

Instituto Superior Minero Metalúrgico
“Dr. Antonio Núñez Jiménez”
Facultad Metalurgia Electromecánica.



Título:

Sistema automatizado para la Gestión del producto final de los servicios de la UEB Organización Empresarial de la Empresa Empleadora del Níquel. (SGPFS)

Trabajo de Diploma para optar por el Título de Ingeniero en Informática.

Autor: Eloy Rafael Jiménez Iglesias.

Tutor: Ing. Yeidel Cuenca Aguilar.

Co-Tutor: Ing. Yodexy Mosqueda Naranjo.

Consultantes:

Ing. Doralkis Leyva Blázquez.

Moa-Holguín

2009



Agradecimientos.

Dedicatoria.

Declaración de Autoría

Declaro que soy el único autor del trabajo de diploma titulado

y que el mismo pertenece a la Empresa Empleadora del Níquel, para que hagan el uso que estime pertinente con este trabajo.

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

Firma del autor

Firma del tutor

Opinión del tutor del trabajo de diploma.

Título del trabajo de diploma: _____

Tutor del trabajo de diploma: _____

Considero que el estudiante se encuentra listo para ejercer como Ingeniero Informático, y se le propone la calificación de ____ puntos.

Firma el presente a los ____ días del mes de _____ del año ____.

Firma del tutor

Opinión del usuario del trabajo de diploma.

El Trabajo de Diploma, titulado: “**Sistema automatizado para la Gestión del producto final de los servicios**”, fue realizado en la Empresa Empleadora del Níquel. Este centro considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

- Totalmente
- Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta Empresa los beneficios siguientes:

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____

Representante de la entidad

Cargo

Firma

Cuño

Resumen

Con el avance que ha devenido en los últimos años de las tecnologías de la información, las computadoras se han ganado un papel protagónico en los diferentes procesos de intercambio de archivos e informatización de los diversos sistemas.

En la UEB Organización Empresarial se mueve un importante volumen de documentación proveniente de los servicios que se prestan en la unidad a diversas empresas, estos documentos son manipulados de forma manual por parte de las personas que allí laboran. La búsqueda y recuperación de la información se ha convertido en un tema de vital importancia, por tal razón es necesaria la creación de un sistema automatizado que permita la gestión del producto de los servicios de la UEB Organización Empresarial.

Con el desarrollo de esta investigación se propone la informatización de la gestión del producto final de los servicios, a través de la creación de una aplicación Web que lo gestione de manera ágil y segura.

Para la realización de la investigación se realizó una revisión bibliográfica sobre las aplicaciones Web y las herramientas para la construcción de las mismas. En este documento se recoge un resumen del estudio bibliográfico realizado, se presenta la metodología de Ingeniería de Software que se siguió para la construcción del software que se propone como solución de la problemática encontrada.

Abstract

With the advance that has become in the last years of the technologies of the information, the computers have been won a protagonistic paper in the different processes of exchange of files and informatización of the diverse systems.

In the UEB Managerial Organization she moves an important documentation volume coming from the services that they are lent in the unit to diverse companies, these documents they are manipulated in a manual way on the part of people that there work. The search and recovery of the information has become a topic of vital importance, for such a reason it is necessary the creation of an automated system that allows the administration of the product of the services of the UEB Managerial Organization.

With the development of this investigation he intends the informatización of the administration of the final product of the services, through the creation of an application Web that negotiates it in an agile and sure way.

For the realization of the investigation he was carried out a bibliographical revision on the applications Web and the tools for the construction of the same ones. In this document a summary of the carried out bibliographical study is picked up, the methodology of Engineering of Software is presented that was continued for the construction of the software that intends as solution of the opposing problem.

Índice

Introducción	1
Capítulo 1 Metodologías y Herramientas	5
1.1 Introducción.	5
1.2 Conceptos Fundamentales.....	5
1.3 Herramientas.	5
1.4 Sistemas de Gestión de Bases de Datos (SGBD).	11
1.5 Lenguajes de Programación para la Web.....	13
1.6 Software Libre.....	18
1.7 Servidores para aplicaciones web.	18
1.8 Metodologías de desarrollo de software.	20
1.9 Patrones Arquitectónicos.	25
1.10 Tecnologías a utilizar.	31
1.11 Arquitectura a utilizar.	33
1.12 Herramientas a utilizar en la propuesta de solución.	34
1.13 Metodología a utilizar para la propuesta de solución.	38
1.14 Conclusiones.	40
Capítulo 2 Características del Sistema	41
2.1 Introducción.	41
2.2 Descripción de los procesos del negocio propuesto.....	41
2.3 Descripción de los Actores y Trabajadores del Negocio.	43
2.3.1 Actores del Negocio.	43
2.3.2 Trabajadores del Negocio.	43
2.4 Modelo de Casos de Uso del Negocio.	43
2.4.1 Descripción de los casos de uso del Negocio.	44
2.5 Captura de Requisitos.	48
2.5.1 Requisitos funcionales.	48
2.5.2 Requisitos no funcionales.	51
2.6 Modelo de C/U del sistema.	53
2.6.1 Actores del sistema para automatización.	54
2.6.2 Descripción de los casos de uso del Sistema.	56
2.7 Conclusiones.	59
Capítulo 3 Análisis y Diseño	60
3.1 Introducción.	60
3.2 Análisis.....	60
3.2.1 Diagramas de clases del análisis.	60
3.3 Diseño.	66
3.3.1 Diagramas de clases del diseño.....	66
3.4 Diagrama de clases persistentes.	74
3.5 Diagrama de despliegue.	75
3.6 Conclusiones.	76

Capítulo 4 Estudio de Factibilidad y Sostenibilidad	77
4.1 Introducción.	77
4.2 Estudio de Factibilidad.	77
4.3 Análisis de Sostenibilidad.	81
Conclusiones Generales.	83
Recomendaciones.	84
Referencias Bibliograficas:	85
Anexos	87

Introducción.

Con la Era Digital y el auge alcanzado por el mercado de productos informáticos a nivel mundial, así como la dependencia creada por las diferentes ramas de la sociedad hacia esos productos y servicios, se ha hecho necesaria la creación de sistemas que mejoren y faciliten el uso de la información. Los sistemas informáticos para la gestión de información son de los software más comunes utilizados a nivel mundial, sobre todo los sistemas para el manejo de grandes volúmenes de información. Entiéndase por “Gestión” la acción de Guardar, Modificar, Eliminar o Generar, algún tipo de reporte para mostrar. La causa fundamental de este auge se debe a la necesidad en todo el mundo de mantener todo tipo de información guardada de forma segura. Esta necesidad se remonta desde los inicios del hombre, donde se escribía en piedra y posteriormente en los papiros. En la actualidad existen 2 formas de guardar la información:

1. La antigua de forma manual o formato duro.
2. De forma digital mediante sistemas automatizados.

Nuestro país también se ha visto envuelto en esta necesidad de guardar de forma segura todo tipo de archivos. Puesto que el entorno que nos rodea nos mueve de forma constante a la par del mundo.

