. 4.2 Muestra mensaje de error. Ir al paso 2.
Prioridad	Secundario
Puntos de extensión.	

Caso de uso	
CUS-2	Administrar usuarios.
Propósito	El administrador establece roles de usuarios y permisos pertinentes.
Actores: Administrador.	
Resumen: El caso de uso inicia cuando el administrador accede al módulo de administración para asignar los roles a los usuarios correspondientes. El sistema mostrará una página donde aparecen los usuarios registrados en el sitio. El administrador podrá seleccionar el usuario a modificar, una vez seleccionado, se muestran los roles a ser seleccionados, y una vez seleccionados estos, los permisos correspondientes al rol específico.	
Referencias	R7, CUS-1.
Acción del actor	Respuesta del sistema
1. Accede a la página de administración de usuarios.	2. Muestra listado de los usuarios.



3. Selecciona usuario a modificar.	4. Muestra roles posibles a asignar.
5. Selecciona rol.	6. Muestra permisos relativos al rol posibles a asignar.
7. Selecciona permisos y clikea el botón aceptar.	8. Actualiza la información referente al cambio de usuario.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Prioridad	Secundario
Puntos de extensión.	

Caso de uso	
CUS-3	Comparar aplicaciones.
Propósito	El usuario compara a lo sumo cinco aplicaciones definiendo o no parámetros de comparación específicos.
Actores Usuario Registrado.	
Resumen: Este caso de uso responde a uno de los objetivos principales del observatorio, y es que los usuarios del mismo tengan la posibilidad no solo de acceder a la información, sino en base a los parámetros predefinidos en la planilla tecnológica, obtener los resultados necesarios para sus condiciones específicas. Se inicia cuando el usuario registrado solicita comparar aplicaciones (hasta cinco), seleccionando parámetros específicos o no. Además, puede seleccionar una aplicación para acceder a la información de la misma. El sistema devuelve los resultados de la comparación.	
Referencias	R6, R11, CUS-1
Acción del actor	Respuesta del sistema
1. Solicita comparar aplicaciones.	2. Muestra Listado de tecnologías. 3. Muestra listado de subcategorías y aplicaciones. 4. Muestra parámetros de comparación.
5. Selecciona aplicaciones a comparar. (Hasta cinco aplicaciones) 6. Selecciona parámetros de comparación. 7. Acciona el botón aceptar.	8. Muestra resultados de la comparación.
Flujo alternativo	
Acción del actor	Respuesta del sistema
5. Selecciona una aplicación y clikea aceptar.	6. Muestra información de la aplicación.



Prioridad	Crítico
Puntos de extensión.	

Caso de uso	
CUS-4	Gestionar tecnología.
Propósito	Un administrador adiciona o elimina una tecnología.
Actores Administrador	
Resumen: El caso de uso se inicia cuando el administrador accede al servicio de gestionar tecnología del sistema y este muestra el listado de las tecnologías existentes con la posibilidad de marcar una y eliminarla, o agregar otra. Al concluir queda actualizada la información.	
Referencias	R1, CUS-1
Acción del actor	Respuesta del sistema
1. Solicita eliminar tecnologías.	2. Muestra listado de tecnologías.
3. Selecciona tecnología y acciona el botón eliminar.	4. Elimina tecnología. 5. Genera noticia de cambio. (Ver CUS-13). 6. Envía mensajes de notificación de cambio. (Ver CUS-14)
Flujo alternativo	
Acción del actor	Respuesta del sistema
1. Solicita adicionar tecnología.	2. Muestra formulario para adicionar tecnología.
3. Introduce datos necesarios y acciona enviar.	4. Registra los datos en la aplicación. Ir al paso 5.
Prioridad	Crítico
Puntos de extensión.	

Caso de uso	
CUS-5	Visualizar información.
Propósito	Un usuario anónimo accede a la aplicación. El sistema visualiza las informaciones que aparecen en el observatorio.
Actores: Anónimo.	
Resumen: Se inicia cuando el usuario anónimo accede al observatorio, y se le muestra la página de inicio del mismo, con las informaciones referentes al funcionamiento de este.	
Referencias	R4, R9
Acción del actor	Respuesta del sistema



1. Accede a la aplicación.	2. Muestra página de inicio con informaciones.
Flujo alternativo	
Acción del actor	Respuesta del sistema
1. Accede a la aplicación y solicita buscar información.	2. Muestra buscador.
3. Introduce datos a buscar.	4. Realiza búsqueda. 5. Muestra resultados.
Prioridad	Secundario
Puntos de extensión.	

Caso de uso	
CUS-6	Visualizar servicios.
Propósito	Un usuario registrado accede a la página de los servicios.
Actores: Usuario registrado	
Resumen: Se inicia cuando el usuario registrado, una vez autenticado, accede a la página de servicios, donde se le muestra un menú con los servicios a los que tiene acceso como miembro del observatorio.	
Referencias	R5, CUS-1
Acción del actor	Respuesta del sistema
1. Accede a la aplicación.	2. Muestra página de inicio. (Ver CUS-5).
3. Solicita ver servicios.	4. Muestra página con los servicios del observatorio.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Prioridad	Secundario
Puntos de extensión.	

Caso de uso	
CUS-7	Modificar Datos Usuario.
Propósito	Un usuario registrado actualiza su perfil de usuario.
Actores: Usuario Registrado.	
Resumen: Se inicia cuando el usuario solicita modificar perfil, una vez autenticado. El sistema mostrará el perfil del usuario con los datos anteriores permitiéndole a este cambiar los datos existentes y guardar los cambios.	



Referencias	R10, CUS-1	
Acción del actor		Respuesta del sistema
1. Solicita servicio configurar perfil.		2. Muestra datos existentes.
3. Modifica datos y acciona aceptar.		4. Verifica validez de información. 5. Actualiza información.
Flujo alternativo		
Acción del actor		Respuesta del sistema
3. Solicita darse baja de servicio de notificaciones vía e-mail.		4. Actualiza datos.
Prioridad		Secundario
Puntos de extensión.		

Caso de uso	
CUS-8	Solicitar asesoría.
Propósito	Un cliente solicita atención personalizada a los miembros del proyecto mediante el observatorio.
Actores: Cliente	
Resumen: Se inicia cuando el cliente accede al sistema, una vez autenticado, y solicita asesoría legal. El sistema muestra un formulario con los datos necesarios para la solicitud, el usuario introduce los mismos y quedan almacenados en el listado de solicitudes.	
Referencias	R12, CUS-1
Acción del actor	Respuesta del sistema
1. El cliente hace la solicitud de asesoría personalizada.	2. Registra solicitud.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Prioridad	Crítico
Puntos de extensión.	

Caso de uso	
CUS-9	Gestionar solicitud.
Propósito	Un colaborador procesa las solicitudes de asesoría hechas a la dirección del proyecto.
Actores: Colaborador.	



Resumen: Se inicia cuando el colaborador accede al servicio de gestionar solicitudes una vez autenticado. El sistema muestra el listado de solicitudes pendientes, que el usuario puede aprobar o rechazar. En caso de aprobar las solicitudes, estas se registran como asesorías en proceso, asignándosele fecha y colaborador que la atenderá. En caso de rechazar la solicitud, esta será borrada de la lista de solicitudes. En ambos casos, el sistema generará un mensaje de formato estándar para informar al usuario el estado de su solicitud. Aceptada con los respectivos datos o rechazada.	
Referencias	R13, R14, CUS-1
Acción del actor	Respuesta del sistema
1. Accede al servicio gestionar solicitud.	2. Muestra listado de solicitudes pendientes.
3. Selecciona solicitud.	4. Muestra opciones para editar solicitud.
5. Rechaza solicitud y cliquee aceptar.	6. Elimina solicitud. 7. Envía mensaje de notificación.
Flujo alternativo	
Acción del actor	Respuesta del sistema
6. Acepta solicitud.	7. Muestra formulario con los datos necesarios para aceptar la solicitud. (Fecha y Colaborador.)
8. Introduce datos y cliquee aceptar.	9. Comprueba validez de los datos. 10. Registra la solicitud. Ir al paso 7.
Prioridad	Crítico
Puntos de extensión.	

Caso de uso	
CUS-10	Gestionar aplicación.
Propósito	Un colaborador adiciona o elimina aplicaciones en el observatorio.
Actores: Colaborador	
Resumen: Se inicia cuando el colaborador accede al servicio de gestionar aplicación. Selecciona la categoría en la que va a trabajar y selecciona adicionar aplicación. El sistema muestra un formulario con todos los datos necesarios para adicionar la misma. Una vez concluida la introducción de los datos, el colaborador acepta y se registra la nueva aplicación. En caso de eliminar, el colaborador selecciona la aplicación y acciona la opción de eliminar. Automáticamente el sistema genera un formulario donde el colaborador debe introducir los motivos por los cuales se propone la eliminación. Este formulario será visible para todos los colaboradores, con la opción para cada uno de votar si o no al cambio. Será eliminada la aplicación solo si todos los colaboradores votan a favor del cambio, después que todos los colaboradores efectúen su voto, el sistema automáticamente eliminará la aplicación si todos los votos son positivos, y mantendrá la aplicación si alguno de los votos difiere. En caso de cambio el sistema genera una notificación y una noticia.	



Referencias	R5, CUS-1, CUS-13, CUS-14	
Acción del actor	Respuesta del sistema	
1. Accede al servicio de gestionar aplicación.	2. Muestra listados de tecnologías con sus aplicaciones.	
3. Selecciona aplicación a editar.	4. Muestra opción de eliminar	
5. Acciona opción de eliminar.	6. Genera formulario para votaciones.	
7. Introduce datos necesarios.	8. Almacena datos.	
9. Vota si. (Todos lo colaboradores).	10. Elimina aplicación. 11. Genera noticia de cambio. (Ver CUS-13). 12. Envía mensajes de notificación de cambio. (Ver CUS-14).	
Flujo alternativo 1.		
Acción del actor	Respuesta del sistema	
9. Vota no. (Al menos uno colaborador).	10. No elimina la aplicación.	
Flujo alternativo 2		
3. Selecciona adicionar aplicación.	4. Muestra formulario con datos necesarios para adicionar aplicación.	
5. Introduce datos necesarios.	6. Valida datos. 7. Almacena datos. (Ir al paso 11).	
Prioridad	Crítico	
Puntos de extensión.		

Caso de uso	
CUS-11	Actualizar aplicación.
Propósito	Un mantenedor elimina, adiciona o modifica parámetros de una aplicación.
Actores: Mantenedor.	
Resumen: Se inicia cuando el mantenedor, una vez autenticado, selecciona una aplicación y acciona la opción de actualizar. El sistema muestra el formulario con los datos previos de la aplicación, dando la posibilidad de cambiar los datos. El mantenedor acepta y los cambios son registrados en la base de datos. En caso de cambio, el sistema genera una notificación y una noticia.	
Referencias	R16, CUS-1, CUS-13, CUS-14
Acción del actor	Respuesta del sistema
1. Solicita actualizar aplicación.	2. Muestra listado de las tecnologías con sus aplicaciones.
3. Selecciona aplicación.	4. Muestra formulario para modificar aplicación.



5. Introduce nuevos datos y da enviar.	6. Registra nuevos datos. 7 Genera noticia de cambio. (Ver CUS-13). 8 Envía mensajes de notificación de cambio. (Ver CUS-14)
Flujo alternativo	
Acción del actor	Respuesta del sistema
Prioridad	Crítico
Puntos de extensión.	

Caso de uso	
CUS-12	Registrar usuario.
Propósito	Un anónimo se registra como usuario del Observatorio de tecnologías libres.
Actores: Anónimo.	
Resumen: Se inicia cuando un usuario accede al sitio y solicita registrarse en el mismo. El sistema muestra un formulario con los datos requeridos para la inscripción. El usuario introduce los datos y el sistema almacena los mismos.	
Referencias	R17
Acción del actor	Respuesta del sistema
1. Solicita registrarse como usuario.	2. Muestra formulario con datos necesarios para el registro.
3. Introduce datos necesarios y da enviar.	4. Verifica validez de los datos. 5. Registra nuevo usuario.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Prioridad	Secundario
Puntos de extensión.	

Caso de uso	
CUS-13	Generar noticia.
Propósito	Generar una noticia informando cambio.
Caso de uso incluido de: CUS-4, CUS-10, CUS-11	



Resumen: Un usuario autorizado realiza algún cambio en aplicaciones o tecnologías, cuando se guardan los cambios, automáticamente el sistema genera una noticia en la página de noticias, informando el cambio.	
Referencias	
Acción del actor	Respuesta del sistema
1. Se realizan cambios en: aplicaciones o tecnologías.	2. Genera noticia informando cambios.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Prioridad	Auxiliar
Puntos de extensión.	

Caso de uso	
CUS-14	Enviar notificación de cambio.
Propósito	Enviar un mensaje vía e-mail con la notificación correspondiente al cambio realizado.
Caso de uso extendido de: CUS-4, CUS-10, CUS-11	
Resumen: Un usuario autorizado realiza algún cambio en aplicaciones o tecnologías, cuando se guardan los cambios, automáticamente el sistema comprueba que usuarios seleccionaron la opción de notificar cambios, y envía un mensaje electrónico informando del cambio realizado.	
Referencias	
Acción del actor	Respuesta del sistema
1. Se realizan cambios en: aplicaciones o tecnologías.	2. Verifica usuarios que solicitaron notificación vía e-mail. 3. Envía mensaje a estos usuarios.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Prioridad	Opcional
Puntos de extensión.	

Caso de uso	
CUS-15	Modificar reporte.
Propósito	El colaborador modifica el reporte de la comparación.



Actores: Colaborador.	
Resumen: El caso de uso inicia cuando el colaborador accede al reporte de resultados para modificar los datos. El sistema muestra el reporte, el colaborador modifica los datos, guarda los cambios e imprime el reporte.	
Referencias	R19, R20, CUS-1
Acción del actor	Respuesta del sistema
1. Accede al reporte.	2. Muestra el reporte.
3. Selecciona modificar. 4. Modifica datos, y acciona aceptar.	5. Guarda los cambios.
6. Selecciona imprimir reporte.	7. Imprime reporte.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Prioridad	Secundario
Puntos de extensión.	



Capítulo 3: Análisis y diseño del sistema.

3.1 Introducción.

En este capítulo se propone traducir los requisitos a una especificación que describe cómo implementar el sistema. El análisis consiste en obtener una visión del sistema que permita entender qué se hace, de modo que sólo se interesa por los requisitos funcionales. Además se presenta el diseño, que se puede traducir como un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, o sea, cómo cumple el sistema sus objetivos.

3.2 Análisis.

Después de identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas, necesarias para la construcción del modelo de análisis, se procede a elaborar los diagramas de clases del análisis por cada caso de uso. Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada.

Para facilitar el análisis, las clases se clasifican de la siguiente forma:

Clase de control.

- ✓ Representan coordinación, secuencia, transacciones, y control de otros objetos.
- ✓ Se usan mucho para encapsular el control de un caso de uso concreto.
- ✓ También derivaciones y cálculos complejos.
- ✓ Manejan y coordinan las acciones y los flujos de control principales, y delegan trabajo a otros objetos (de interfaz y de entidad).

Clase de entidad.

- ✓ Modelan información que posee una larga vida.
- ✓ Modelan la información y el comportamiento asociado a algún fenómeno o concepto, como una persona, un objeto del mundo real, o un suceso del mundo real.

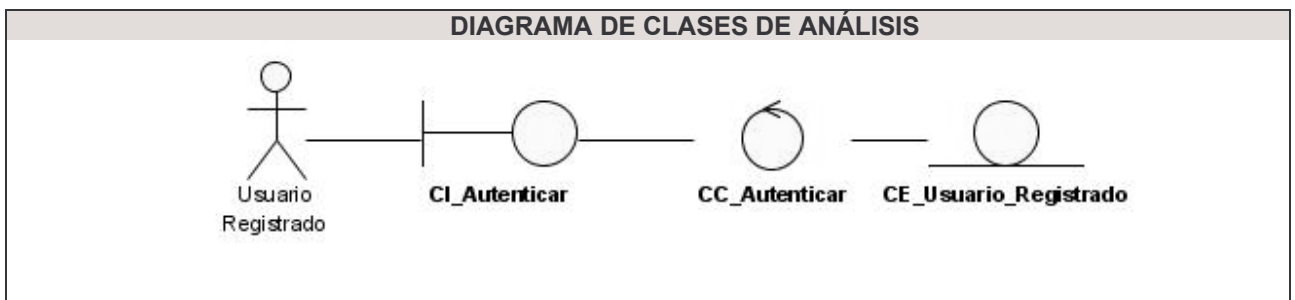


- ✓ Derivan normalmente de una clase de entidad del negocio.
- ✓ Puede tener comportamiento complejo.
- ✓ Aísla los cambios en la información que representa.

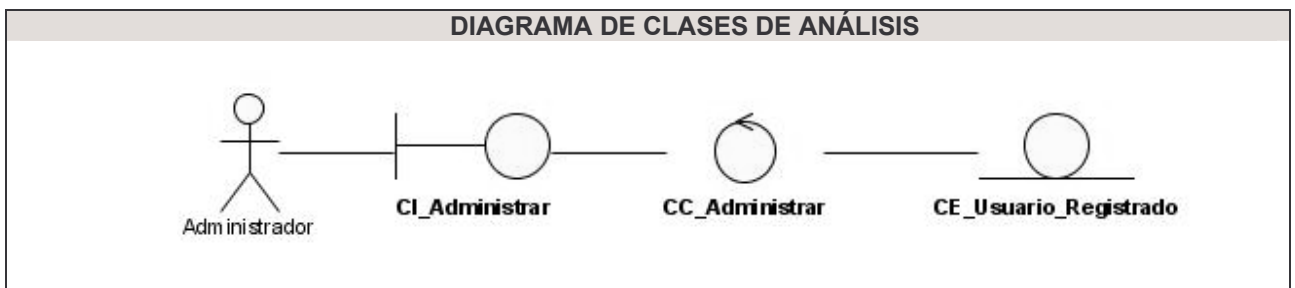
Clase de interfaz.

- ✓ Modelan la interacción entre el sistema y sus actores (usuarios y sistemas externos).
- ✓ Reciben y presentan información y peticiones de y hacia los actores.
- ✓ Reúnen los requisitos en los límites del sistema.
- ✓ Suelen ser abstracciones de ventanas, formularios, interfaces de impresoras, sensores, terminales.
- ✓ Describan lo que se obtiene con la interacción, no el proceso físico de cómo se ejecuta.
- ✓ Una clase de interfaz por cada actor, y viceversa.

3.2.1 Diagrama de clases del análisis de Caso de Uso: Autenticar usuario.

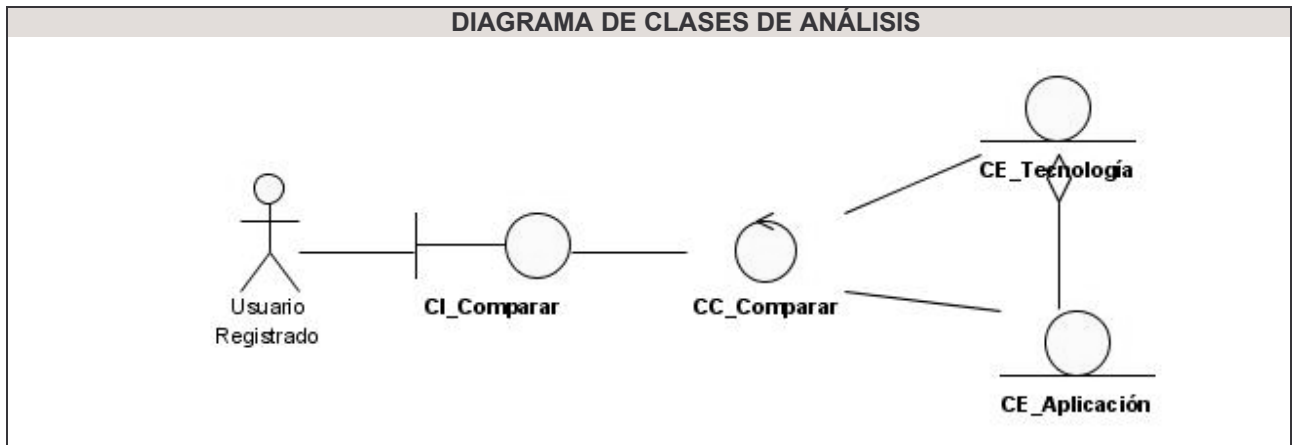


3.2.2 Diagrama de clases del análisis de Caso de Uso: Administrar usuario.

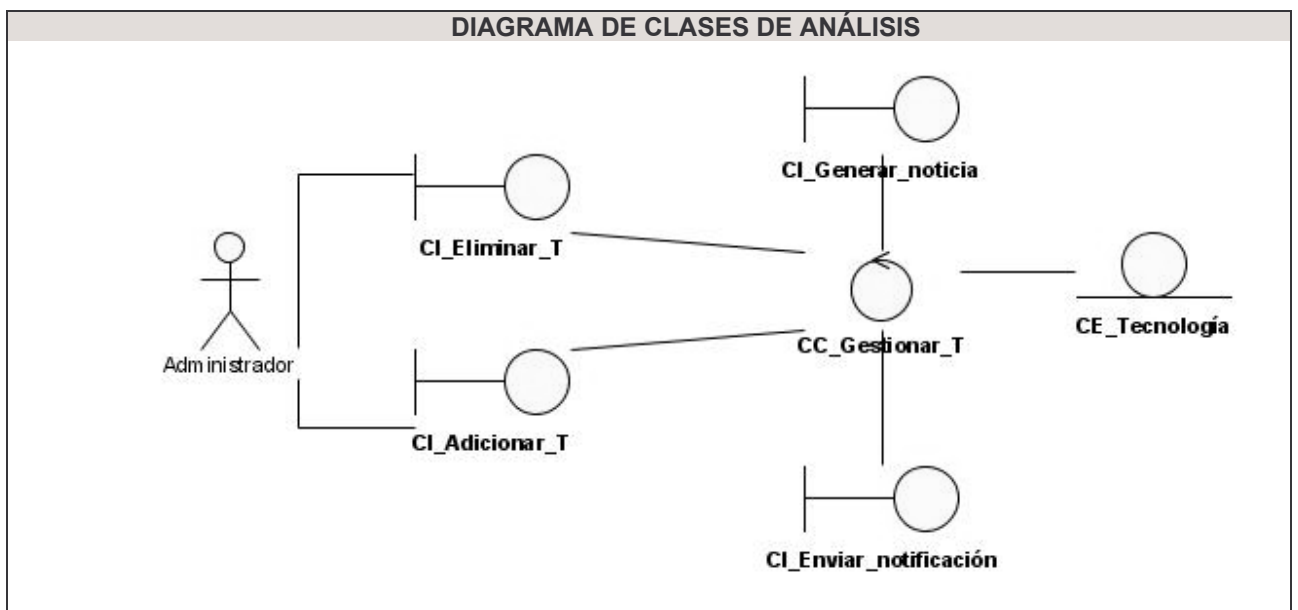




3.2.3 Diagrama de clases del análisis de Caso de Uso: Comparar aplicaciones.

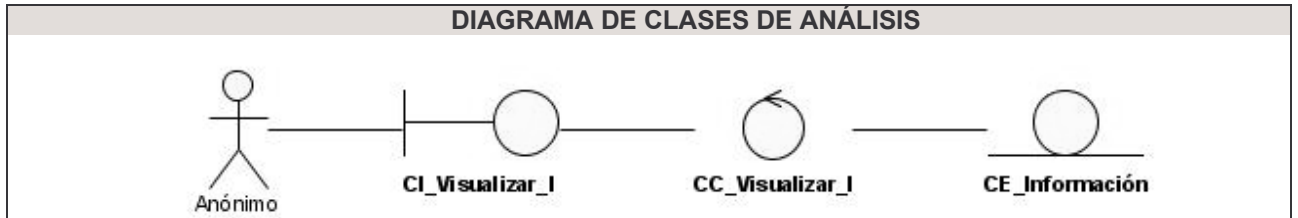


3.2.4 Diagrama de clases del análisis de Caso de Uso: Gestionar tecnología.

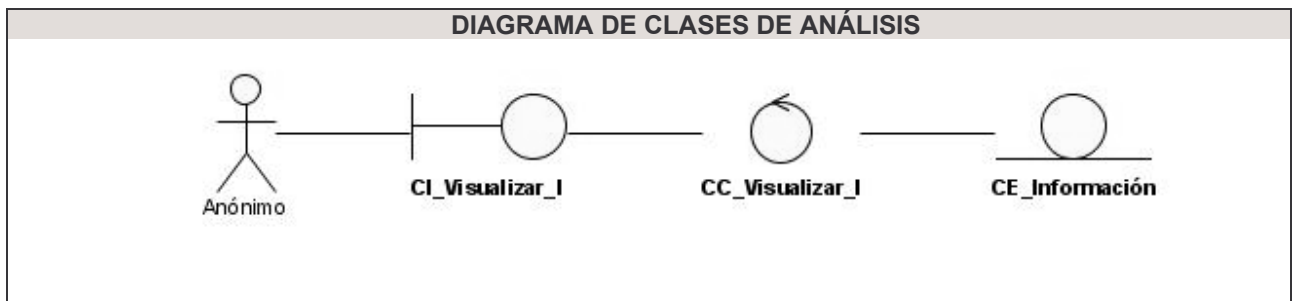




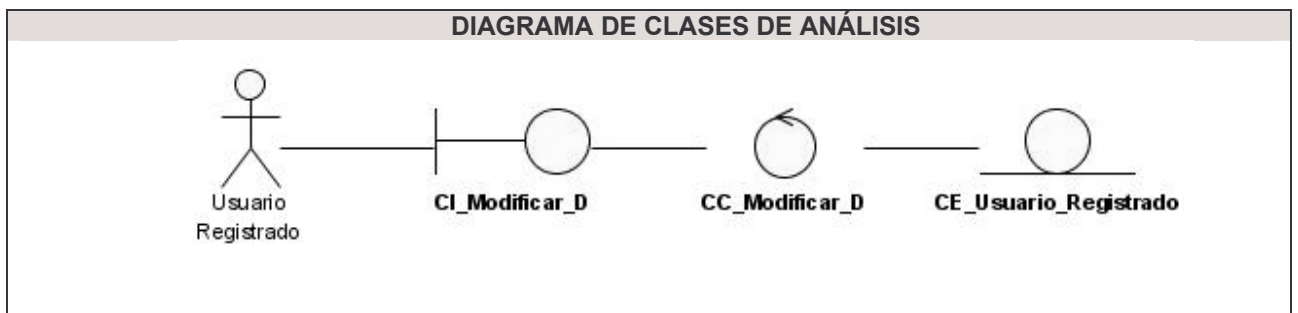
3.2.5 Diagrama de clases del análisis de Caso de Uso: Visualizar información.