La Empleadora del Níquel es una empresa que se encarga de prestar servicios de Organización Empresarial a las empresas del níquel y a terceros. La UEB Organización Empresarial es la encargada de prestar estos servicios entre los que se encuentran además los de Psicología y Seguridad Industrial.

De forma más específica los servicios de organización empresarial generan gran cantidad de documentación. Todos estos documentos se realizan de forma manual y se guardan en formato duro. Esto crea el inconveniente de que al paso del tiempo los volúmenes de papeles van creciendo sin límites. A su vez se presenta la problemática de que estos archivos se vayan perdiendo o deteriorando. Los clientes pueden pedir anualmente un mismo servicio para un área determinada, si se pierde la información referente a este no se puede hacer una revisión de si se han cumplido los objetivos propuestos. Por esta causa es necesaria una forma segura de guardarla.

A pesar de ser una empresa de servicios esta no cuenta con una forma de que sus clientes puedan revisar las prestaciones anteriores recibidas. Esto trae consigo que los clientes no puedan documentarse sobre los estudios realizados a su empresa. La lentitud de la propagación de la información es una realidad, pues los clientes no pueden informarse en tiempo real como va un servicio que se este llevando a cabo.

A partir de esta situación surge el **problema científico**:

- No se cuenta con una forma segura de guardar la información.
- El cliente no tiene acceso a servicios anteriores.
- Si se pierde la información no hay forma de comprobar si se han cumplido las expectativas con el servicio prestado anteriormente.

El **objeto** de esta investigación es:

- Los servicios de Organización Empresarial que se realizan en la UEB O. Empresarial de la Empleadora del Níquel.

Dicho objeto enmarca el **campo de acción** de esta investigación es:

- El producto final de los servicios de O. Empresarial que presta la UEB Organización Empresarial de la Empresa Empleadora del Níquel.

El **objetivo** principal de esta investigación es:

- Desarrollar un sistema automatizado que permita la Gestión de la información del producto final de la UEB Organización Empresarial.

Los **objetivos específicos** que se persiguen son:

- Elaborar el Marco teórico conceptual.
- Realizar un diagnóstico sobre los procesos y servicios de la UEB Servicios de Organización Empresarial de la Empresa Empleadora del Níquel.
- Realizar análisis, diseño e implementación del sistema para lograr un mejor manejo de la información.

- Documentar el sistema automatizado.

Idea a defender:

- Si se crea un sistema automatizado que posibilite la Gestión de la información, se mejora la calidad de planificación del control de prestaciones y la organización del ciclo de vida del servicio.

Para lograr el objetivo propuesto se han trazado una serie de **tareas** a realizar:

- Investigar los distintos servicios que se prestan en la empresa.
- Buscar referencias bibliográficas.
- Indagar cómo se genera y guarda la información de los servicios.
- Estudiar las condiciones tecnológicas disponibles en la empresa y en el país para el desarrollo de la aplicación.
- Seleccionar las tecnologías a utilizar de acuerdo a la tarea anterior.
- Realizar el modelo del negocio y levantamiento de requisitos.
- Elaborar el modelo de análisis y diseño del sistema.
- Efectuar la implementación de la aplicación.
- Confeccionar el Manual de Usuario.

Para complementar estas tareas se han empleado **métodos** teóricos y empíricos de la investigación científica. Entre los métodos empíricos usados podemos citar la observación, entrevista y el análisis de documentos para la recopilación de la información. La observación se utilizó para ver la funcionalidad de la UEB y el comportamiento del problema. La entrevista; donde se realizó una conversación planificada con el fin de obtener información individual o colectiva y determinar los principales requerimientos del sistema. Mediante el análisis de documentos se supo como funcionan actualmente los procesos de la UEB.

Los métodos teóricos proporcionaran calidad en la investigación. Entre los teóricos se uso el método histórico y lógico donde se realiza la búsqueda de antecedentes que pueda tener el software, si existe actualmente alguno en funcionamiento con las mismas características, la

situación actual de los software de gestión y sus tendencias y las herramientas más potentes y usadas en el mundo para su confección. En el desarrollo del proceso de investigación se usaron el análisis y síntesis para la recopilación y el procesamiento de la información obtenida en los métodos empíricos y arribar a las conclusiones de la investigación. El hipotético deductivo utilizó en la elaboración de la hipótesis y para su verificación. La modelación permitió realizar un estudio en la gestión de la información de la UEB.

Para cumplir con las tareas propuestas se han definido y seguido los siguientes pasos lógicos:

- Se realizó un estudio de las distintas metodologías de desarrollo de software, de las herramientas, tecnologías y patrones arquitectónicos para hacer una selección de las mejores, para utilizar en la confección del software de acuerdo a sus características.
- Se contextualizó el problema, enmarcando los objetivos específicos que el cliente deseaba.
- Se realizó el análisis y diseño del software con todas las especificaciones requeridas.
- Se efectuó un Estudio de Factibilidad y Sostenibilidad.

1.1 Introducción.

Hoy en día la informática va evolucionando considerablemente y junto con ella la producción de los diversos software. El siguiente capítulo es el resultado de una investigación acerca de las herramientas y metodologías existentes con las cuales se le pudiera dar solución a nuestro problema planteado.

1.2 Conceptos Fundamentales.

La *ingeniería de software (ISW)* afecta a la economía y las sociedades de muchas maneras. Dicho término se refiere al área de las ciencias de la computación que trata con la construcción de sistemas de software, los cuales son tan grandes y complejos que se construyen con equipos de ingenieros. Es un proceso definido paso a paso, que facilita la especificación, el diseño, la implementación y las pruebas de una solución de software, para un conjunto de requisitos explícitos, de modo eficiente y eficaz. Esto requiere que antes de empezar se tenga: objetivos claros, planes para lograr los objetivos, procedimientos que implementan los planes, un ambiente conducente al logro de los objetivos.

La *gestión de software (GS)* es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos; garantizando la calidad del producto. La GS comprende un conjunto de gestiones, en los cuales se encuentra la gestión de configuración y la gestión de proyecto.