3.2.6 Diagrama de clases del análisis de Caso de Uso: Visualizar servicios.

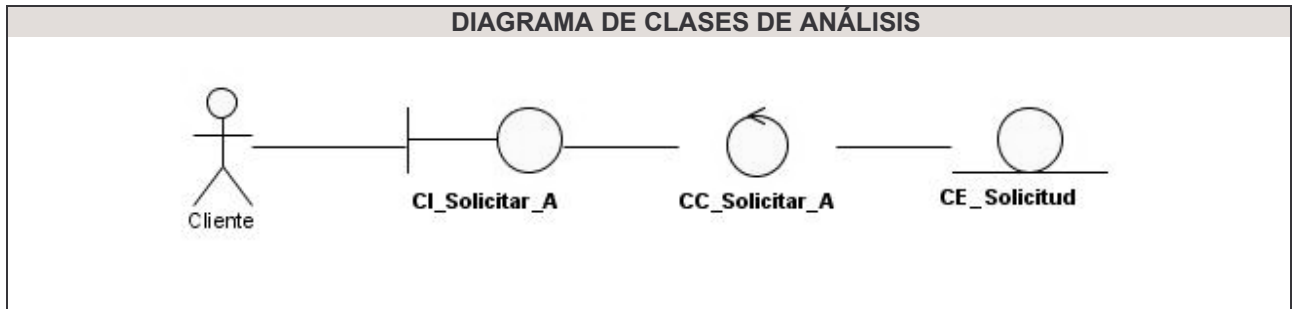


3.2.7 Diagrama de clases del análisis de Caso de Uso: Modificar datos usuario.

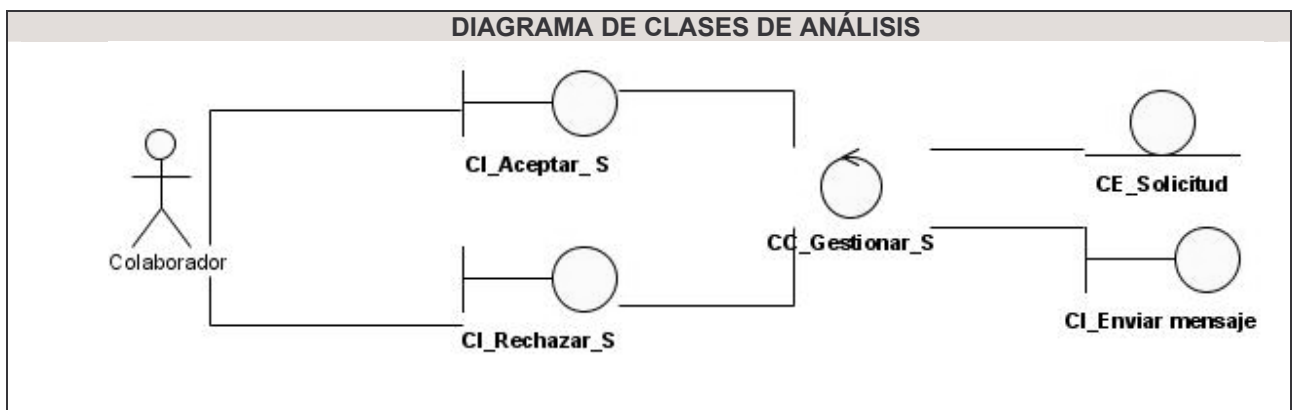




3.2.8 Diagrama de clases del análisis de Caso de Uso: Solicitar asesoría.

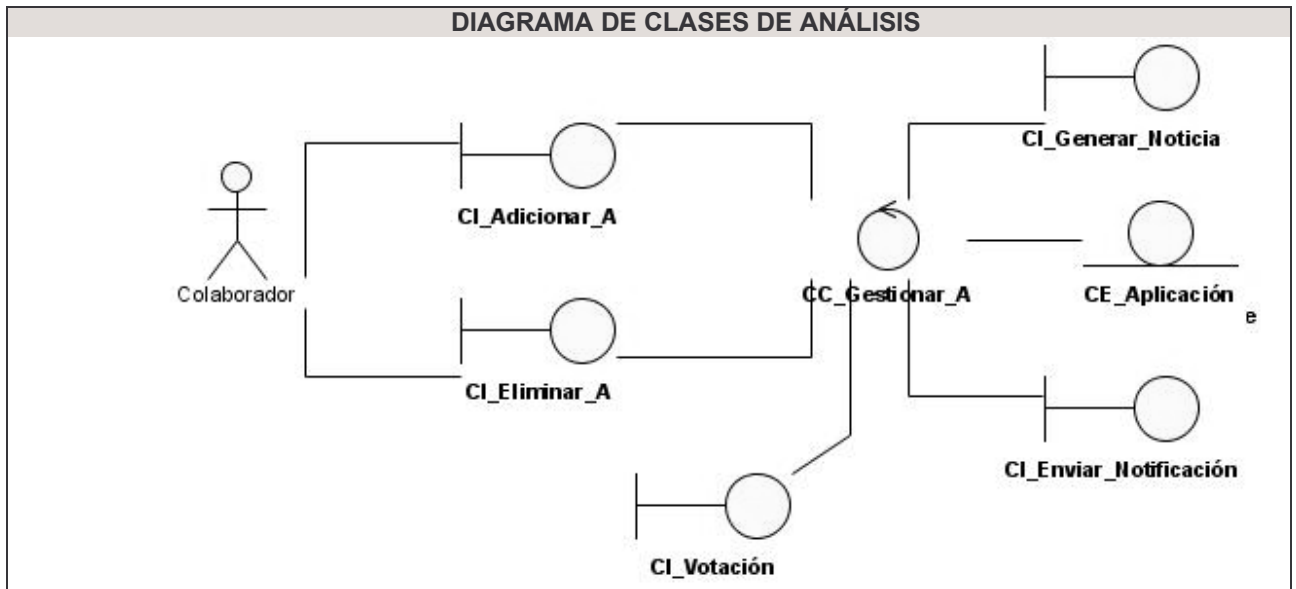


3.2.9 Diagrama de clases del análisis de Caso de Uso: Gestionar solicitud.

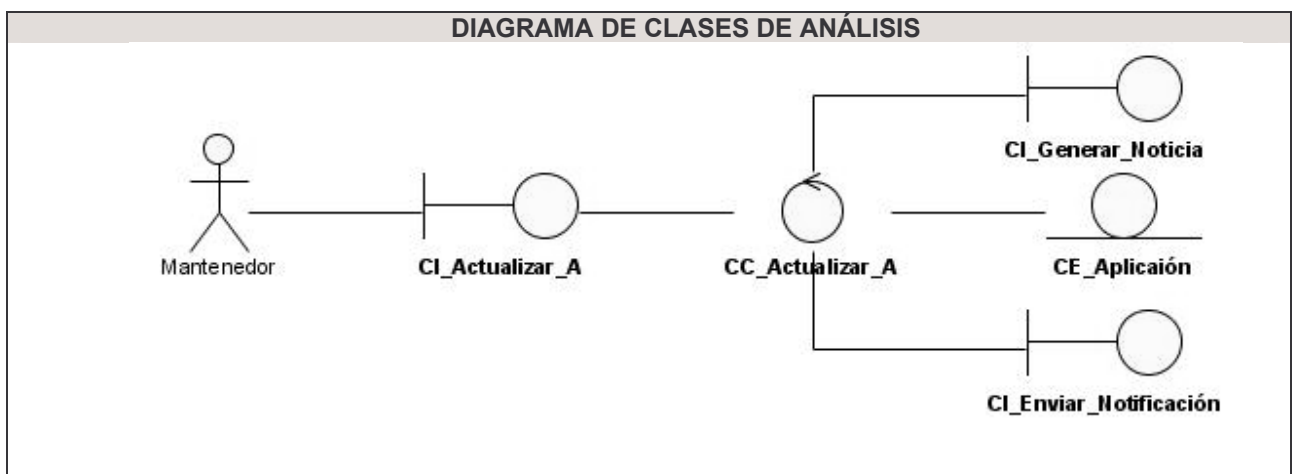




3.2.10 Diagrama de clases del análisis de Caso de Uso: Gestionar aplicación.

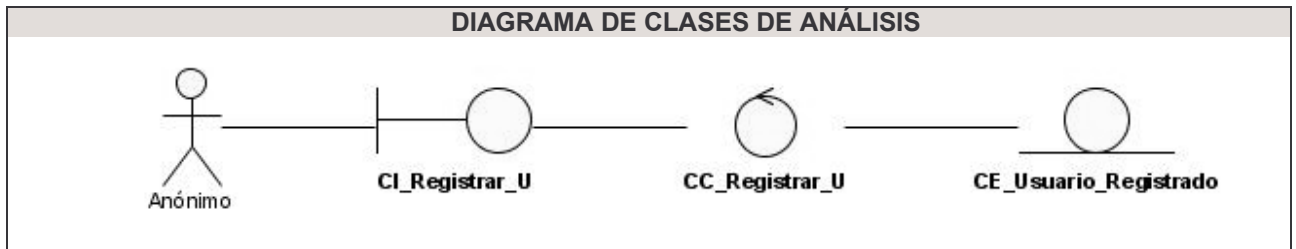


3.2.11 Diagrama de clases del análisis de Caso de Uso: Actualizar aplicación.

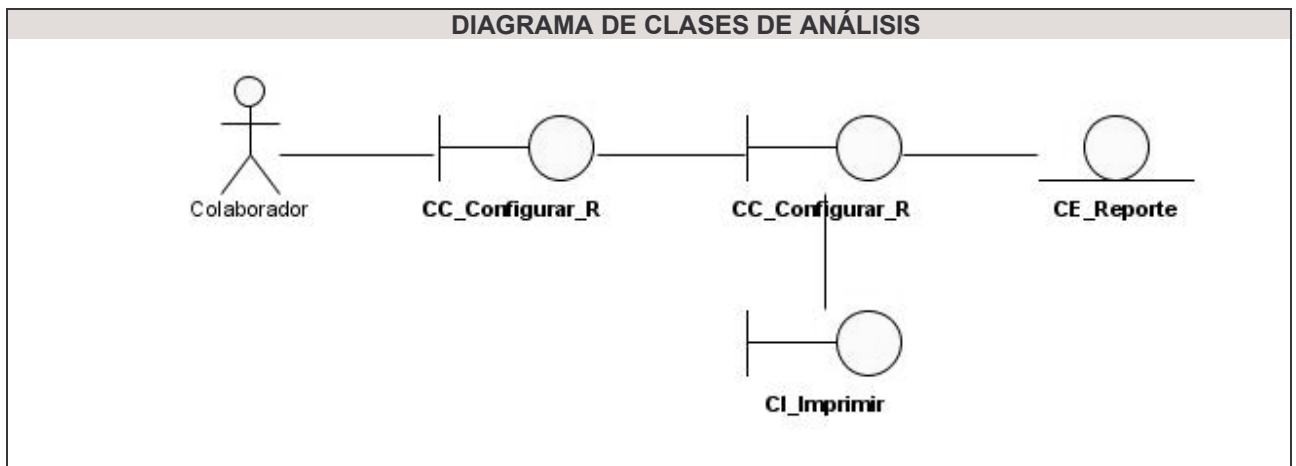




3.2.12 Diagrama de clases del análisis de Caso de Uso: Registrar Usuario.



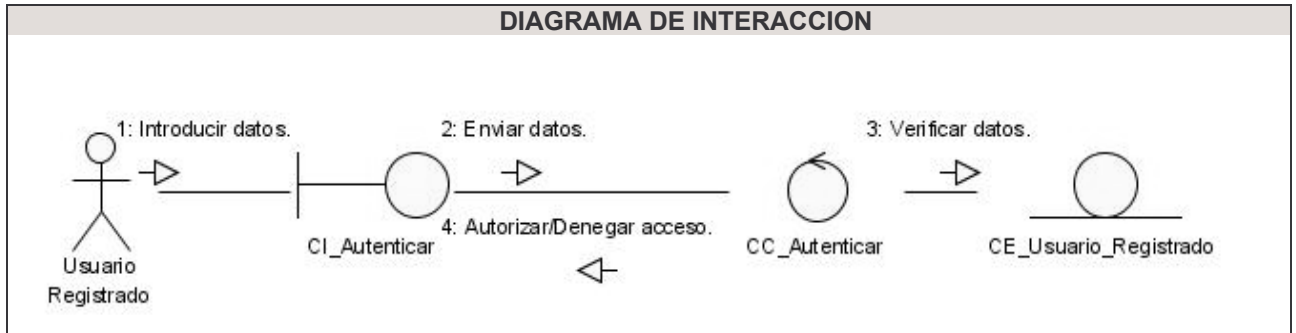
3.2.13 Diagrama de clases del análisis de Caso de Uso: Configurar reporte.



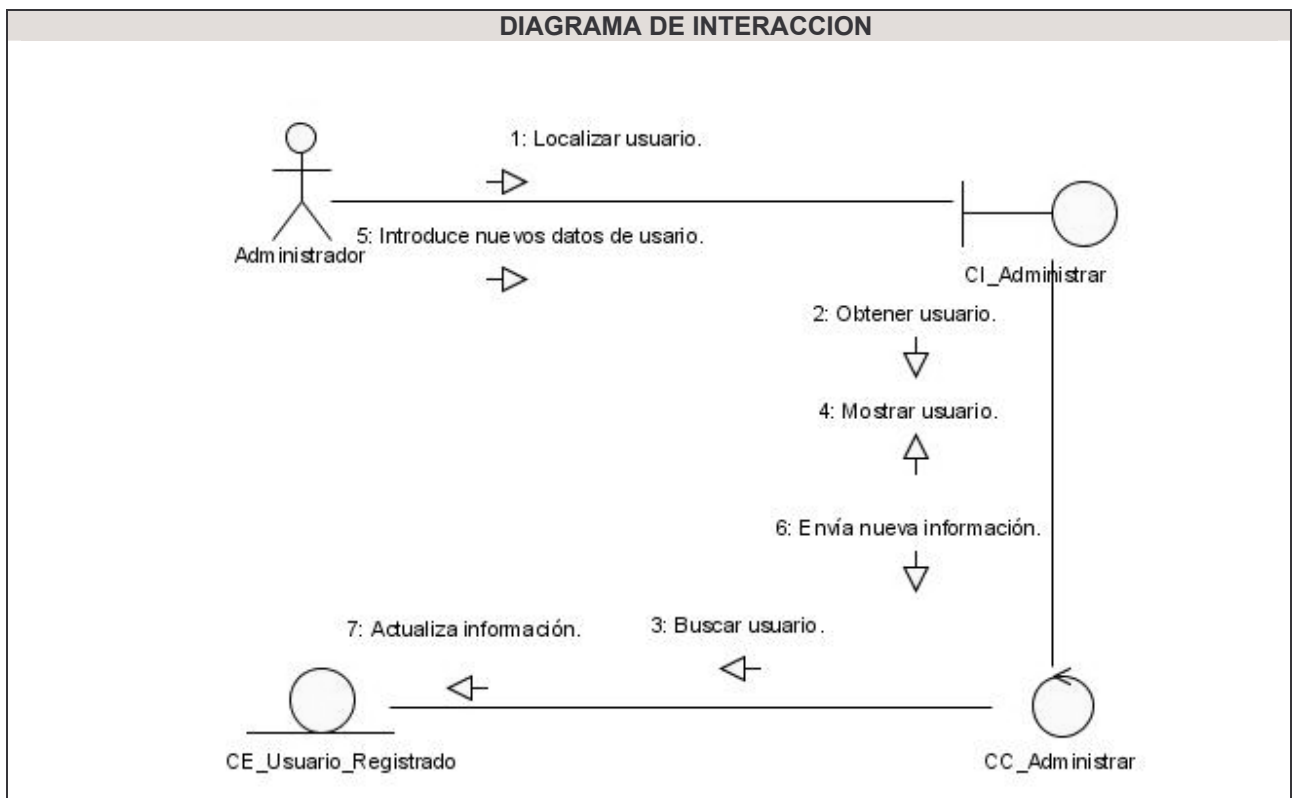


3.3 Diagramas de Interacción.

3.3.1 Diagrama de colaboración del caso de uso: Autenticar usuario.

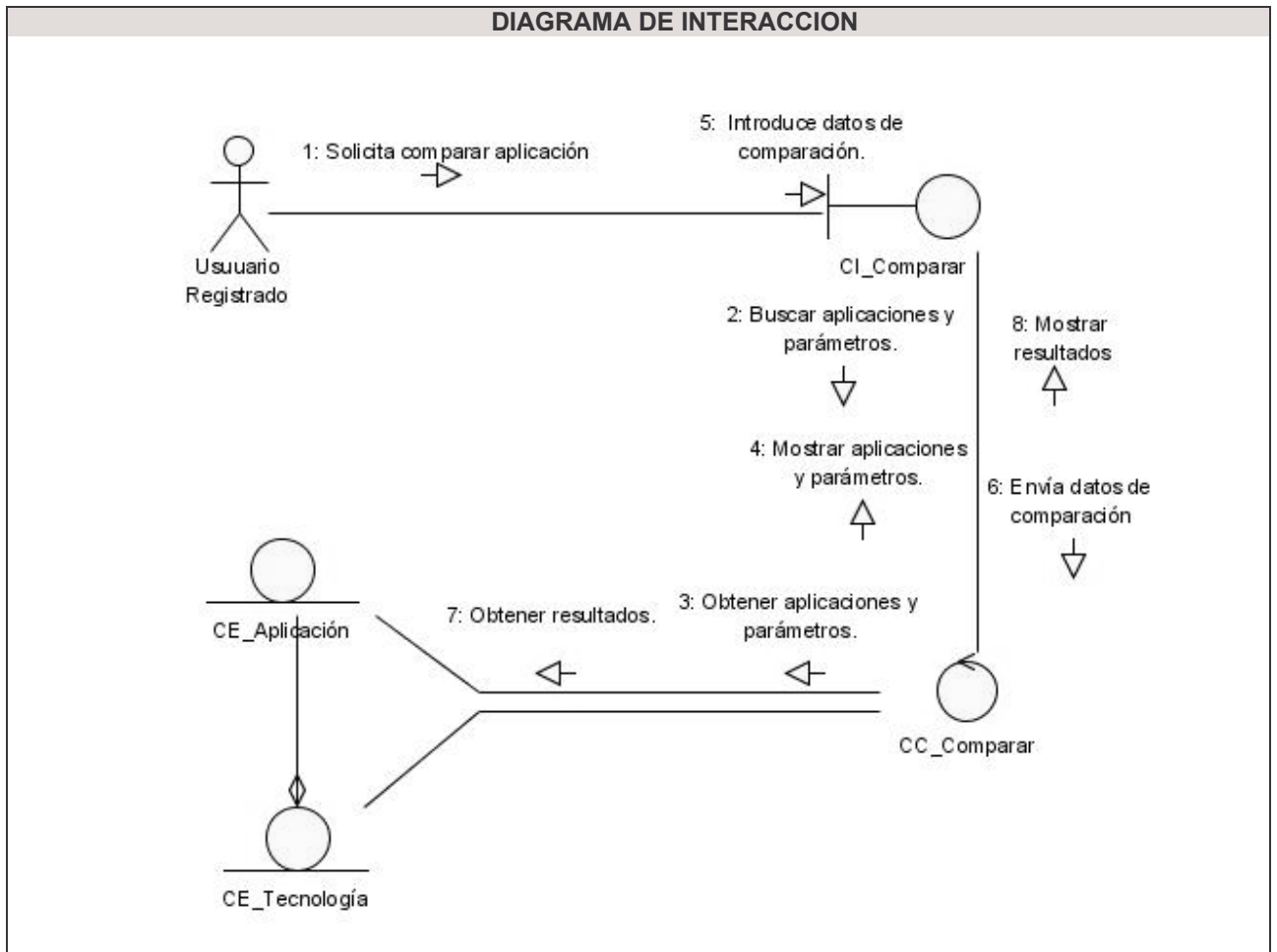


3.3.2 Diagrama de colaboración del caso de uso: Administrar usuario.



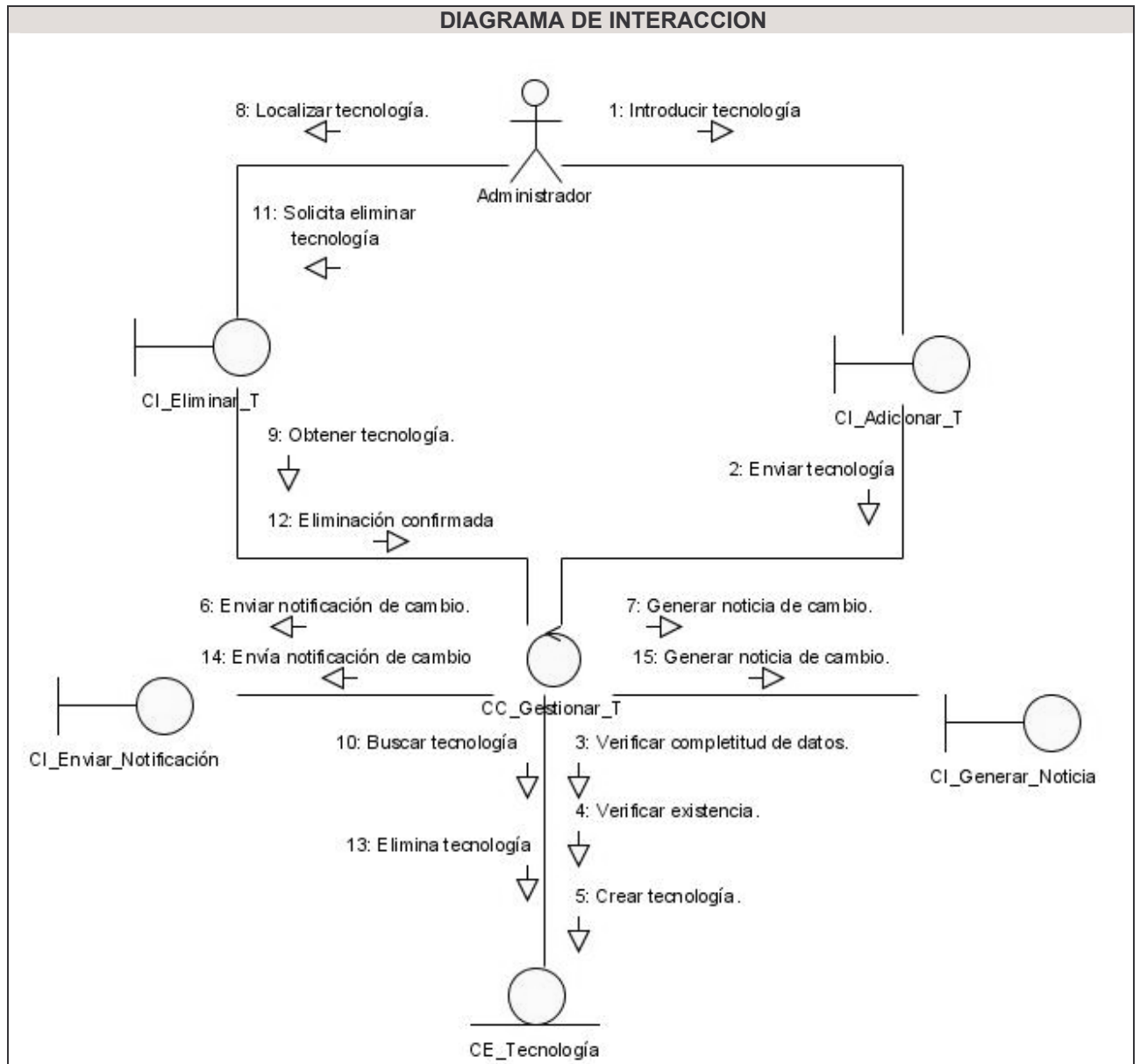


3.3.3 Diagrama de colaboración del caso de uso: Comparar aplicaciones.



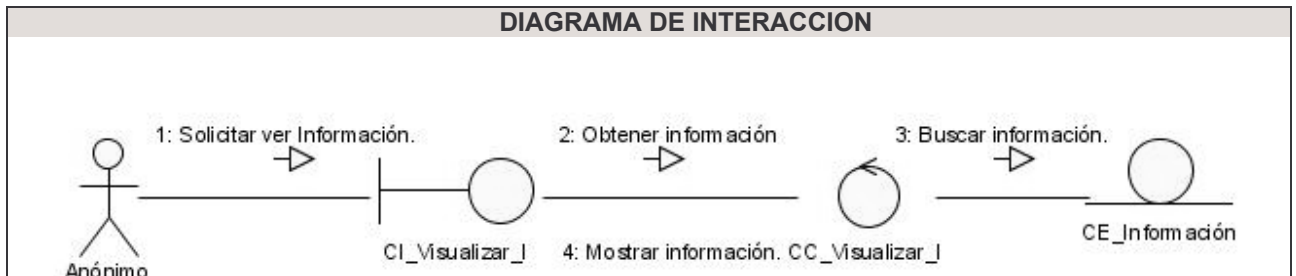


3.3.4 Diagrama de colaboración del caso de uso: Gestionar tecnología.

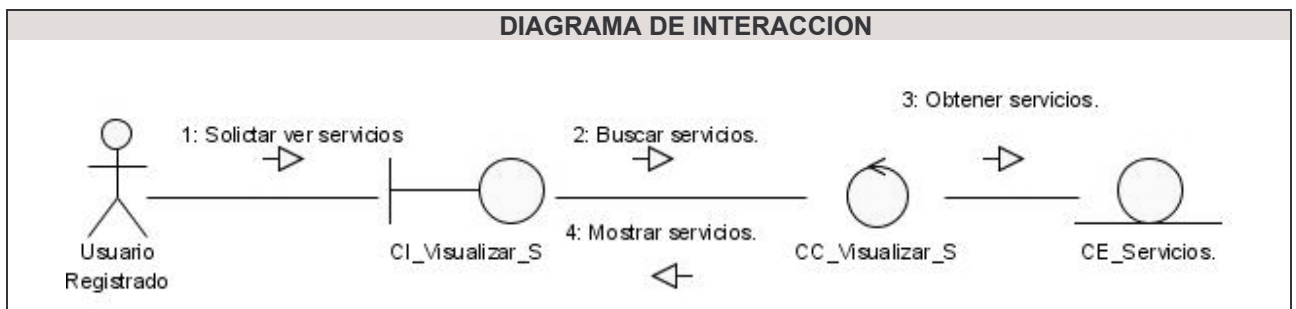




3.3.5 Diagrama de colaboración del caso de uso: Visualizar información.

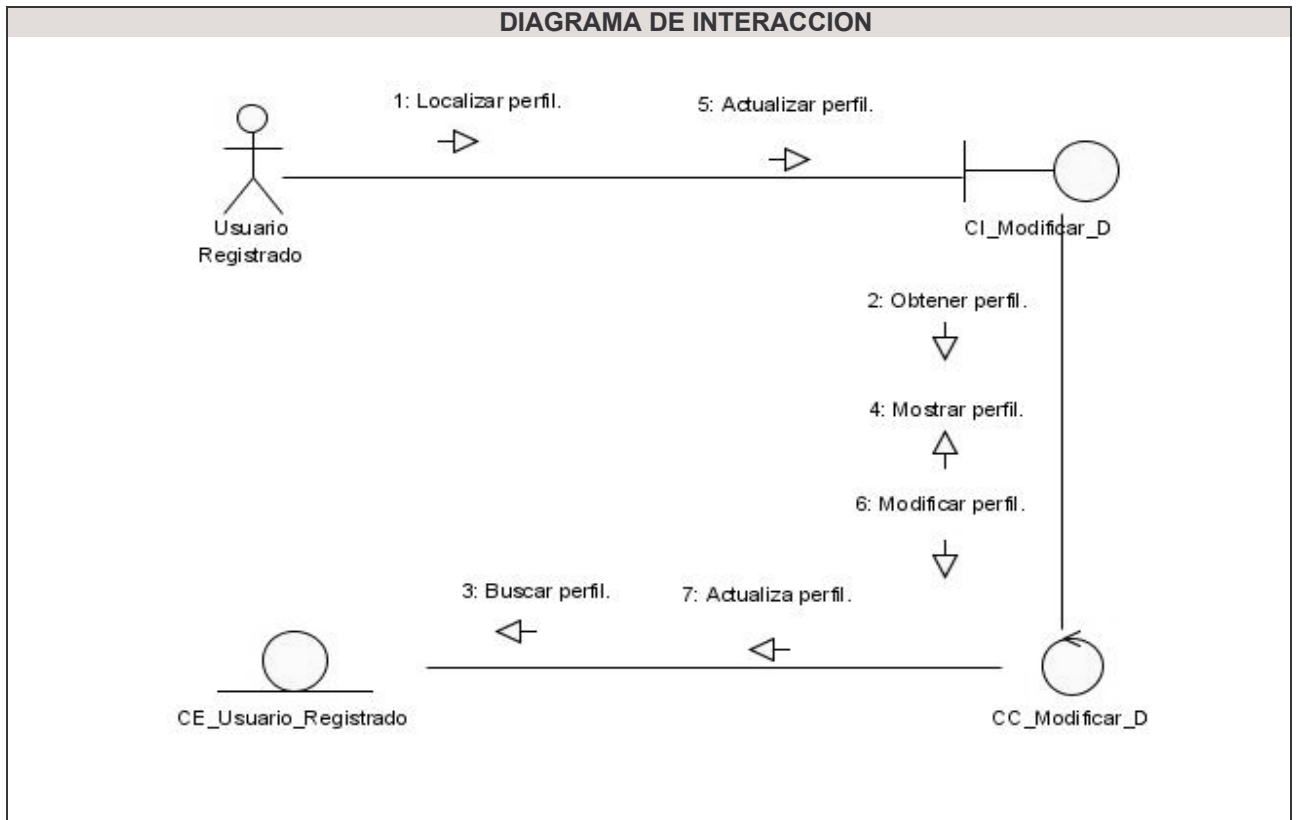


3.3.6 Diagrama de colaboración del caso de uso: Visualizar servicios.

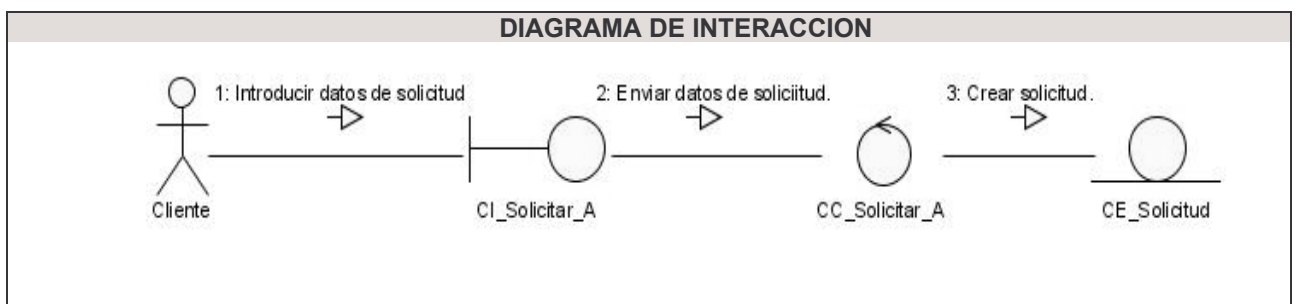




3.3.7 Diagrama de colaboración del caso de uso: Modificar datos de usuario.

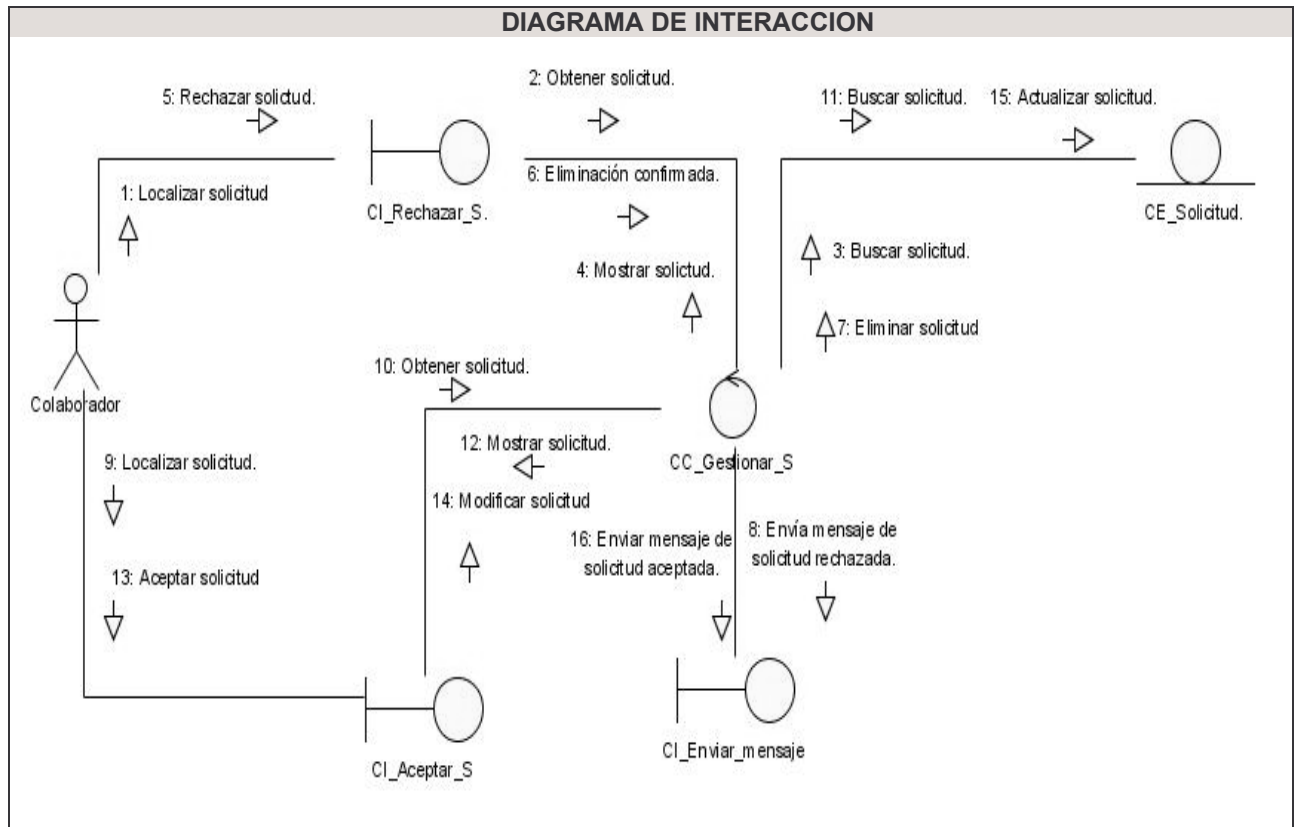


3.3.8 Diagrama de colaboración del caso de uso: Solicitar asesoría.



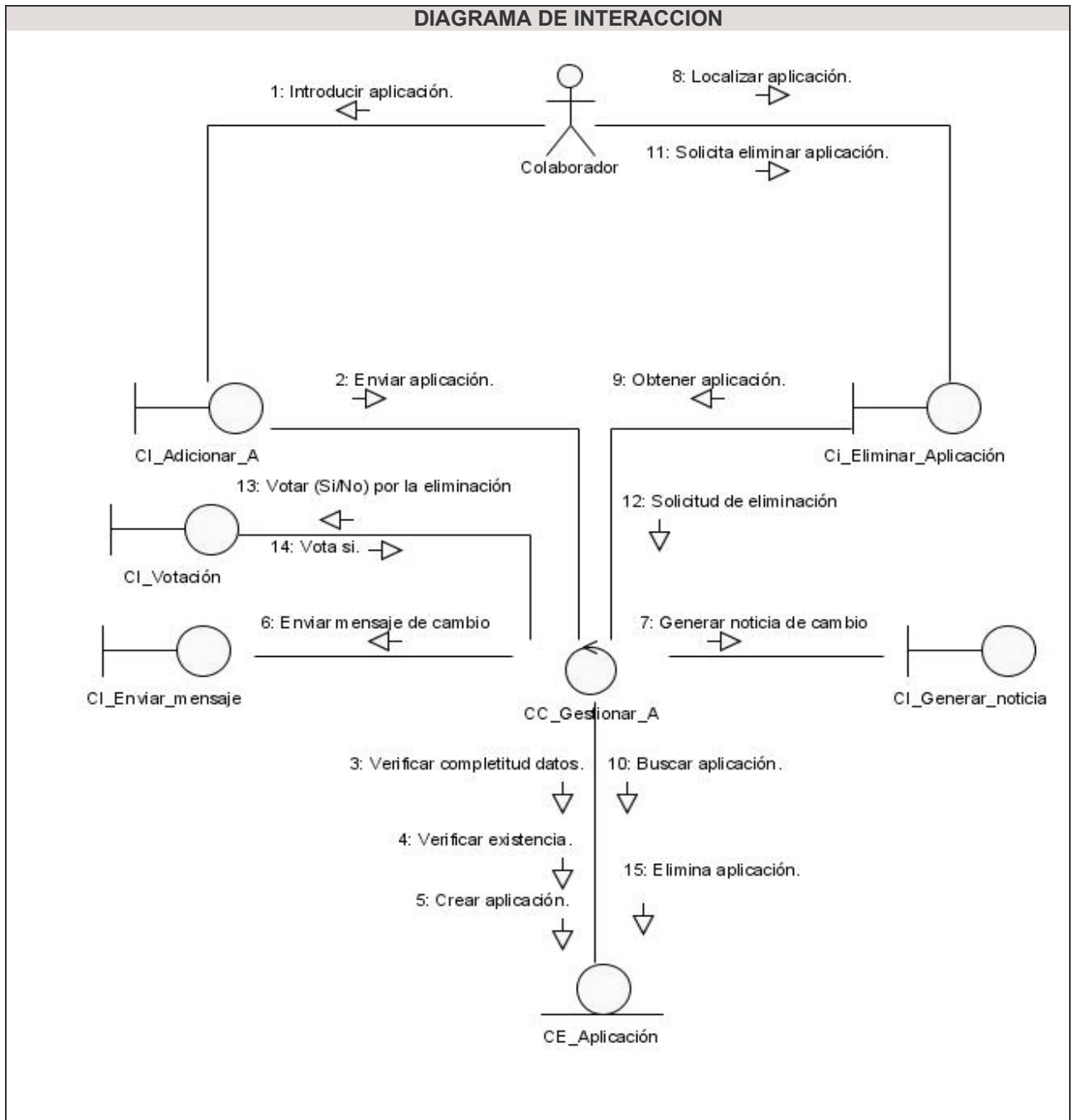


3.3.9 Diagrama de colaboración del caso de uso: Gestionar solicitud.



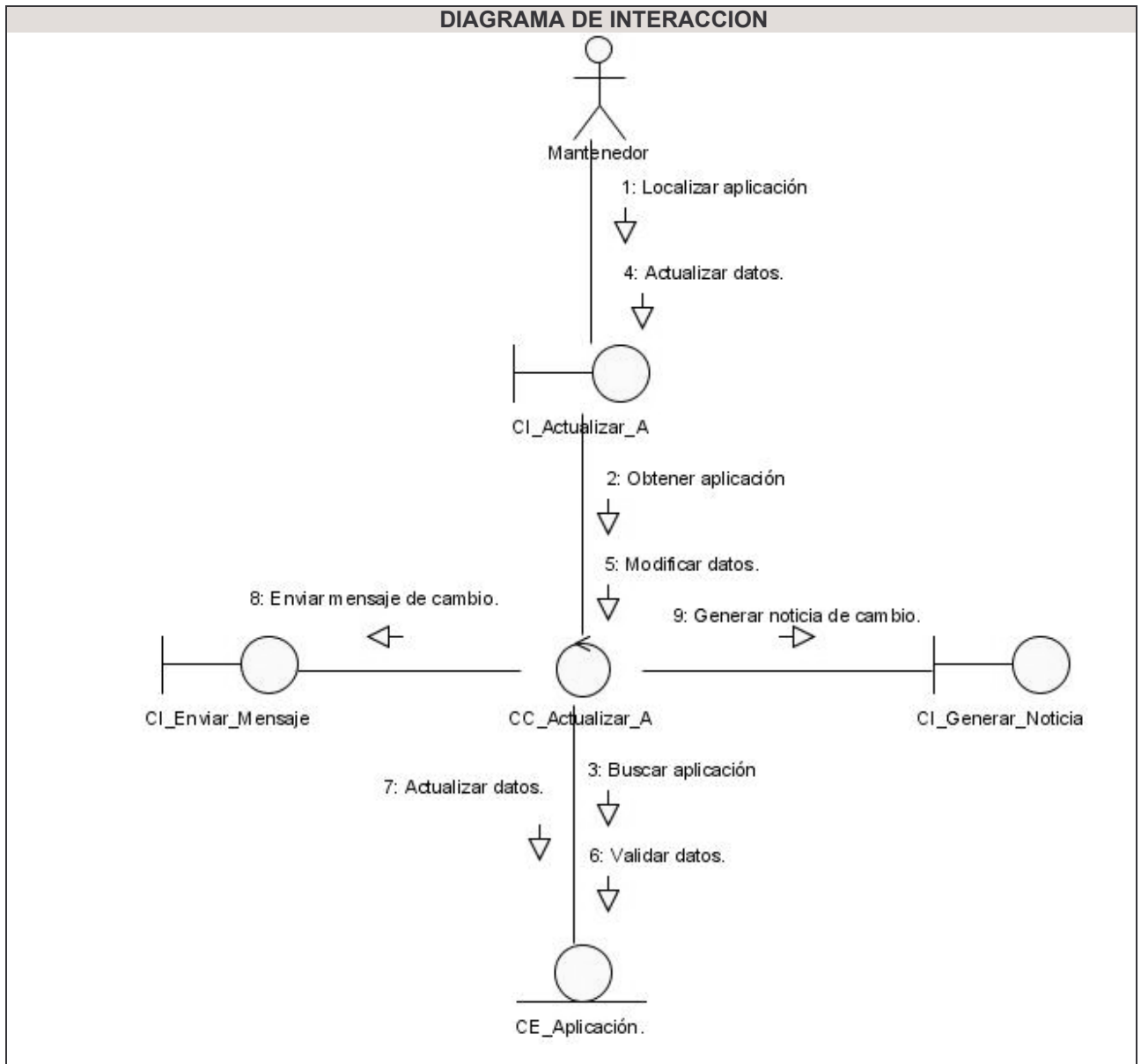