1.3 Herramientas.

Herramientas CASE

CASE es una sigla, que corresponde a las iniciales de: Computer Aided Software Engineering; y en su traducción al Español significa Ingeniería de Software Asistida por Computadora. El concepto de CASE es muy amplio; y una buena definición genérica, que pueda abarcar esa amplitud de conceptos, sería la de considerar a la Ingeniería de Software Asistida por Computadora (CASE), como la aplicación de métodos y técnicas, las cuales son

útiles a las personas para comprender las capacidades de las computadoras, por medio de programas de procedimientos y su respectiva documentación.

Tipos de herramientas CASE

No existe una única clasificación de herramientas CASE y, en ocasiones, es difícil incluirlas en una clase determinada. Podrían clasificarse atendiendo a:

1. Las plataformas que soportan.
2. Las fases del ciclo de vida del desarrollo de sistemas que cubren.
3. La arquitectura de las aplicaciones que producen.
4. Su funcionalidad.

Las herramientas CASE en función de las fases del ciclo de vida que abarcan, se pueden agrupar de la forma siguiente:

- Herramientas integradas, I-CASE (Integrated CASE, CASE integrado): abarcan todas las fases del ciclo de vida del desarrollo de sistemas. Son llamadas también CASE workbench.
- CASE (WORKBENCH): un conjunto de herramientas integradas diseñadas para trabajar conjuntamente y automatizar (o proporcionar una ayuda automatizada) totalmente el ciclo de vida del software, que incluye el análisis, el diseño, la codificación y la prueba.

Herramienta(s) que comprende(n) alguna(s) fase(s) del ciclo de vida de desarrollo de software:

- Herramientas de alto nivel, U-CASE (Upper CASE - CASE superior o front-end) orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo: análisis y diseño.
- Herramientas de bajo nivel, L-CASE (Lower CASE - CASE inferior o back-end) dirigidas a las últimas fases del desarrollo: desarrollo e implantación.
- Juegos de herramientas o toolkits: son el tipo más simple de herramientas CASE. Automatizan una fase dentro del ciclo de vida. Dentro de este grupo se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento.

Las herramientas I-CASE se basan generalmente en una metodología. Tienen un repositorio (BD del proyecto) y aportan técnicas para todas las fases del ciclo de vida. Sin embargo, no

todas ellas son modernas en el sentido de aprovechar la potencia de las estaciones de trabajo o la utilización de lenguajes de alto nivel o técnicas de construcción de prototipos.

Una alternativa posible a los I-CASE es utilizar una U-CASE para análisis y diseño, combinada con otras herramientas más modernas para las fases de desarrollo y pruebas. En este caso, habría que vigilar cuidadosamente la integración entre las distintas herramientas.

Otra posible clasificación, utilizando la funcionalidad como criterio principal, es la siguiente:

- Herramientas de planificación de sistemas de gestión: sirven para modular los requisitos de información estratégica de una organización. Proporcionan un "metamodelo" del cual se pueden obtener sistemas de información específicos. Su objetivo principal es ayudar a comprender mejor cómo se mueve la información entre las distintas unidades organizativas. Estas herramientas proporcionan una ayuda importante cuando se diseñan nuevas estrategias para los sistemas de información y cuando los métodos y sistemas actuales no satisfacen las necesidades de la organización.
- Herramientas de Análisis y Diseño: permiten al desarrollador crear un modelo del sistema que se va a construir y también la evaluación de la validez y consistencia de este modelo. Proporcionan un grado de confianza en la representación del análisis y ayudan a eliminar errores con anticipación. Entre ellas podemos encontrar:
 1. Herramientas de análisis y diseño (Modelado).
 2. Herramientas de creación de prototipos y de simulación.
 3. Herramientas para el diseño y desarrollo de interfaces.
- Herramientas de programación: se engloban aquí los compiladores, los editores y los depuradores de los lenguajes de programación convencionales. Ejemplos de estas herramientas son:
 1. Herramientas de codificación convencionales.
 2. Herramientas de codificación de cuarta generación (asociadas a SGBD)
 3. Herramientas de programación orientadas a objetos.

- Herramientas de integración y prueba: sirven de ayuda a la adquisición, medición, simulación y prueba de los equipos lógicos desarrollados. Entre las más utilizadas están:
 1. Herramientas de análisis estático.
 2. Herramientas de generación de casos de prueba.
- Herramientas de gestión de prototipos: los prototipos son utilizados ampliamente en el desarrollo de aplicaciones, para la evaluación de especificaciones de un sistema de información, o para un mejor entendimiento de cómo los requisitos de un sistema de información se ajustan a los objetivos perseguidos.
- Herramientas de mantenimiento: la categoría de herramientas de mantenimiento se puede subdividir en:
 1. Herramientas de Ingeniería Inversa.
 2. Herramientas de reestructuración y análisis de código.
 3. Herramientas de reingeniería.
- Herramientas de gestión de proyectos: la mayoría de las herramientas CASE de gestión de proyectos, se centran en un elemento específico de la gestión del proyecto, en lugar de proporcionar un soporte global para la actividad de gestión. Utilizando un conjunto seleccionado de las mismas se puede: realizar estimaciones de esfuerzo, coste y duración, hacer un seguimiento continuo del proyecto, estimar la productividad y la calidad, etc. Existen también herramientas que permiten al comprador del desarrollo de un sistema, hacer un seguimiento que va desde los requisitos de condiciones técnicas iniciales, hasta el trabajo de desarrollo que convierte estos requisitos en un producto final. Se incluyen dentro de las herramientas de control de proyectos las siguientes:
 1. Herramientas de planificación de proyectos.
 2. Herramientas de seguimiento de requisitos.
 3. Herramientas de gestión y medida.
- Herramientas de soporte: se engloban en esta categoría las herramientas que recogen las actividades aplicables en todo el proceso de desarrollo, como las que se relacionan a continuación:

1. Herramientas de documentación.
2. Herramientas para software de sistemas.
3. Herramientas de control de calidad.
4. Herramientas de bases de datos

Rational Rose: Es una herramienta software para el modelado visual mediante Unified Modeling Language (UML) de sistemas software, permite especificar, analizar, diseñar el sistema antes de codificar. Mantiene la consistencia de los modelos del sistema software, además de la generación de documentos automáticamente. Otra funcionalidad de la herramienta es que permite realizar la ingeniería inversa de un producto.

Visual Paradigm: Es una herramienta CASE que utiliza “UML”: como lenguaje de modelado.