3.3.10 Diagrama de colaboración del caso de uso: Gestionar aplicación.



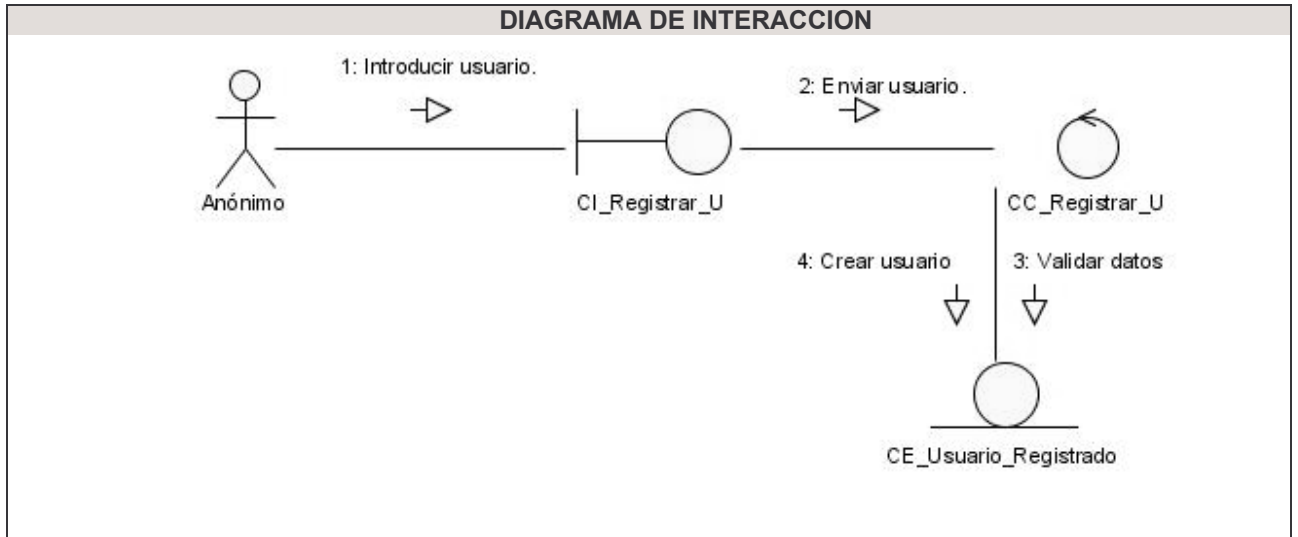


3.3.11 Diagrama de colaboración del caso de uso: Actualizar aplicación.



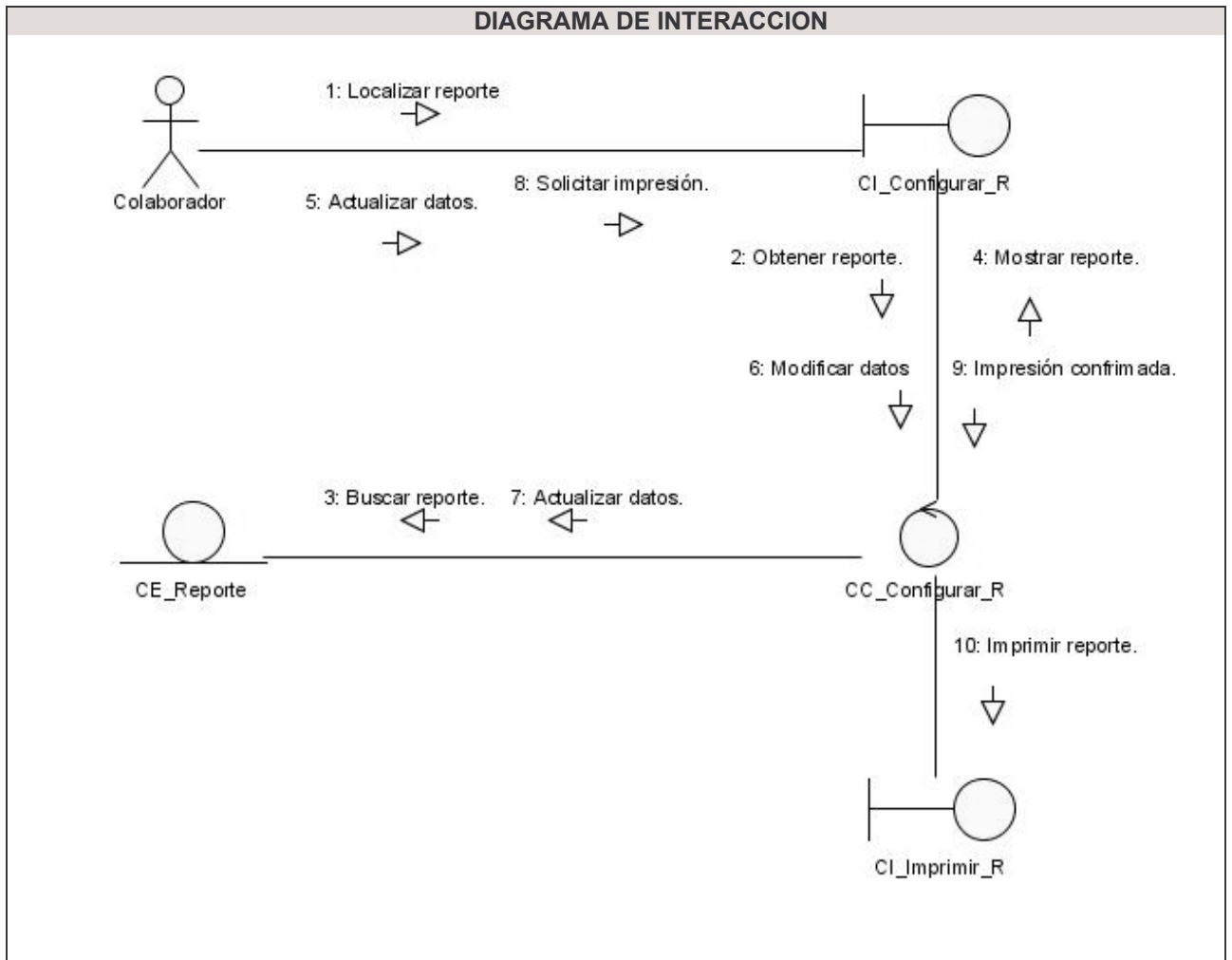


3.3.12 Diagrama de colaboración del caso de uso: Registrar usuario.





3.3.13 Diagrama de colaboración del caso de uso: Configurar reporte.





3.4 Diseño.

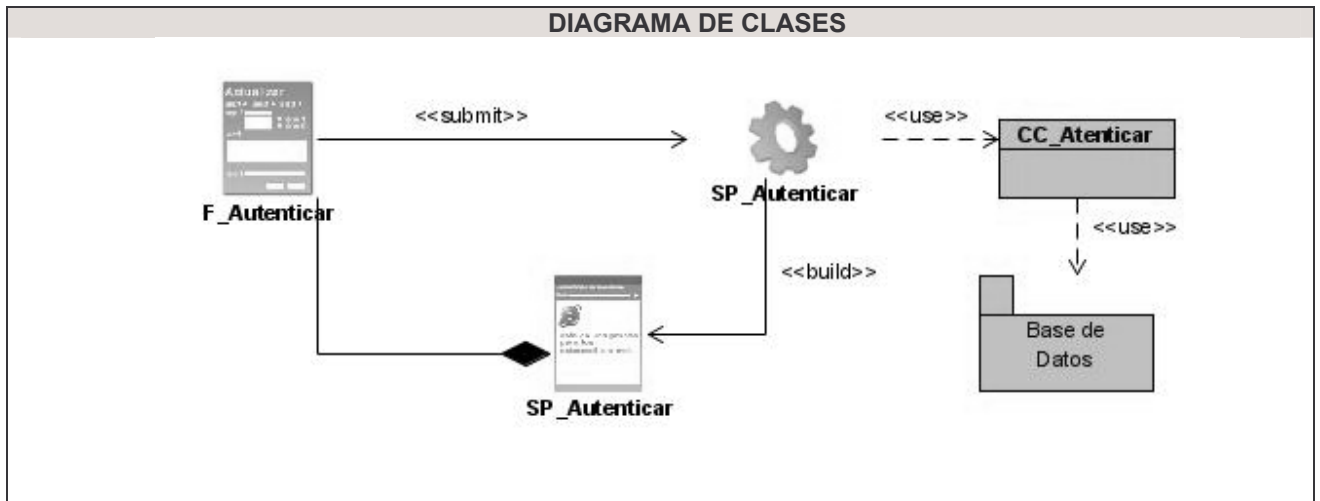
RUP permite a través del flujo de trabajo Diseño, utilizar las ventajas de una tecnología orientada a objetos. La Programación Orientada a Objetos (POO u OOP según siglas en inglés) es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (es decir, datos), comportamiento (esto es, procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar. [14]

En este subepígrafe se presentan los resultados obtenidos luego de realizar, a partir del análisis, el diseño del sistema propuesto. Estos resultados se refieren fundamentalmente a los diagramas de clases del diseño, diagramas de interacción, descripción de las clases del diseño y modelo de datos.

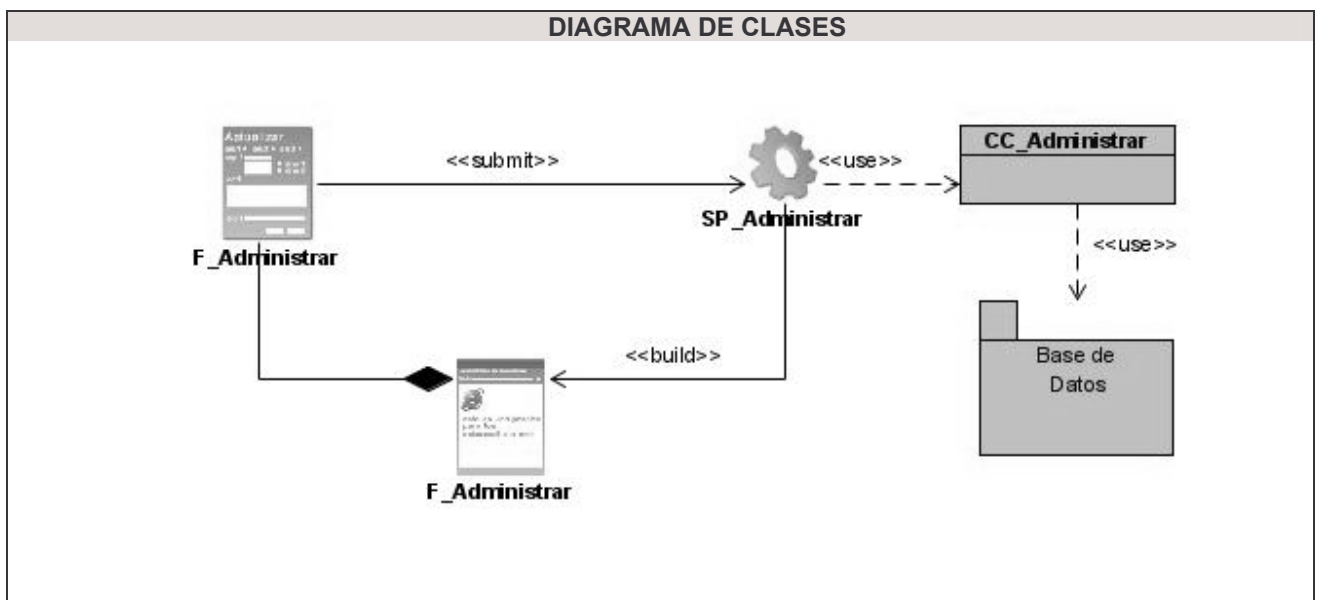


3.4.1 Diagramas de Clases del Análisis.

3.4.1.1 Diagrama de Clases del Diseño del Caso de uso: Autenticar usuario.

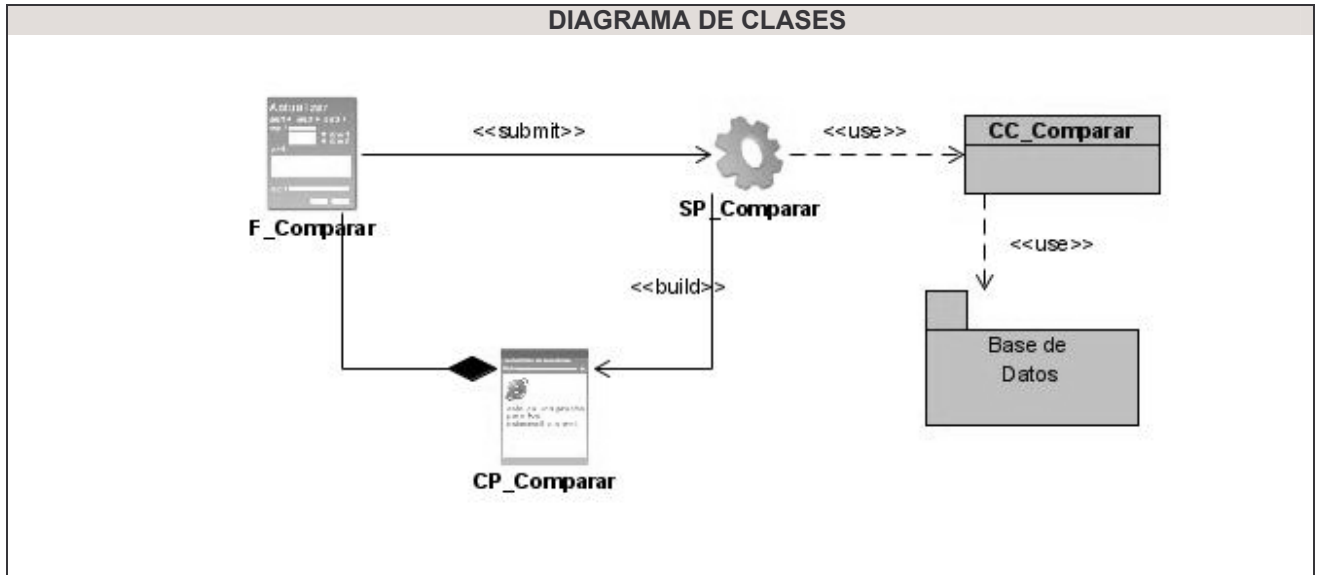


3.4.1.2 Diagrama de Clases del Diseño del Caso de uso: Administrar usuario.



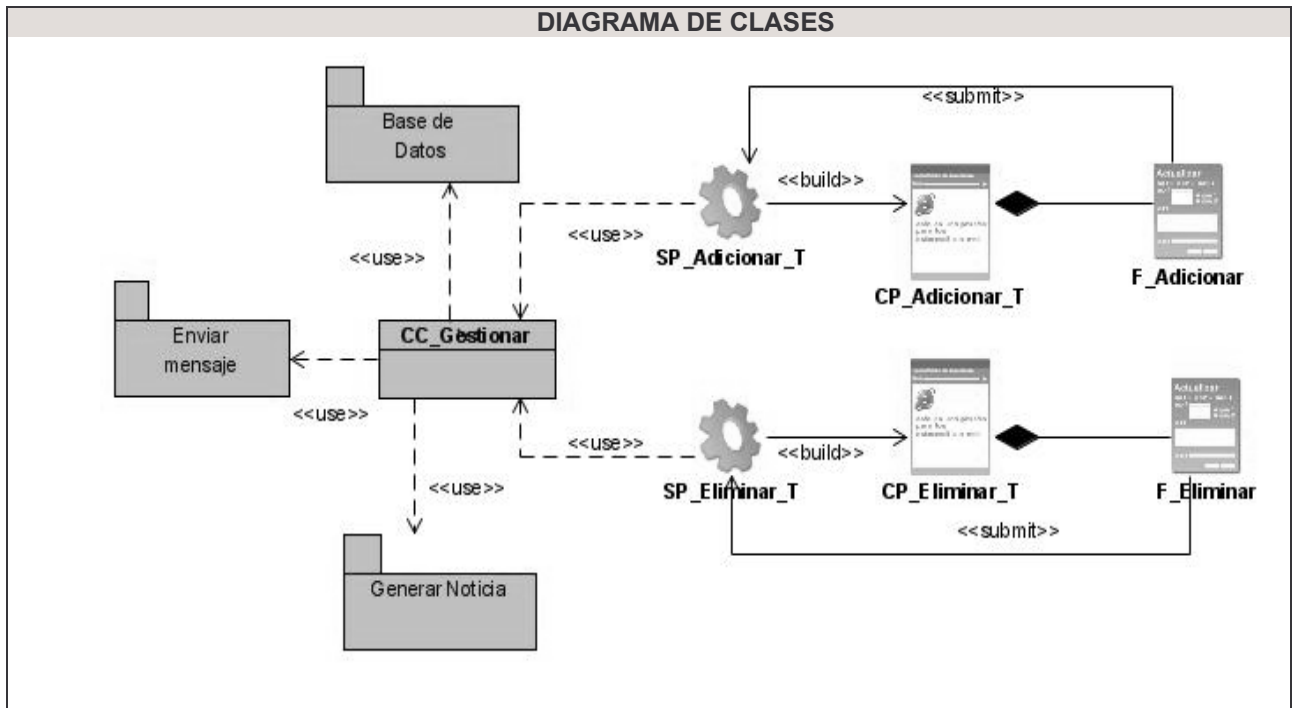


3.4.1.3 Diagrama de Clases del Diseño del Caso de uso: Comparar aplicaciones.

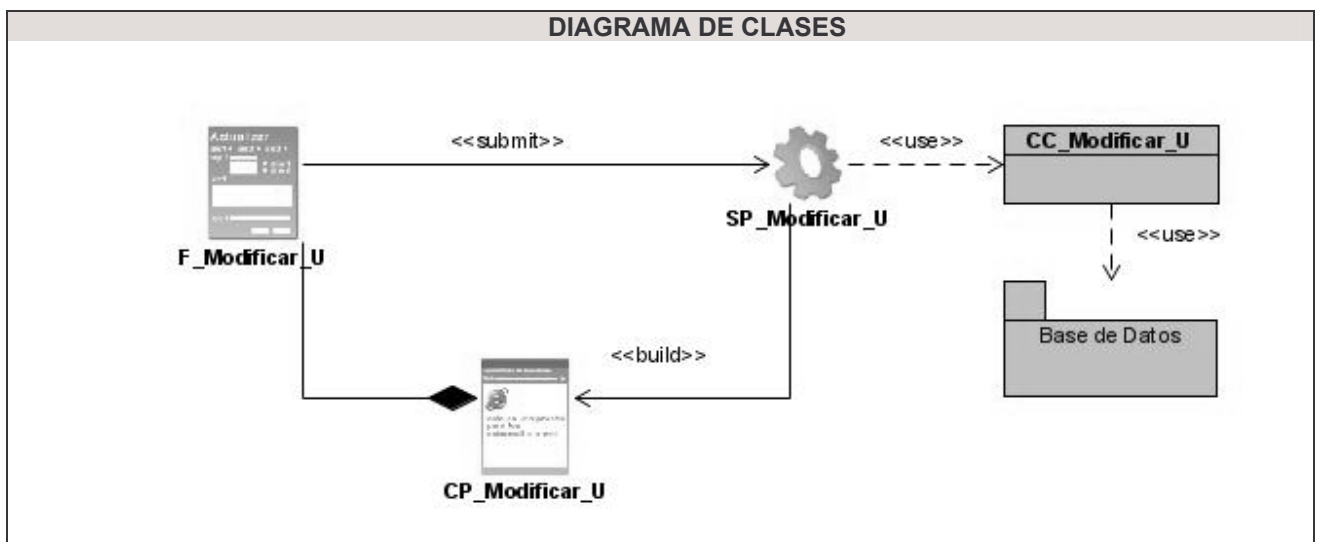




3.4.1.4 Diagrama de Clases del Diseño del Caso de uso: Gestionar tecnología.

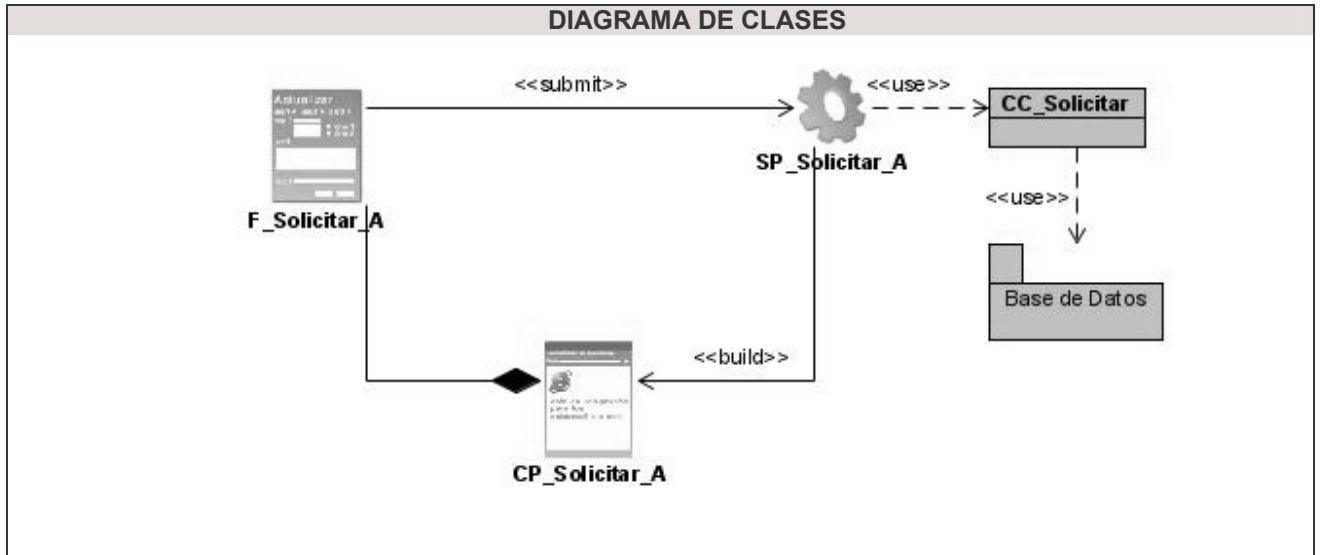


3.4.1.5 Diagrama de Clases del Diseño del Caso de uso: Modificar datos de Usuario.

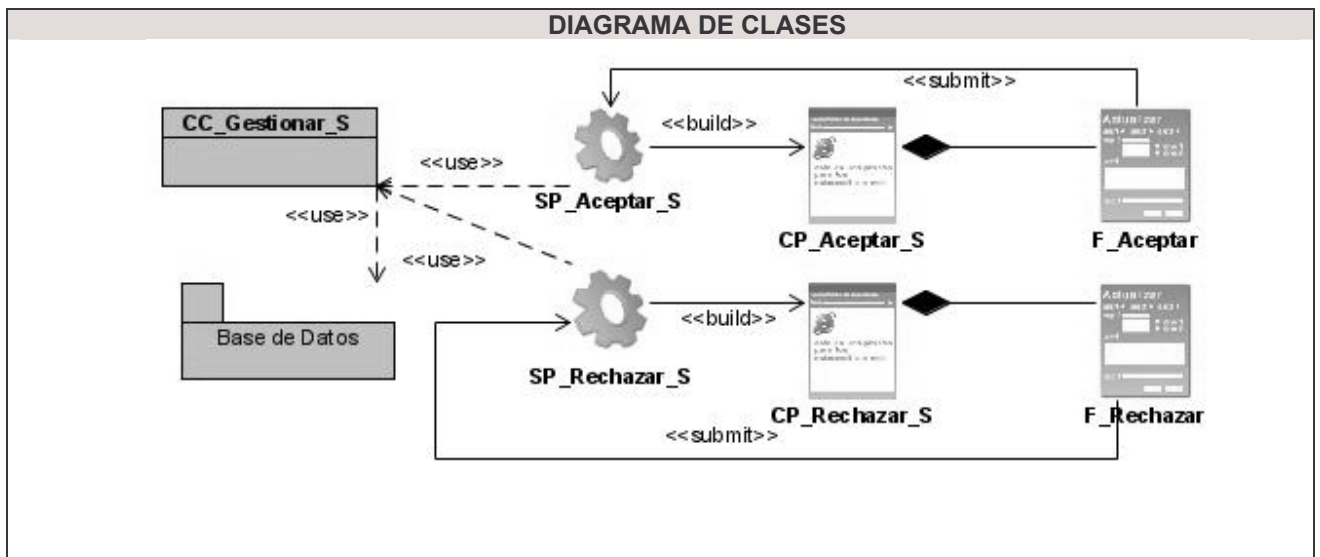




3.4.1.6 Diagrama de Clases del Diseño del Caso de uso: Solicitar Asesoría.

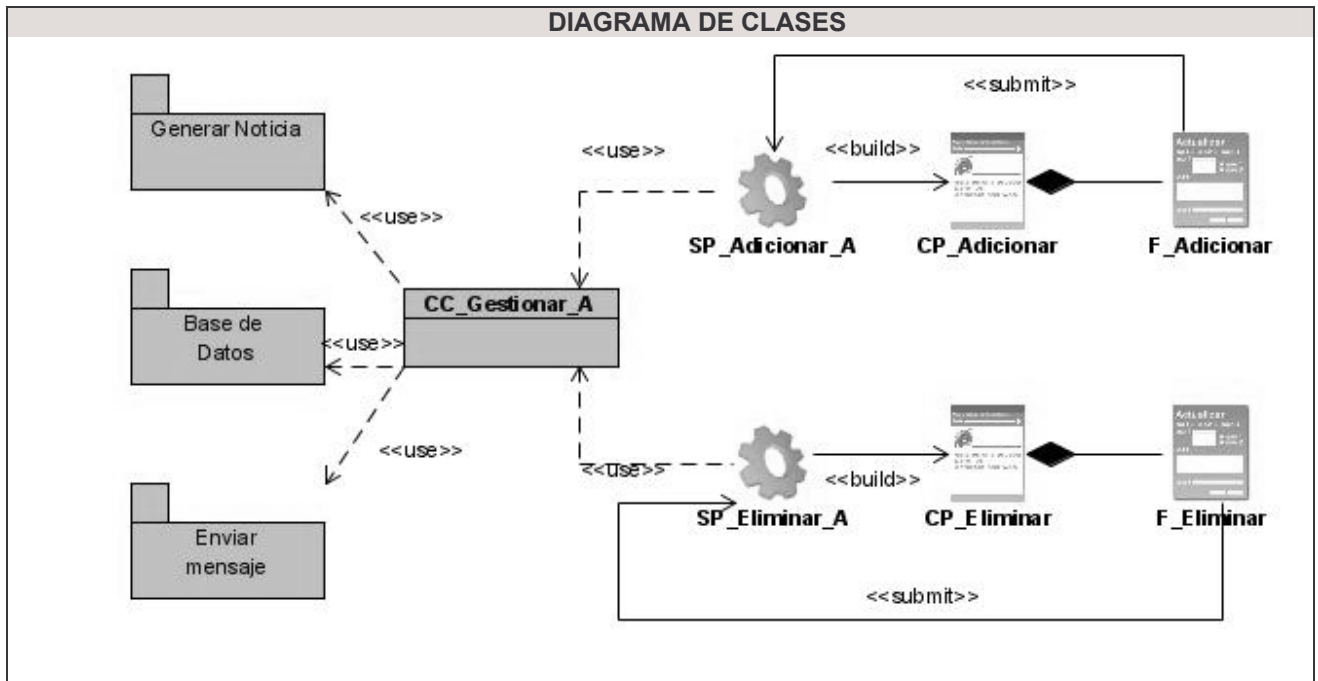


3.4.1.7 Diagrama de Clases del Diseño del Caso de uso: Gestionar Solicitud.



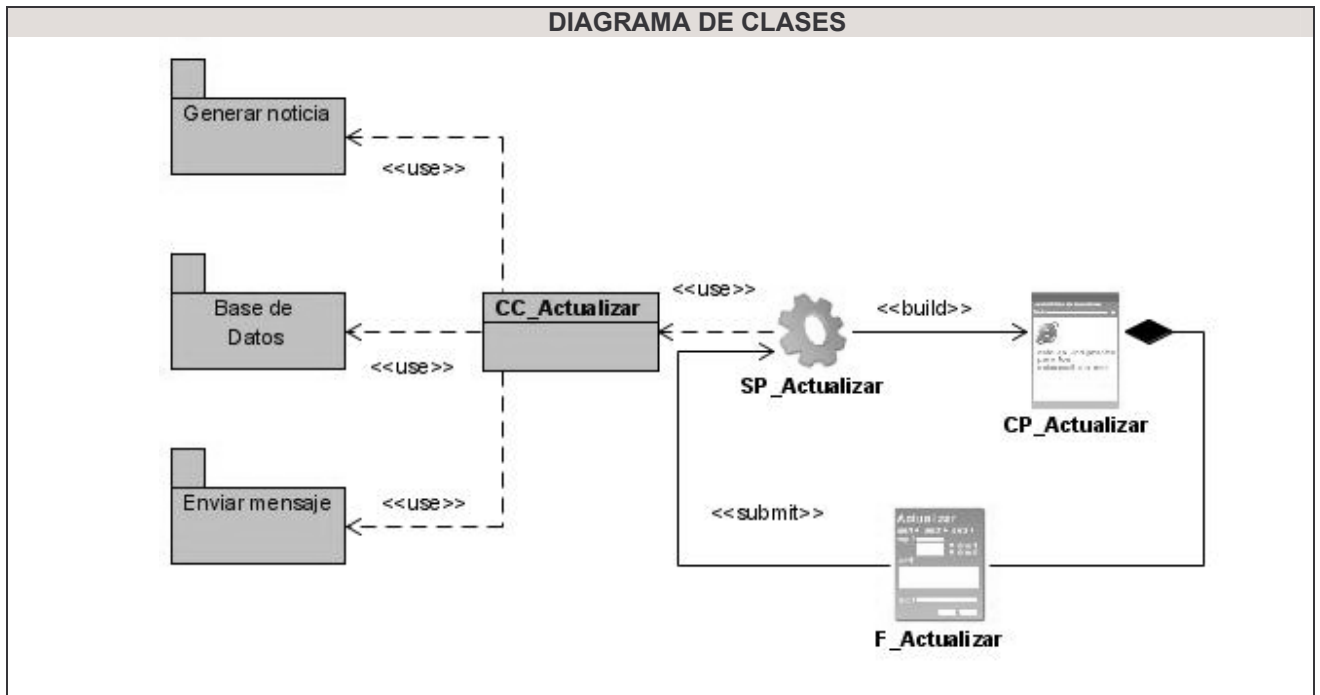


3.4.1.8 Diagrama de Clases del Diseño del Caso de uso: Gestionar Aplicación.

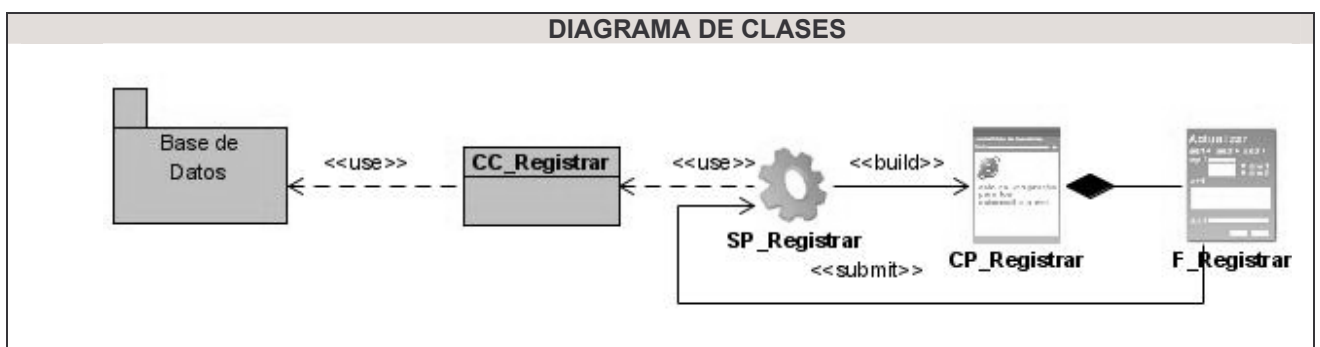




3.4.1.9 Diagrama de Clases del Diseño del Caso de uso: Actualizar Aplicación.

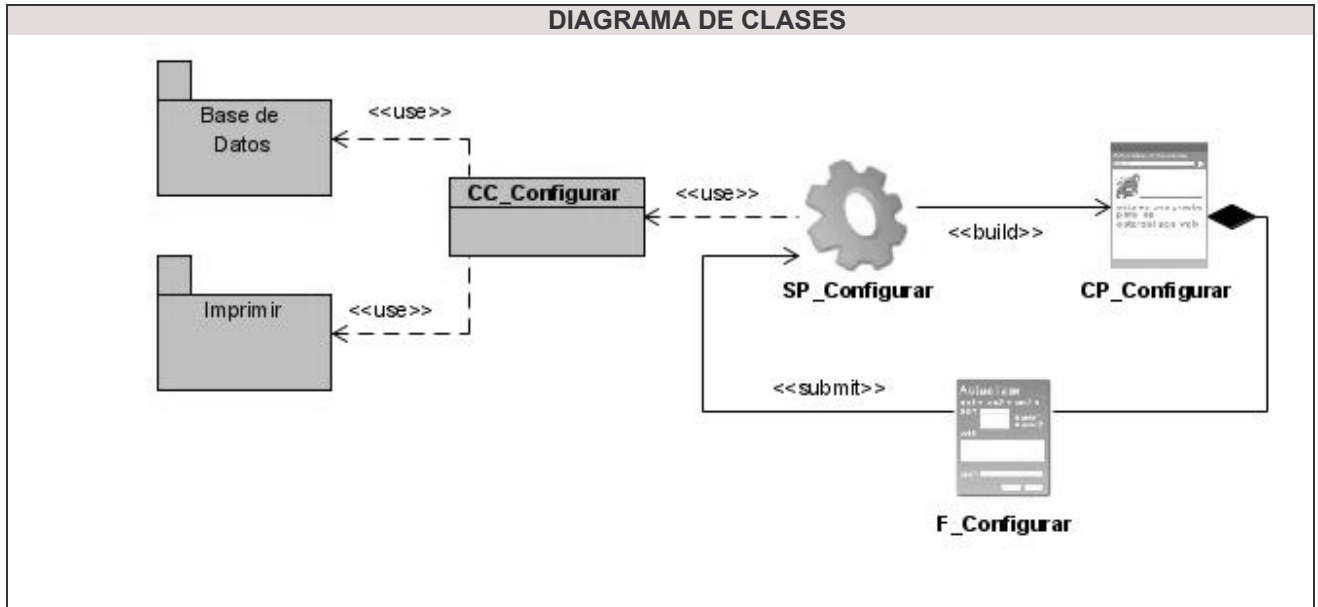


3.4.1.10 Diagrama de Clases del Diseño del Caso de uso: Registrar Usuario.





3.4.1.11 Diagrama de Clases del Diseño del Caso de uso: Configurar reporte.



3.4.2 Descripción de las clases del diseño.

Nombre: SP_Autenticar	
Tipo de clase: controladora	
Atributo	Tipo
Usuario	varchar
Contraseña	Varchar (encriptado MD5)
Para cada responsabilidad:	
Nombre:	Crear_Sesion_Nav ()
Descripción:	Este método crea una sesión en el navegador.
Nombre:	Verifica_User ()
Descripción:	Verifica que exista el usuario en la base de datos y además que los datos sean correctos.

Nombre: SP_Administrar	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Listar_User()
Descripción:	Muestra el listado de los usuarios que existen.



Nombre:	Modificar_Rol (ID_U, permiso)
Descripción:	Permite asignar un rol a un usuario determinado.
Nombre:	Ver_datos_U
Descripción:	Permite visualizar los datos de los usuarios.
Nombre:	Mostrar_Rol
Descripción:	Muestra la descripción del rol.

Nombre: SP_Comparar	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Mostrar_List_T()
Descripción:	Muestra listado de las tecnologías existentes.
Nombre:	Mostrar_List_A()
Descripción:	Muestra listado de las aplicaciones existentes.
Nombre:	Comp_A(ID_A: array, parámetro: array)
Descripción:	Permite comparar las aplicaciones a través de los parámetros.
Nombre:	Mostrar_Inf_A()
Descripción:	Muestra la información referente a una aplicación.