Se integra con las siguientes herramientas Java:

- Eclipse/IBM WebSphere
- JBuilder
- NetBeans IDE
- Oracle JDeveloper
- BEA Weblogic

Permite realizar ingeniería tanto directa como inversa, es capaz de desplegar todas las clases asociadas a las tablas (siguiendo el patrón de diseño Una Clase-Una Tabla). Para gestionar la persistencia y el mapeo de estas clases con la base de datos utiliza Hibernate para Java y NHibernate en el caso de un proyecto .Net. Además, la herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos como Web o .Pdf, y permite realizar control de versiones. Cabe destacar igualmente su robustez, usabilidad y portabilidad.

Easy CASE: Esta herramienta permite automatizar las fases de análisis y diseño dentro del desarrollo de una aplicación, para poder crear las aplicaciones eficazmente desde

procesamiento de transacciones a la aplicación de bases de datos de cliente/servidor, así como sistemas de tiempo real. Permite capturar los detalles de diseño de un sistema y comunicar las ideas gráficamente, para que sean fáciles de ver y entender. Para un diseño legítimo y modelado de datos, procesos y eventos, permite crear y mantener diagramas de flujo de datos, diagramas de entidad-relación y mapas de estructura.

MS Project: Microsoft Project es un potente programa propietario de gestión de proyectos que se utiliza y demanda cada vez más por parte de las empresas, para crear planes de proyecto, introducción de datos reales de evolución y realizar un completo seguimiento de cada una de sus partes, así como contabilizar la variación que se produce en el transcurso de un proyecto respecto a lo que inicialmente se había programado (línea base). Con este programa se gestionan y controlan tanto las tareas que componen un proyecto, como los recursos que se utilizan para su desarrollo, y las asignaciones recurso-tarea.[1]

Gantt Project: Software de escritorio, desarrollado en Java, nos permite dividir nuestro proyecto en un árbol de tareas y asignar recursos humanos a cada una de ellas, establecer dependencias entre tareas, generar informes PDF y HTML, intercambiar datos con MS Project y hojas de cálculo. Es una herramienta gratuita para crear una completa planificación de un proyecto de forma visual. Un punto interesante es que permite establecer dependencias entre las tareas, de esta forma, una tarea no podrá empezar hasta que esté acabada la anterior. Permite exportar el trabajo a una imagen (JPG, PNG), PDF y HTML.[2]

Open Workbench: Es una aplicación de código abierto que proporciona un robusto sistema de planificación y funcionalidades de gestión. Es una alternativa gratuita a Microsoft Project. Ofrece un avanzado sistema de planificación de proyectos, con tareas (y tareas clave) y con una eficaz estructura de trabajo. Una vez creado el proyecto, se podrá, asignar recursos y utilizar herramientas de planificación, ejecución, control y seguimiento.[3]

1.4 Sistemas de Gestión de Bases de Datos (SGBD).

Los sistemas de bases de datos están diseñados para gestionar grandes volúmenes de información. Generalmente, las bases de datos requieren gran cantidad de espacio de almacenamiento, por lo que las bases de datos de las organizaciones se miden en términos de gigabytes o terabytes de datos. Un gigabyte equivale a 1000 megabytes (un billón de bytes), y un terabyte equivale a un millón de megabytes (un trillón de bytes). Un sistema de bases de datos tiene como objetivo simplificar y facilitar el acceso a los datos y hacer que los tiempos de respuesta a las solicitudes de los usuarios sean muy reducidos.

De forma sencilla, un sistema de gestión de bases de datos se puede definir como una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. Adoración de Miguel lo define como un conjunto coordinado de programas, procedimientos, lenguajes, etc. que suministra, tanto a los usuarios no informáticos como a los analistas, programadores o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad.[4]

Microsoft SQL Server

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, Sybase ASE, PostgreSQL o MySQL.[5]

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. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

- Soporte de transacciones.
- Estabilidad.

- Escalabilidad.
- Es multiplataforma.

Ha sido criticada por algunos especialistas la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre de 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux.[6]

MySQL Server.

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. MySQL AB pertenece a Sun Microsystems desde enero de 2008. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero las empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C. Al contrario que proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual. MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.[7]

Es muy fácil de instalar y administrar y tiene una gran comunidad de usuarios. Obtener información sobre cómo realizar cualquier tarea sobre MySQL es bastante sencillo gracias a la gran comunidad de usuarios que tiene. Es muy indicada para iniciarse en el mundo de las bases de datos puesto que dispone de una infinidad de utilidades, tutoriales y documentación.

El gran propulsor de la base de datos MySQL ha sido sin duda el lenguaje interpretado para Web PHP. MySQL ha crecido al mismo paso que ha crecido la comunidad de PHP. Cuando

llegó la popularización del Linux y de PHP como plataforma para aplicaciones Web de bajo coste se encontró que la única base de datos gratuita decente del momento era MySQL pues en ese momento PostgreSQL 6.X era muy lento y pesado. Hoy muchos sitios Web populares usan MySQL como SGBD.

PostgreSQL

Es un Sistema Gestor de Bases de Datos, liberado bajo la licencia BSD o Berkeley Software Distribution permite el uso del código fuente en software no libre, posee una serie de características positivas respecto a otros gestores.

1. Gran escalabilidad. Es ajustable al número de procesadores y a la cantidad de memoria que posee el sistema de forma eficiente, por este motivo es capaz de soportar una mayor cantidad de peticiones simultáneas.
2. Tiene la capacidad de almacenar procedimientos almacenados en la propia base de datos.
3. Multiusuario, con arquitectura cliente-servidor y control de privilegios de acceso.
4. Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.
5. La velocidad del código del motor de datos ha sido incrementada aproximadamente en un 20% - 40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fue lanzada.

1.5 Lenguajes de Programación para la Web.

Con la aparición de Internet, la programación web ha adquirido disimiles avances, mediante el trascurso de los años, producto al desarrollo de los lenguajes de programación para la web que facilitaron y propiciaron un desarrollo vertiginoso hasta nuestros días. Dichos lenguajes se clasifican en dos partes fundamentales que reconocen la propia arquitectura Cliente/Servidor de esta plataforma de desarrollo: los lenguajes del lado del Servidor y los lenguajes del lado del Cliente.

Entre los lenguajes del lado del servidor se puede encontrar PHP, ASP, Perl, JSP, C++, Python. Estos se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a Bases de Datos, tratamiento de la Información etc. Del lado del cliente se encuentran XHTML, AJAX, CSS, JScript que son totalmente independientes del servidor, son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento.

Esta distinción en los lenguajes ha sido necesaria debido a que la Web funciona en modo "Desconectado", o sea, un usuario a través de un navegador hace una petición de una página Web a un Servidor Web (Request), el Servidor recepciona la petición, la procesa y le envía la Respuesta al Cliente(Response), este la recepciona y se desconecta.

XHTML (eXtensible HyperText Markup Language)

El lenguaje XHTML es muy similar al lenguaje HTML. De hecho, XHTML no es más que una adaptación de HTML al lenguaje XML. Técnicamente, HTML es descendiente directo del lenguaje SGML, mientras que XHTML lo es del XML (que a su vez, también es descendiente de SGML). [8]

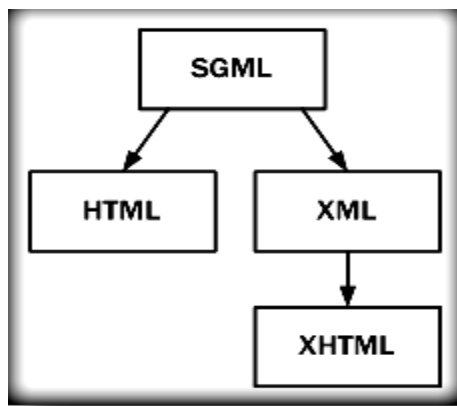


Figura 1.1:

CSS (Cascading Style Sheets)

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Mientras que el lenguaje HTML/XHTML se utiliza para marcar los contenidos, es decir, para designar lo que es un párrafo, lo que es un titular o lo que es una lista de

elementos, el lenguaje CSS se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, la tabulación con la que se muestran los elementos de una lista, etc.[9]

Javascript

Javascript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con Javascript podemos crear diferentes efectos e interactuar con nuestros usuarios. Este lenguaje posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros. Es necesario resaltar que hay dos tipos de JavaScript: por un lado está el que se ejecuta en el cliente, este es el Javascript propiamente dicho, aunque técnicamente se denomina Navigator JavaScript. Pero también existe un Javascript que se ejecuta en el servidor, es más reciente y se denomina LiveWire Javascript. [10]

ASP (Active Server Pages)

El ASP es una tecnología dinámica funcionando del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del script son ejecutadas para enviar al navegador únicamente el código HTML resultante. La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que éste se encuentra únicamente en los archivos del servidor que al ser solicitado a través del web, es ejecutado, por lo que los usuario no tienen acceso más que a la página resultante en su navegador.[11]

PERL

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para el web. Perl es un acrónimo de Practical Extracting and Reporting Language, que viene a indicar que

se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros. Es un lenguaje libre de uso, eso quiere decir que es gratuito. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows[12]. Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como Javascript o ASP. Esto quiere decir que el código de los scripts en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito. Además es extensible a partir de otros lenguajes, ya que desde Perl podremos hacer llamadas a subprogramas escritos en otros lenguajes. También desde otros lenguajes podremos ejecutar código Perl.

JSP (JavaServer Pages)

Es una tecnología orientada a crear páginas web con programación en Java. Con JSP podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. El motor de las páginas JSP está basado en los servlets de Java -programas en Java destinados a ejecutarse en el servidor-, aunque el número de desarrolladores que pueden afrontar la programación de JSP es mucho mayor, dado que resulta mucho más sencillo aprender que los servlets.[13]

Python

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.[14]

PHP

PHP (Personal Home Page) es el acrónimo de Hypertext Preprocessor. Es un lenguaje de programación del lado del servidor gratuito y multi plataforma, rápido, con una gran librería de funciones y mucha documentación. Es también un lenguaje interpretado y embebido en el HTML. Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 6, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades de las aplicaciones Web actuales. PHP, en el caso de estar montado sobre un servidor Linux o Unix, es más rápido que ASP, dado que se ejecuta en un único espacio de memoria y esto evita las comunicaciones entre componentes COM que se realizan entre todas las tecnologías implicadas en una página ASP. PHP es la gran tendencia en el mundo de Internet. Últimamente se puede observar un ascenso imparable, ya que cada día son muchísimas más las páginas Web que lo utilizan para su funcionamiento, según las estadísticas, PHP se utiliza en más de 10 millones de páginas, y cada mes realiza un aumento del 15%. Resumiendo, el PHP corre en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte sobre unas 20 Bases de Datos y contiene unas 40 extensiones estables sin contar las que se están experimentando, además de que:

Es software libre, lo que implica menos costes y servidores más baratos que otras alternativas.

Es muy rápido. Su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.

Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si se está familiarizado con esta sintaxis, le resultara muy fácil aprender PHP.

Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP.

PHP tiene una de las comunidades más grandes en Internet, con lo que no es complicado encontrar ayuda, documentación, artículos, noticias, y más recursos.

1.6 Software Libre.

El término software libre es tan viejo como las mismas computadoras, puesto que no se puede hablar del hardware sin mencionar el software ya que estos van de la mano. En los inicios de la era de las computadoras, años 60, los programadores compartían el código para ayudarse entre sí, ya no era considerado un producto sino una vía que los productores daban a sus clientes para que pudieran utilizar su hardware, hasta que las compañías productoras de computadoras en los años 70 decidieron convertir el código en un producto comercial.

Software libre o código abierto (open source) es el software que su código puede ser modificado, copiado y redistribuir libremente. Existe la tendencia a confundir el término libre a gratis, el hecho de que un software pueda ser modificado no significa que siempre se pueda descargar libremente de Internet. Existen compañías que venden sus productos y permiten que se le pueda estudiar y modificar su código, otras como Ubuntu sus CDs son gratis lo que cobran son los servicios, etc.

El software libre se define de la siguiente forma:

- Permite copiar el programa.
- Se permite ejecutar el programa con cualquier propósito, ya sea privado, público, comercial, etc.
- Posibilita estudiar y modificar el código del programa.
- Mejorar el programa y realizar redistribuciones del mismo.

1.7 Servidores para aplicaciones web.

Internet Information Services (IIS)

Es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web).[15]

Zope

Es un servidor de aplicaciones web de código abierto escrito en el lenguaje de programación Python. Puede ser manejado casi totalmente usando una interfaz de usuario basada en páginas Web.

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

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

Lo más característico de Zope es su base de datos orientada a objetos, llamada ZODB o Zope Object Database. Esta base de datos almacena objetos ordenados en un sistema similar a un sistema de ficheros, pero cada objeto tiene propiedades, métodos u otros objetos. Esta aproximación es muy diferente de las base de datos relacionales habituales. Sin embargo, Zope dispone de múltiples conectores para las diferentes bases de datos relacionales y ofrece sistemas básicos de conexión y consulta abstrayéndolos como objetos.[16]

Apache

El Apache fue hecho para proveer un alto grado de calidad y fortaleza para las implementaciones que utilizan el protocolo HTTP. Está ligado a la plataforma (Linux, Windows, UNIX) sobre la cual los individuos o instituciones pueden construir sistemas confiables con fines experimentales o para resolver un problema específico de la organización.