Nombre: SP_Adicionar_T	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Registrar_New_T()
Descripción:	Permite introducir una tecnología nueva.
Nombre:	Generar_Not_C()
Descripción:	Generar una noticia en la página de noticias al producirse el cambio.
Nombre:	Enviar_Not()
Descripción:	En caso de ocurrir algún cambio, permite enviar un mensaje al usuario suscrito al servicio correspondiente, informando del cambio.
Nombre:	Validar_dat(datos)
Descripción:	Permite validar los datos introducidos, de manera que los mismos sean correctos.

Nombre: SP_Eliminar_T	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Mostrar_T()



Descripción:	Este método muestra las tecnologías almacenadas.
Nombre:	Eliminar_T()
Descripción:	Elimina una tecnología de la base de datos.
Nombre:	Generar_Not_C()
Descripción:	Generar una noticia en la página de noticias al producirse el cambio.
Nombre:	Enviar_Not()
Descripción:	En caso de ocurrir algún cambio, permite enviar un mensaje al usuario suscrito al servicio correspondiente, informando del cambio.

Nombre: SP_Modificar_U	
Tipo de clase: controladora	
Atributo	Tipo
ID_U	Integer
Para cada responsabilidad:	
Nombre:	Mostrar_D_Ex()
Descripción:	Muestra los datos del usuario almacenados en la base de datos.
Nombre:	Validar_Inf(información)
Descripción:	Valida que los datos introducidos sean válidos.
Nombre:	Actualizar()
Descripción:	Garantiza que la nueva información sea almacenada.

Nombre: SP_Solicitar_A	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Registrar_S(Nueva_solicitud)
Descripción:	Almacena en la base de datos la solicitud realizada.

Nombre: SP_Aceptar_S	
Tipo de clase: controladora	
Atributo	Tipo
ID_U	integer
Para cada responsabilidad:	
Nombre:	Validar_D(Datos)
Descripción:	Se encarga de verificar que los datos sean correctos.
Nombre:	Registrar_S()
Descripción:	Registra la solicitud aceptada en la base de datos como Solicitud Aceptada.
Nombre:	Enviar_N
Descripción:	Envía un mensaje al usuario que hizo la solicitud diciendo que la misma ha sido aceptada, informando además fecha en que se realizará, y colaborador que la atenderá.



Nombre: SP_Rechazar_A	
Tipo de clase: controladora	
Atributo	Tipo
ID_U	Integer
Para cada responsabilidad:	
Nombre:	Mostrar_L_S()
Descripción:	Muestra el listado de solicitudes existentes en la base de datos, pendientes a revisar.
Nombre:	Eliminar_Sol()
Descripción:	Elimina una solicitud de la base de datos.

Nombre: SP_Adicionar_A	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Registrar_New_A()
Descripción:	Permite introducir una aplicación nueva.
Nombre:	Generar_Not_C()
Descripción:	Generar una noticia en la página de noticias al producirse el cambio.
Nombre:	Enviar_Not()
Descripción:	En caso de ocurrir algún cambio, permite enviar un mensaje al usuario suscrito al servicio correspondiente, informando del cambio.
Nombre:	Validar_dat(datos)
Descripción:	Permite validar los datos introducidos, de manera que los mismos sean correctos.

Nombre: SP_Eliminar_A	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Mostrar_A()
Descripción:	Este método muestra las aplicaciones almacenadas.
Nombre:	Eliminar_A()
Descripción:	Elimina una aplicación de la base de datos.
Nombre:	Generar_Not_C()
Descripción:	Generar una noticia en la página de noticias al producirse el cambio.
Nombre:	Enviar_Not()
Descripción:	En caso de ocurrir algún cambio, permite enviar un mensaje al usuario suscrito al servicio correspondiente, informando del cambio.



Nombre: SP_Actualizar	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Listar_A()
Descripción:	Muestra el listado de las aplicaciones existentes en la base de datos.
Nombre:	Actualizar_D(datos, ID_A)
Descripción:	Permite introducir nuevos datos de una aplicación específica.
Nombre:	Select_ap(ID_A)
Descripción:	Permite seleccionar una aplicación.
Nombre:	Generar_Not_C()
Descripción:	Generar una noticia en la página de noticias al producirse el cambio.
Nombre:	Enviar_Not()
Descripción:	En caso de ocurrir algún cambio, permite enviar un mensaje al usuario suscrito al servicio correspondiente, informando del cambio.

Nombre: SP_Registrar	
Tipo de clase: controladora	
Atributo	Tipo
Usuario	varchar
Contraseña	varchar (encriptado MD5)
Para cada responsabilidad:	
Nombre:	Validar_D()
Descripción:	Verifica que los datos sean correctos.
Nombre:	Registrar_U()
Descripción:	Registra el usuario en la Base de Datos.

Nombre: SP_Configurar	
Tipo de clase: controladora	
Atributo	Tipo
ID_Re	Integer
Para cada responsabilidad:	
Nombre:	Mostrar_R()
Descripción:	Muestra un reporte determinado.
Nombre:	Actualizar_R(datos)
Descripción:	Guardar los nuevos datos en el reporte correspondiente.
Nombre:	Imprimir()
Descripción:	Manda a imprimir el repote seleccionado.

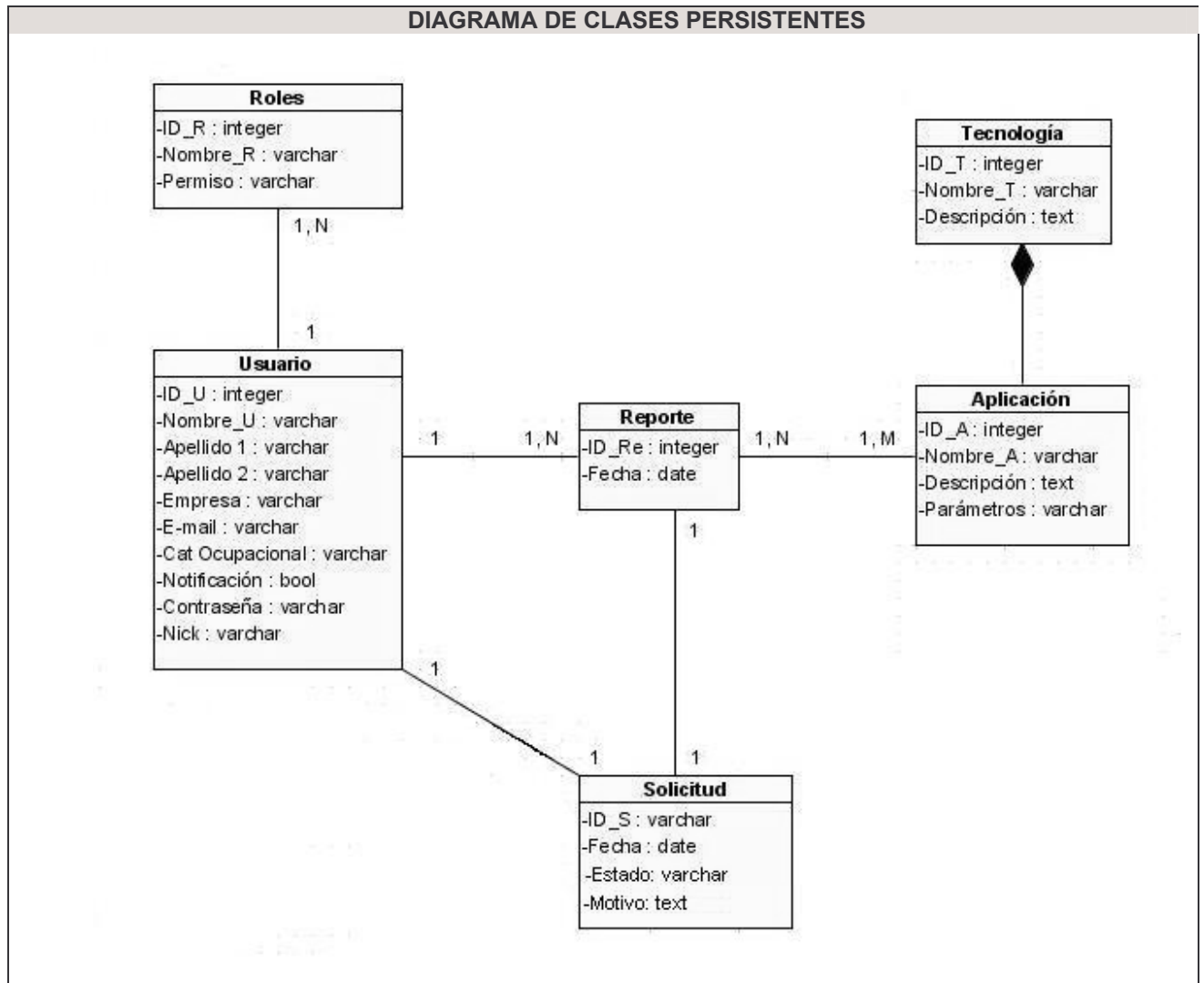


3.5 Diseño de la Base de Datos.

Se presenta a continuación el diseño de la Base de Datos, modelo que se representa mediante el diagrama de clases persistentes una vez seleccionadas estas. Estas clases representan los datos que se obtienen y se almacenan en todo el proceso del producto, y las relaciones entre ellos. A partir del diagrama de clases persistentes, se presenta el modelo de datos correspondiente, así como la descripción de las tablas de este modelo de datos.

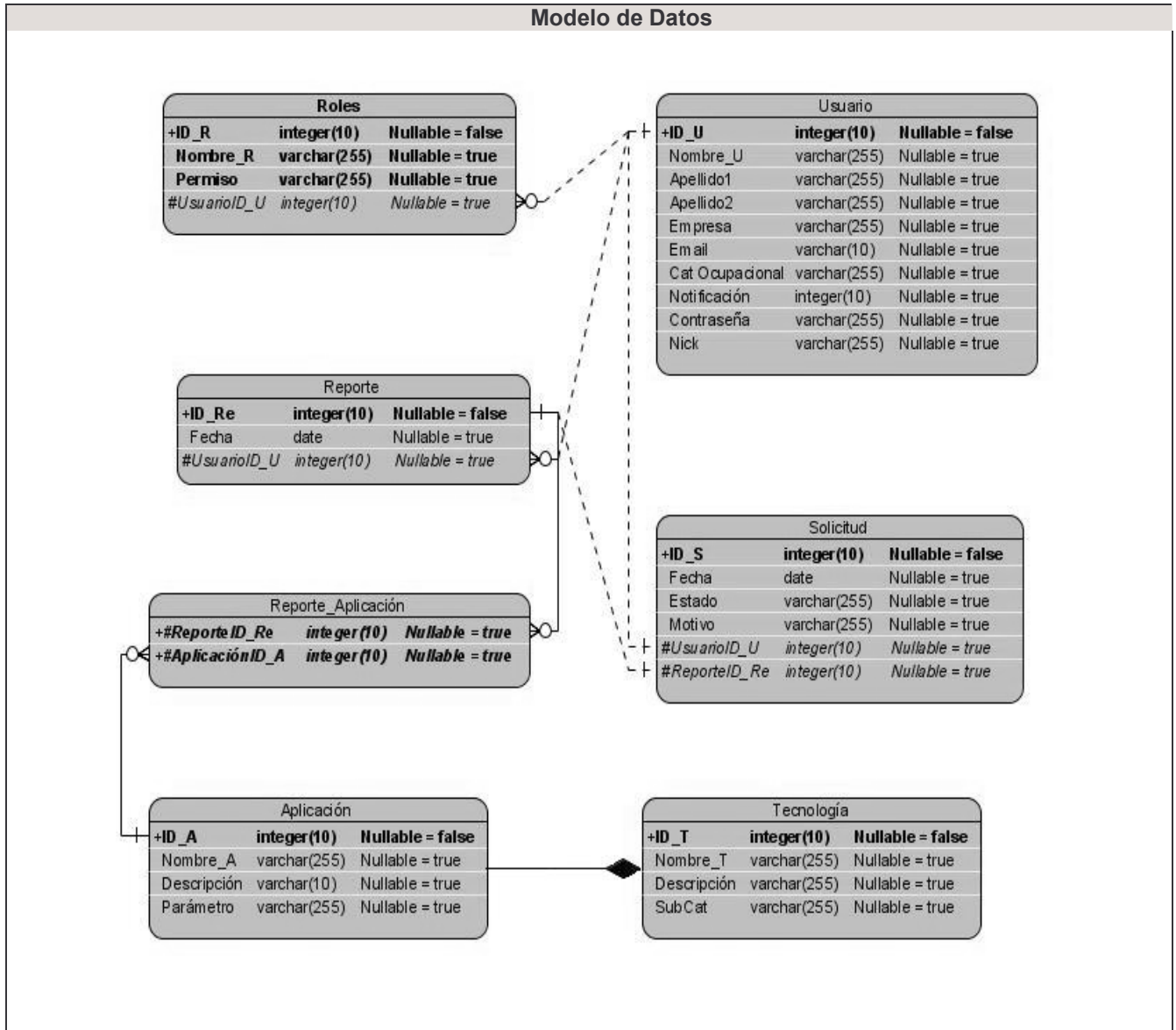


3.5.1 Diagrama de Clases Persistentes.





3.5.2 Modelo de la Base de datos.





3.5.3 Descripción de las tablas de la Base de Datos.

Nombre: Roles		
Descripción: Esta clase contiene todos los roles que pueden ser asignados a los usuarios.		
Atributo	Tipo	Descripción
ID_R	integer	Identificador de la tabla.
Nombre_R	varchar	Nombre del rol que será asignado.
Permiso	varchar	Permiso de un rol específico.

Nombre: Usuario		
Descripción: Esta tabla contiene la información de todos los usuarios del Observatorio.		
Atributo	Tipo	Descripción
ID_U	integer	Identificador de la tabla.
Nombre_U	varchar	Nombre del usuario.
Apellido1	varchar	Primer apellido del usuario.
Apellido2	varchar	Segundo apellido del usuario.
Empresa	varchar	Nombre de la empresa a la que pertenece.
E-mail	varchar	Dirección de correo electrónico del usuario.
Cat ocupacional	varchar	Categoría ocupacional del usuario. (Estudiante, ingeniero, licenciado, obrero, etc.)
Notificación	bool	Contiene la decisión del usuario de recibir notificación vía e-mail o no, cuando se realice alguna actualización en el Observatorio.
Contraseña	varchar	Contraseña de acceso al Observatorio del usuario.
Nick	varchar	Nombre con que aparece en el Observatorio y que además utiliza para autenticarse.

Nombre: Solicitud		
Descripción: Esta clase contiene las solicitudes de asesoría realizadas por los usuarios.		
Atributo	Tipo	Descripción
ID_S	integer	Identificador de la tabla.
Fecha	date	Contiene la fecha que se le atención a la solicitud. O sea, en que fecha será atendido el usuario solicitante.
Estado	varchar	Contiene el estado de la solicitud. Puede ser: en espera, aceptada, rechazada, terminada.



Motivo	text	Contiene la descripción que introduce el usuario justificando el por qué solicita la asesoría.
--------	------	--

Nombre: Reporte		
Descripción: Esta tabla contiene los reportes generados de las asesorías personalizadas.		
Atributo	Tipo	Descripción
ID_R	integer	Identificador de la tabla.
Fecha	date	Fecha en que se realizó el reporte.

Nombre: Tecnología.		
Descripción: Esta tabla contiene las tecnologías que sustenta el Observatorio.		
Atributo	Tipo	Descripción
ID_T	integer	Identificador de la tabla.
Nombre_T	varchar	Contiene el nombre de la tecnología.
Descripción	text	Contiene una descripción de la tecnología.
Subcat	varchar	Se refiere a las categorías específicas en que se clasifican las categorías generales.

Nombre: Aplicación.		
Descripción: Esta tabla contiene las aplicaciones		
Atributo	Tipo	Descripción
ID_A	integer	Identificador de la tabla.
Nombre_A	varchar	Se refiere al nombre de la aplicación.
Descripción	text	Descripción a modo texto que caracteriza la aplicación.
Parámetros	varchar	Se refiere a características específicas de una aplicación.



Conclusiones.

La migración a Software Libre constituye un reto y una necesidad para Cuba, no solo desde el punto de vista moral y ético, sino también desde el punto de vista económico y social. Acometer un proceso de migración para todas las entidades sin contar con los recursos necesarios para garantizar el mismo, sería un grave error.

Una vez concluido este trabajo, se cuenta con una propuesta de diseño para proporcionar el desarrollo de la herramienta Observatorio de Tecnologías Libres que permitirá orientar a los organismos del estado en procesos de migración a plataformas libres sobre qué tecnologías y aplicaciones utilizar en cada momento. Es necesario agregar que esta propuesta constituye un punto de partida no estático, que permite mejorar lo ya establecido y que favorecerá la creación de un servicio colaborativo para el incremento racional de la información.

Dicha propuesta especifica el funcionamiento del Observatorio de manera funcional y organizacional, definiendo las responsabilidades de cada usuario y de que manera se desarrollarán los principales procesos.

Se puede decir que se investigaron todas las clasificaciones posibles para las tecnologías libres.

Además se elaboró un sistema de reportes, clasificación comparación para cada tecnología bien definida, o sea, una clasificación global de tecnologías libres que serán observadas en principio en el Observatorio de Tecnologías Libres, especificando aplicaciones y parámetros de comparación de las mismas.

Por tanto, se concluye que los objetivos propuestos para el presente trabajo de diploma, han sido cumplidos satisfactoriamente.



Recomendaciones.

- ✓ Implementar el Observatorio de Tecnologías Libres sobre las especificaciones planteadas en el trabajo de diploma.
- ✓ Seguir ampliando la clasificación de Tecnologías Libres inicialmente propuesta.
- ✓ Organizar el equipo de trabajo del proyecto que soportará el Observatorio de Tecnologías Libres.
- ✓ Definir procedimiento y estrategia de trabajo para lograr la óptima utilización del servicio una vez implementado.
- ✓ Que la propia Universidad de las Ciencias Informáticas y por consiguiente los organismos del estado, se acojan a este servicio para efectuar el proceso de migración.