El Apache es un software libre, porque sus desarrolladores defienden la teoría de que las transmisiones usando la red deben estar en las manos de todos, y que las compañías de software deben hacer el dinero ofertando servicios con valor añadido tales como módulos especializados, soportes, entre otros, y no siendo dueñas de un protocolo. Así, el proyecto de crear una implementación robusta con referencia absolutamente libre para quien lo quiera usar es un buen paso para evitar la propiedad sobre los protocolos.

1.8 Metodologías de desarrollo de software.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Las técnicas indican cómo debe ser realizada una actividad técnica determinada identificada en la metodología. Se debe tener en consideración que una técnica determinada puede ser utilizada en una o más actividades de la metodología de desarrollo de software. Todo desarrollo de software es riesgoso y difícil de controlar, pero si no lleva una metodología de por medio, lo que se obtiene es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica ¿qué es lo que hay que obtener? a lo largo del desarrollo del proyecto pero no, cómo hacerlo. Indica cómo hay que obtener los distintos productos parciales y finales.

Aquí se describen las principales características de tres de las más famosas y conocidas metodologías de desarrollo de software: Proceso Unificado de Racional (Rational Unified Process, RUP), Programación Extrema (Extreme Programming, XP) y Microsoft Solution Framework (MSF).

Rational Unified Process (RUP)

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:



Figura 1.2: Fases de la Metodología RUP.

- **Inicio:** El Objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción:** En esta etapa el objetivo es obtener la capacidad operacional inicial.
- **Transición:** El objetivo es llegar a obtener la versión lista para su instalación en las condiciones reales.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo:

- **Modelado empresarial:** Entendiendo las necesidades del negocio.
- **Requerimientos:** Traslado de las necesidades del negocio a un sistema automatizado.
- **Análisis y Diseño:** Traslado de los requerimientos dentro de la arquitectura de software.
- **Implementación:** Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- **Pruebas:** Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.
- **Despliegue:** Describe las actividades asociadas al garantizar que el producto de software esté disponible para los usuarios.
- **Gestión de cambios y configuración:** Explica cómo controlar y sincronizar la evolución del conjunto de productos de trabajo que componen un sistema de software.

- **Gestión de proyecto:** Planificación del proyecto, la gestión del riesgo, la supervisión del progreso y la métrica.
- **Entorno:** Organiza esos elementos de método que proporcionan el entorno de desarrollo de software que da soporte al equipo de desarrollo, incluidos los procesos y las herramientas.

Distribución: Hacer todo lo necesario para la salida del proyecto

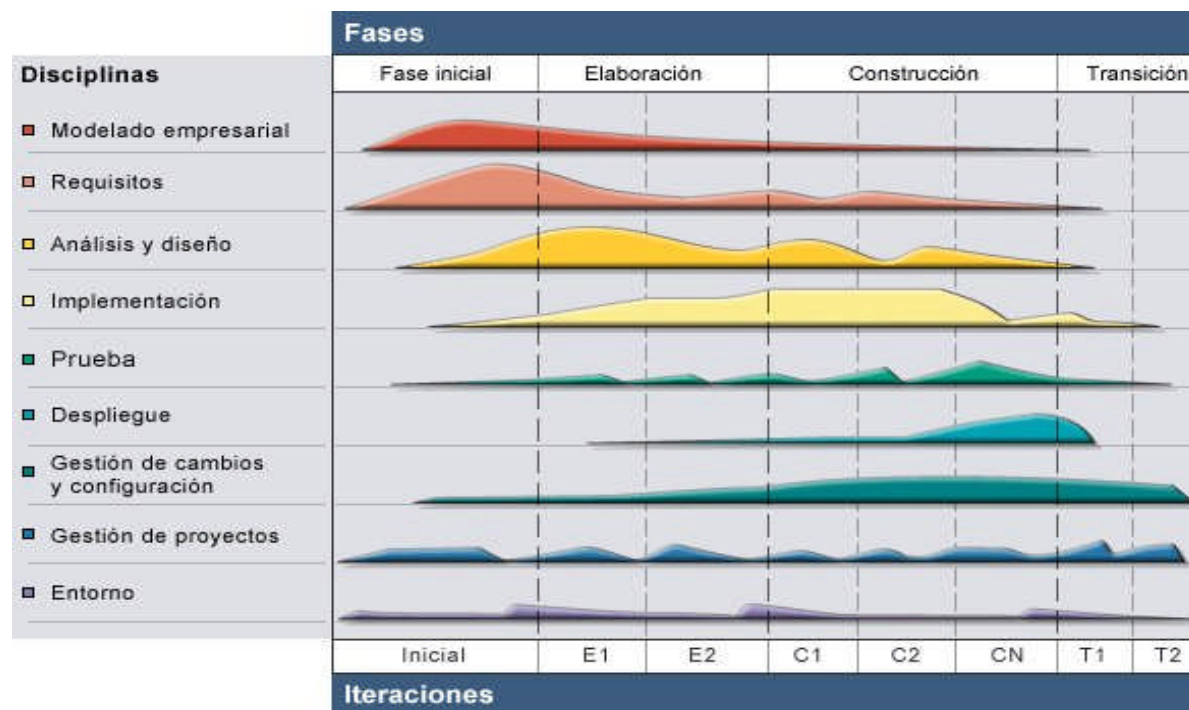


Figura 1.3: Fases e Iteraciones de la Metodología RUP.

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierta luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendrá en cada entregable o en cada iteración.

Los elementos del RUP son:

Actividades: Son los procesos que se llegan a determinar en cada iteración.

Trabajadores: Son las personas o entes involucrados en cada proceso.

Artefactos: Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

Extreme Programming (XP)

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

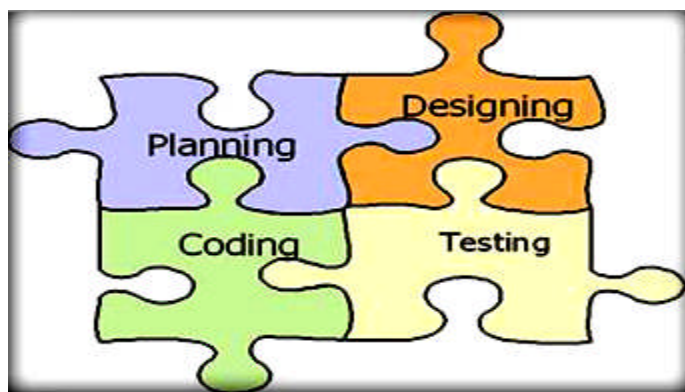


Figura 1.4: Metodología Extreme Programming.

La metodología XP se basa en:

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

¿Qué es lo que propone XP?

- Añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo, Derechos del Cliente.
- Saber el estado real y el progreso del proyecto.
- Añadir, cambiar o quitar requerimientos en cualquier momento.
- Obtener lo máximo de cada semana de trabajo.
- Decidir como se implementan los procesos.
- Crear el sistema con la mejor calidad posible.
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- Estimar el esfuerzo para implementar el sistema.
- Cambiar los requerimientos en base a nuevos descubrimientos.

Lo fundamental en este tipo de metodología es:

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

Microsoft Solution Framework (MSF)

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



Figura 1.5: Metodología MSF.

MSF tiene las siguientes características:

- **Adaptable:** es altamente adaptable, lo que permite entregar soluciones de alta calidad y de manera rápida mientras que minimiza el riesgo asociado a todo proyecto ya que éste considera las causas más comunes que hacen fracasar los proyectos de tecnología.
- **Escalable:** puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren aproximadamente 50 personas.
- **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.
- **Tecnología Agnóstica:** puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

1.9 Patrones Arquitectónicos.

Patrón de Arquitectura MVC (Modelo Vista Controlador)

Para el diseño de aplicaciones con sofisticados interfaces se utiliza el patrón de diseño Modelo-Vista-Controlador. La lógica de un interfaz de usuario cambia con más frecuencia

que los almacenes de datos y la lógica de negocio. Si realizamos un diseño ofuscado, es decir, una forma de mezclar los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando necesitemos cambiar la interfaz, tendremos que modificar trabajosamente los componentes de negocio. Mayor trabajo y más riesgo de error.

Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Elementos del patrón:

- **Modelo:** Datos y reglas de negocio.
- **Vista:** Muestra la información del modelo al usuario.
- **Controlador:** Gestiona las entradas del usuario.

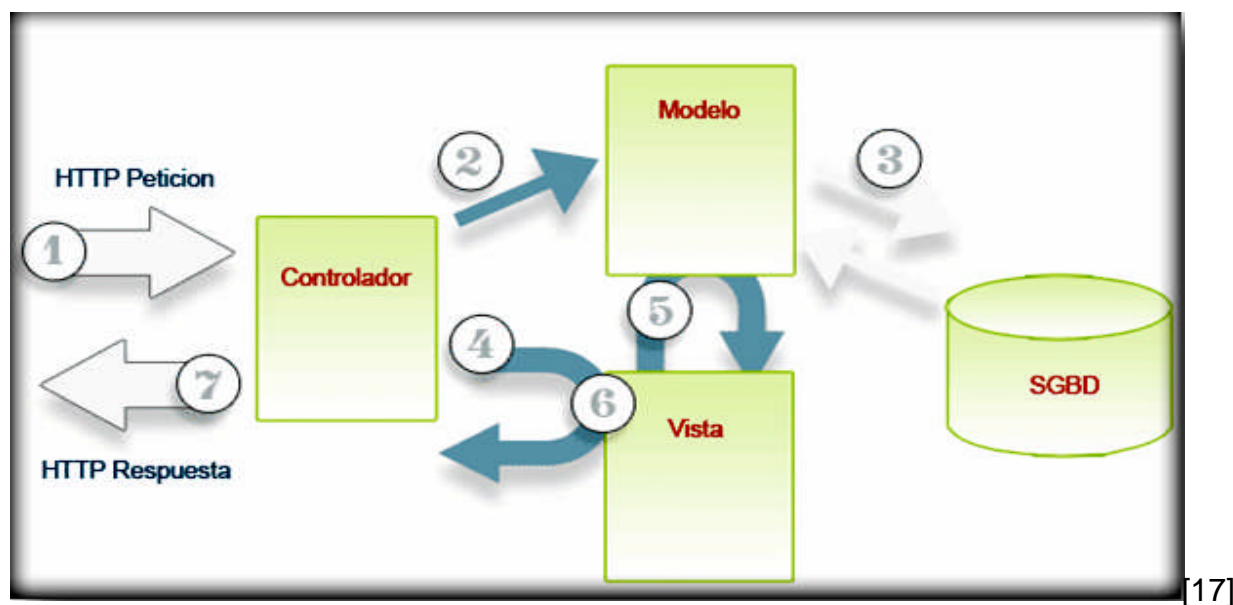


Figura 1.6: Modelo Vista Controlador.

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador. Un ejemplo clásico es el de la información de una base de datos, que se puede presentar de diversas formas: diagrama de pastel, de barras, tabular, etc. Analizan cada componente:

1. El **modelo** es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.

- Definir las reglas de negocio (la funcionalidad del sistema). (Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor").
- Llevar un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero bat que actualiza los datos, un temporizador que desencadena una inserción, etc).

2. El **controlador** es responsable de:

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada a actualizar. (Una petición al modelo puede ser "Obtener_tiempo_de_entrega dado una nueva orden").

3. Las **vistas** son responsables de:

- Recibir datos del modelo y mostrarlo al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de actualizar, para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

Arquitecturas en Capas

Los sistemas o arquitecturas en capas constituyen uno de los estilos que aparecen con mayor frecuencia mencionados como categorías mayores del catálogo o por el contrario, como una de las posibles encarnaciones de algún estilo más envolvente. (Garlan y Shaw) definen el estilo en capas como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior.

La arquitectura por capas es un estilo de arquitectura en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario.

La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.

Además permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de los niveles, simplemente es necesario conocer las *API* que existen entre niveles.

En el diseño de sistemas informáticos actuales se suele usar las arquitecturas multinivel o **Programación por capas**. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten). El diseño más en boga actualmente es el diseño en tres niveles (o en tres capas).