- [1] JACOBSON, Ivar, BOOCH, Grady, RUMBAUGH, James. El proceso unificado de desarrollo de software, Pearson Educación S.A., 2000.
- [2] WIKIPEDIA. *PHP*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/Php>.
- [3] WIKIPEDIA. *MySQL*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/MySQL>.
- [4] LIBRARY, F. S. *Visual Paradigm for UML Professional Edition Details*, 2003. [2007]. Disponible en:
http://textver.filehungry.com/spanish/product/windows_software/programming/specialized_tools/visual_paradigm_for_uml_professional_edition
- [5] PARADIGM, V. *Visual Paradigm for UML 6.0 SP1*, 2007. [2007]. Disponible en:
<http://www.visual-paradigm.com/news/vpsuite30sp1/vpuml60sp1.jsp>
- [6] PARADIGM, V. *Why Visual Paradigm for UML?*, 2007. [2007]. Disponible en:
<http://www.visual-paradigm.com/product/vpuml>
- [7] WIKIPEDIA. *Sistema Operativo*, 2007. [2007]. Disponible en:
<http://es.wikipedia.org/wiki/SO>
- [8] WIKIPEDIA. *Audio*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/Audio>
- [9] WIKIPEDIA. *Bases de datos*, 2007. [2007]. Disponible en:
http://es.wikipedia.org/wiki/Base_de_datos
- [10] WIKIPEDIA. *Gráficos*, 2007. [2007]. Disponible en:
<http://es.wikipedia.org/wiki/Gr%C3%A1ficos>
- [11] WIKIPEDIA. *Internet*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/Internet>
- [12] WIKIPEDIA. *Software*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/software>



[13] WIKIPEDIA. *Video*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/video>

[14] WIKIPEDIA. *Programación Orientada a Objetos.*, 2006. [2007]. Disponible en:
<http://es.wikipedia.org/wiki/POO>



Bibliografía.

Booch, G., Rumbaugh, J., Jacobson, I. *“El Lenguaje Unificado de Modelado”*. Addison-Wesley. 1999.

DATE, C. J. *Sistemas de Bases de Datos*. La Habana, 2003. 915 p.

FSF/UNESCO. *Free Software Directory*, 2006. [2007]. Disponible en: <http://directory.fsf.org/>

GHOSH, R. A. *Economic impact of open source software on innovation and the competitiveness of the Information and Communication Technologies (ICT) sector in the EU*, MERIT, 2006. [2007].

Disponible en: http://10.33.20.200:3389/Literatura/Cientifica/SWL_English_Books/

HERNÁNDEZ, P. G. *Diccionario de la Informática*. Barcelona, Mitre, 1986. 277 p.

IVAR JACOBSON, G. B., JAMES RUMBAUGH. *El proceso unificado de desarrollo de Software*. La Habana, Felix Varela, 2004.

LARMAN, C. *UML y patrones*. La Habana, Felix Varela, 2004. 291 p.

LIBRARY, F. S. *Visual Paradigm for UML Professional Edition Details*, 2003. [2007]. Disponible en: http://textver.filehungry.com/spanish/product/windows_software/programming/specialized_tools/visual_paradigm_for_uml_professional_edition

MYSQL. *MySQL*, 2007. [2007]. Disponible en: <http://mysql.com>

PARADIGM, V. *Visual Paradigm for UML 6.0 SP1*, 2007. [2007]. Disponible en: <http://www.visual-paradigm.com/news/vpsuite30sp1/vpuml60sp1.jsp>

PARADIGM, V. *Why Visual Paradigm for UML?*, 2007. [2007]. Disponible en: <http://www.visual-paradigm.com/product/vpuml>



PHP. *PHP*, 2007. [2007]. Disponible en: <http://php.net>

PRESSMAN, R. *Ingeniería del Software*. La Habana, 2005. p.

STALLMAN, R. *Free Software: Free society*, Free Software Foundation, 2002. [2007]. Disponible en: http://10.33.20.200:3389/Literatura/Cientifica/SWL_English_Books/Free_Software_Free_Society.pdf

STALLMAN, R. *El Movimiento del Software Libre y el sistema operativo GNU/Linux*. Ciudad de la Habana, Universidad de la Habana, 2007.

STALLMAN, R. *El Movimiento del Software Libre y el sistema operativo GNU/Linux.*, Wiki.prod, 2007. [2007]. Disponible en: http://wiki.prod.uci.cu/index.php/Conferencia_de_Stallman_en_la_UH

UCI. *Efectos sociales del Software Libre*, 2006. [2007]. Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=8563>

WIKIPEDIA. *Audio*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/Audio>

WIKIPEDIA. *Bases de datos*, 2007. [2007]. Disponible en: http://es.wikipedia.org/wiki/Base_de_datos

WIKIPEDIA. *Gráficos*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/Gr%C3%A1ficos>

WIKIPEDIA. *Internet*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/Internet>.

WIKIPEDIA. *MySQL*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/MySQL>.

WIKIPEDIA. *PHP*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/Php>

WIKIPEDIA. *Programación Orientada a Objetos.*, 2006. [2007]. Disponible en: <http://es.wikipedia.org/wiki/POO>

WIKIPEDIA. *Sistema Operativo*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/SO>



WIKIPEDIA. *Software*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/software>

WIKIPEDIA. *Video*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/video>



Glosario de términos.

APACHE: Es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows , Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

CMS: Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido y el diseño.

CORBA: Es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.

DIAGRAMA: Es la representación grafica de un conjunto de elementos.

DOMINIO: Un dominio es un conjunto de ordenadores conectados en una red que confían a uno de los equipos de dicha red la administración de los usuarios y los privilegios que cada uno de los usuarios tiene en dicha red.

FIREFOX: Es un navegador de Internet, con interfaz gráfica de usuario desarrollado por la Corporación Mozilla y un gran número de voluntarios externos.

HERRAMIENTAS CASE: Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

HTML: Lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

INOCUO: Desarmado, insustancial.



JAVA: Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 1990. Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible.

LDAP: Es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

LICENCIAS GPL: Licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

LIVECD: Un LiveCD o LiveDVD (traducido en ocasiones como CDvivo o CD autónomo) es un sistema operativo (normalmente acompañado de un conjunto de aplicaciones) almacenado en un medio extraíble, tradicionalmente un CD o un DVD (de ahí su nombre), que puede ejecutarse desde éste sin necesidad de instalarlo en el disco duro del ordenador, para lo cual usa la memoria RAM como disco duro virtual y el propio medio como sistema de ficheros.

MICROKERNEL: Al tipo de kernel de un sistema operativo que provee un conjunto de primitivas o llamadas al sistema mínimas, para implementar servicios básicos como espacios de direcciones, comunicación entre procesos y planificación básica.

MODELO DE CASO DE USO DEL SISTEMA: Es un modelo del sistema que contiene actores, casos de uso y sus relaciones.

ORACLE: Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation.



OPENOFFICE: OpenOffice.org es una suite ofimática de software libre y código abierto que incluye herramientas como procesador de textos, hoja de cálculo, presentaciones, herramientas para el dibujo vectorial y base de datos.

PERL: Lenguaje Práctico para la Extracción e Informe (ver abajo) es un lenguaje de programación diseñado por Larry Wall creado en 1987. Perl toma características del C, del lenguaje interpretado shell (sh), AWK, sed, Lisp y, en un grado inferior, muchos otros lenguajes de programación.

PHISHING: Es un término informático que denomina el uso de un tipo de ingeniería social caracterizado por intentar adquirir información confidencial de forma fraudulenta (como puede ser una contraseña o información detallada sobre tarjetas de crédito u otra información bancaria).

PROTOCOLOS: Protocolo de red o también Protocolo de Comunicación es el conjunto de reglas que especifican el intercambio de datos u órdenes durante la comunicación entre las entidades que forman parte de una red.

PYTHON: Lenguaje de programación creado por Guido van Rossum en el año 1990. Es comparado habitualmente con TCL, Perl, Scheme, Java y Ruby. En la actualidad Python se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation.

RAM: Se trata de una *memoria de semiconductor* en la que se puede tanto leer como escribir información. Es una memoria volátil, es decir, pierde su contenido al desconectar la energía eléctrica.

REQUERIMIENTO: Condición que el sistema debe cumplir o capacidad que debe tener.

REPOSITORIO: Un repositorio es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos. Está preparado para distribuirse habitualmente sirviéndose de una red informática como Internet o en un medio físico como un disco compacto.



UNIX: Es un sistema operativo portable, multitarea y multiusuario; desarrollado en principio por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.



Anexos.

Anexo 1.

Clasificación para tecnologías libres.

1. Sistema Operativo.

Aplicaciones.

- ✓ Debian es una distribución libre del sistema operativo GNU/Linux.
- ✓ Suse es una distribución libre del sistema operativo GNU/Linux.
- ✓ Red Hat Linux es una distribución libre del sistema operativo GNU/Linux.
- ✓ Ubuntu es una distribución libre del sistema operativo GNU/Linux.
- ✓ Gentoo.

Parámetros.

Tipos:	General.	Técnico.	Soporte de arquitectura.	Lenguajes que soporta.
1	Creador.	Kernel de Linux.	*x86 *x86-64 *IA-64	Spanish
2	Distribución Base.	Compilador.	*ppc *ppc64 *sparc32	English
3	Licencia.	Formato de paquete.	*sparc64 *arm *hppa *mips	Portuguese.
4	Propósito.		*sh *s390 *alpha *68k	Japanese.
5				French.

2. Audio.

Sub categoría.	Aplicación 1.	Aplicación 2.	Aplicación 3.
Quemado de CD.	K3B	Gnome-Baker	NeroLinux
Reproductores CD.	Kplayer	Autorun	CD-ROM Control
MP3.	xmms	ProMP3	GNUmp3d



Editores de Audio.	Audacity.	Audio-Convert	GNU-Sound
--------------------	-----------	---------------	-----------

Parámetros: Quemado de CD.

Tipos:	General.	Técnico.
1	Versión.	Consumo de memoria virtual.
2	Licencia.	Velocidad de quemado.
3	Ambiente para el que fue diseñado.	Tiempo de borrado.
4		Tipos de Cds soportados.

Parámetros: Reproductores CD.

Tipos:	General.	Técnico.
1	Creador.	Velocidad de reproducción
2	Licencia.	Ambiente para el que fue diseñado.
3	Soporte al software.	Facilidad de trabajo

Parámetros: MP3.

Tipos:	General.	Técnico.
1	Creador.	Tipo de servicio que ofrecen.
2	Licencia.	

Parámetros: Editores de Audio.

Tipos:	General.	Técnico.
1	Creador.	Formatos soportados.
2	Licencia.	Tiempo de edición.

3. Base de Datos.

Sub categoría.	Aplicación 1.	Aplicación 2.	Aplicación 3.
Gestor de BD.	MySQL.	PostgreSQL	
Cientes de Gestores de Base de datos.	phpmyadmin	pgadmin	webmin

Parámetros: Gestores de Base de datos.



Tipos:	General.	Características fundamentales.
1	Creador.	ACID.
2	Licencia.	Unicode
3	Última versión estable.	Integridad
4		Transacciones

Parámetros: Clientes de Gestores de Base de datos.

Tipos:	General.	Características fundamentales.
1	Creador.	Gestores de BD que soportan
2	Licencia.	Posibilidades de configuración
3	Última versión estable.	

4. Gráficos.

Sub categoría.	Aplicación 1.	Aplicación 2.	Aplicación 3.
2D.	Gimp	GimpShop	KColourPaint
3D.	Blender	K3D	Quats
Editores gráficos.	CinePaint	Gimp	gpaint
Visor de imágenes.	eog	qview	imgseek

Parámetros: 2D.

Tipos:	General.	Técnicos.
1	Creador.	Formatos de imágenes soportados.
2	Licencia.	Optimización del uso de los recursos.
3	Ambiente para el que fue diseñado.	Velocidad de trabajo.

Parámetros: 3D.

Tipos:	General.	Técnicos.
1	Creador.	Optimización de los recursos del sistema.
2	Licencia.	Tipos de ficheros que exportan.
3	Facilidades de la interfaz.	Velocidad de trabajo.

Parámetros: Editores gráficos.



Tipos:	General.	Técnicos.
1	Creador.	Formatos de imágenes soportados.
2	Licencia.	Optimización de los recursos del sistema.
3	Diseño de la interfaz.	

Parámetros: Visor de imágenes.

Tipos:	General.	Técnicos.
1	Creador.	Configurabilidad de la aplicación.
2	Licencia.	Soporte para distintos formatos de imágenes.
3	Facilidades de uso.	

5. Aplicaciones Internet.

Sub categoría.	Aplicación 1.	Aplicación 2.	Aplicación 3.
Navegadores Web.	Mozilla.	Mozilla Firefox	Ephifany
Email (Cliente de correo.)	Evolution	Mozilla thunderbird	Kmail
Mensajería Instantánea.	Gaim	Gajim	Kopete

Parámetros: Navegadores Web.

Tipos:	General.	Técnicos.
1	Creador.	Soporte a las tecnologías (CSS, JavaScript, etc).
2	Licencia.	Facilidad de personalización.
3	Facilidades de uso.	Soporte para complementos.

Parámetros: Email (Clientes de correo.)

Tipos:	General.	Técnicos.
1	Creador.	Soporte de protocolos.
2	Licencia.	Soporte de configuraciones seguras.
3	Facilidades de uso.	



Parámetros: Mensajería instantánea.

Tipos:	General.	Técnicos.
1	Creador.	Soporte a distintos protocolos.
2	Licencia.	Soporte a configuraciones seguras.
3	Versión estable.	

6. Desarrollo de software.

Sub categoría.	Aplicación 1.	Aplicación 2.	Aplicación 3.
Librerías.	Zlib	GNUSASL	GNOME VFS
Compiladores.	Free Pascal Compiler.	g++	gcc
Editores.	Kdevelop IDE	Monodevelop	Bluefish
Lenguajes de programación.	C++	Java	PHP
Shell	Bash	mksh	rootsh
Control de versiones.	CVS	RapidSVN	KdeSvn

Parámetros: Librerías.

Tipos:	General.	Técnicos.
1	Creador.	Tipo de programación (Estructurada, OO)
2	Licencia.	Lenguaje de implementación.
3	Documentación.	

Parámetros: Compiladores.

Tipos:	General.	Técnicos.
1	Creador.	Lenguajes que soporta.
2	Licencia.	Presición en la detección de sintáxis erróneas.
3		

Parámetros: Editores.

Tipos:	General.	Técnicos.
--------	----------	-----------



1	Creador.	Lenguaje de programación que reconoce.
2	Licencia.	Posibilidad de autocompletamiento
3	Versión estable.	Debugging complementario.

Parámetros: Lenguajes de Programación.

Tipos:	General.	Técnicos.
1	Creador.	Aspecto al que esta orientado (Web ó Desktop)
2	Licencia.	Tipo de lenguaje
3	Versión estable.	

Parámetros: Shell.

Tipos:	General.	Técnicos.
1	Creador.	Velocidad de respuesta.
2	Licencia.	Interacción con el sistema.
3	Versión estable.	Seguridad.

Parámetros: Control de versiones.

Tipos:	General.	Técnicos.
1	Creador.	Tipo de aplicación (Cliente, Servidor).
2	Licencia.	Configurabilidad de la aplicación.
3	Versión estable disponible.	

7. Ofimática.

Sub categoría.	Aplicación 1.	Aplicación 2.	Aplicación 3.
Traductores.	Wordtrans	Stardict	
Procesadores de texto.	Abiword	OpenOffice	Gnumeric

Parámetros: Traductores.

Tipos:	General.	Técnicos.
1	Creador.	Soporte para distintos lenguajes.
2	Licencia.	Facilidad de uso y configurabilidad.



3	Última versión disponible.	
---	----------------------------	--

Parámetros: Procesadores de texto.

Tipos:	General.	Técnicos.
1	Creador.	Número de funcionalidades.
2	Licencia.	Tipos de formatos soportados.
3	Facilidad de uso.	

8. Video.

Sub categoría.	Aplicación 1.	Aplicación 2.	Aplicación 3.
Manipulación.	Cinellerra	CinePaint	Kino
Reproducción.	Mplayer	Xine	Totem

Parámetros: Manipulación.

Tipos:	General.	Técnicos.
1	Creador.	Formatos soportados
2	Licencia.	Cantidad de plugins disponibles.
3	Facilidad de uso.	Optimización de los recursos del sistema.

Parámetros: Reproducción.

Tipos:	General.	Técnicos.
1	Creador.	Cantidad de formatos soportados.
2	Licencia.	Plataformas soportadas.
3	Facilidad de configuración.	Soporte para codecs externos.

9. Herramientas de servidores.

Sub categoría.	Aplicación 1.	Aplicación 2.	Aplicación 3.
Servidores.	Bind	NeDNS	pdnsd
Directorio.	Gnome-Ldap	Open-Ladp	Gq
Email	xmail	exim	Courier
Web	Apache	boa	yhttpd

Parámetros: Servidores.



Tipos:	General.	Técnicos.
1	Creador.	Seguridad
2	Licencia.	Características en el diseño.
3	Facilidad de configuración.	Plataformas soportadas.

Parámetros: Directorio.

Tipos:	General.	Técnicos.
1	Creador.	Seguridad
2	Licencia.	Características en el diseño.
3	Facilidad de configuración.	Plataformas soportadas.

Parámetros: Email.

Tipos:	General.	Técnicos.
1	Creador.	Seguridad.
2	Licencia.	Características en el diseño.
3	Facilidad de configuración.	Plataformas soportadas.

Parámetros: Web.

Tipos:	General.	Técnicos.
1	Creador.	Seguridad
2	Licencia.	Características en el diseño.
3	Facilidad de configuración.	Plataformas soportadas