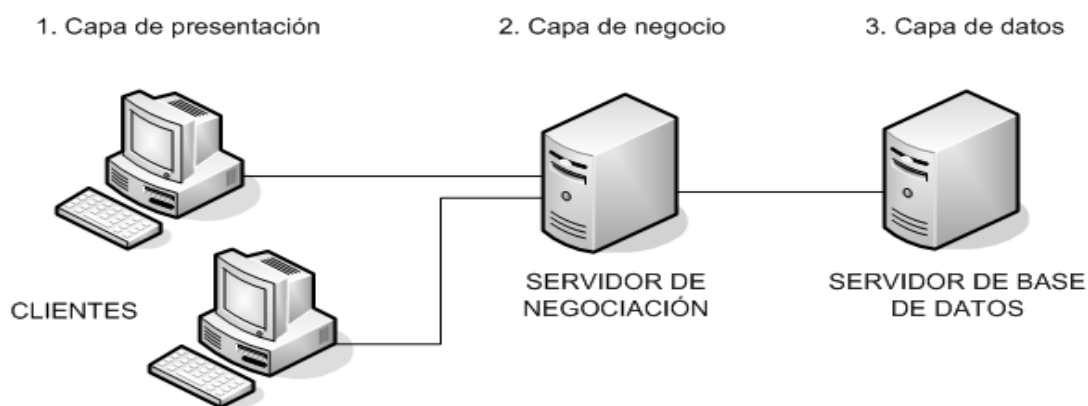


Figura 1.7. Arquitectura en Capa.

Capas o niveles

- **Capa de presentación o interfase:** Es la capa que le permite al usuario interactuar con el sistema, captura y le comunica la información al mismo, dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.
- **Capa de negocio o lógica:** Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se

denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

- **Capa de datos:** Es donde se ubican los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de los mismos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en un único ordenador (no sería lo normal), si bien lo más usual es que haya una multitud de ordenadores donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio. Si por el contrario fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en uno o más ordenadores que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de ordenadores sobre los cuales corre la capa de datos y otra serie de ordenadores sobre los cuales corre la base de datos.

En una arquitectura de tres niveles, los términos "capas" y "niveles" no significan lo mismo ni son similares. El término "capa" hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: Presentación/ Lógica de Negocio/ Datos.

En cambio, el término "nivel", corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física. Por ejemplo:

Una solución de tres capas (presentación, lógica, datos) que residen en un solo ordenador (Presentación+lógica+datos). Se dice, que la arquitectura de la solución es de tres capas y *un nivel*.

Una solución de tres capas (presentación, lógica, datos) que residen en dos ordenadores (presentación+lógica, lógica+datos). Se dice que la arquitectura de la solución es de tres capas y *dos niveles*.

Una solución de tres capas (presentación, lógica, datos) que residen en tres ordenadores (presentación, lógica, datos). La arquitectura que la define es: solución de tres capas y *tres niveles*.

Ventajas:

- El estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite a los implementadores la partición de un problema complejo en una secuencia de pasos incrementales.
- El estilo admite muy naturalmente optimizaciones y refinamientos.
- Proporciona amplia reutilización. Al igual que los tipos de datos abstractos, se pueden utilizar diferentes implementaciones o versiones de una misma capa en la medida que soporten las mismas interfaces de cara a las capas adyacentes. Esto conduce a la posibilidad de definir interfaces de capa estándar, a partir de las cuales se pueden construir extensiones o prestaciones específicas.

Desventajas:

- No admiten un buen *mapeo* en una estructura jerárquica. Incluso cuando un sistema se puede establecer lógicamente en capas, consideraciones de rendimiento pueden requerir acoplamientos específicos entre capas de alto y bajo nivel.
- A veces es también extremadamente difícil encontrar el nivel de abstracción correcto, por ejemplo, la comunidad de comunicación ha encontrado complejo mapear los protocolos existentes en el framework *ISO*, de modo que muchos protocolos agrupan diversas capas, ocasionando que en el mercado proliferen los *drivers* o los *servicios monolíticos*.
- Los cambios en las capas de bajo nivel tienden a filtrarse hacia las de alto nivel, en especial si se utiliza una modalidad relajada; también se admite que la arquitectura en

capas ayuda a controlar y encapsular aplicaciones complejas, pero complica no siempre razonablemente las aplicaciones simples.

1.10 Tecnologías a utilizar.

PHP v5.0 o superior

Las iniciales PHP significan "PHP Hypertext Pre-processor y se trata de un lenguaje de programación que es usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. Es un lenguaje de programación usado generalmente para la creación de contenido para sitios o aplicaciones Web. La versión 5 de PHP presenta un magnífico trabajo con el Paradigma Orientado a Objeto que permite la reutilización de código entre otras facilidades.

Ventajas de trabajar con PHP comparado con otros lenguajes similares:

- Es un lenguaje multiplataforma.
- Rapidez de ejecución.
- Mantiene un bajo consumo de recursos de máquina.
- Gran seguridad, muy poca probabilidad de corromper los datos.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Posee una amplia documentación en Internet, incluyendo una gran variedad de ejemplos y de ayudas.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear los formularios para la Web.
- No requiere definición de tipos de variables ni manejo detallado de bajo nivel.

AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas se ejecutan en el cliente, es decir, en el navegador de los usuarios y

mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

AJAX es una combinación de tres tecnologías ya existentes:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje script por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos asíncronicamente con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON(JavaScript Object Notation) y hasta EBML(Meta Lenguaje Binario Extendible).

Como el DHTML (HTML Dinámico), LAMP (se refiere a un conjunto de subsistemas software necesarios para alcanzar una solución global, en este caso configurar sitios web o Servidores dinámicos con un esfuerzo reducido), AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

Ventajas:

- Recuperación asíncrona de datos, el usuario no tiene que esperar después de una petición.
- Acercamiento del ambiente de escritorio a la Web.
- No requiere plugins (es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica).
- Se reduce el tamaño de la información intercambiada.

MySQL

MySQL es muy rápido, fiable y fácil de usar, surge para manipular bases de datos muy grandes. Es un sistema multiplataforma de base de datos relacionales, lo que da velocidad y flexibilidad, cuenta con un sistema de privilegios contraseñas muy seguro que permite la autenticación básica para el acceso al servidor. MySQL es un sistema de administración de Base de Datos. Opera en una arquitectura cliente/servidor. Es el sistema gestor de bases de datos "Open Source" más popular, o sea que puede ser bajado de Internet y usarlo sin tener que pagar, además que cualquiera puede estudiar su código y adecuarlo a las necesidades que requiera.

Luego de analizadas las características y facilidades de los SGBD presentados, y las de la herramienta a desarrollar se decide usar el MySQL como SGBD, por las siguientes razones:

- No se necesitará de un manejo complejo de la información.
- El PHP maneja más fácil al MySQL que al SQL Server, debido a la gran cantidad de funciones que tiene explícitas.
- El MySQL es multiplataforma.
- El MYSQL no tiene precio en el mercado, se adquiere libremente.

Apache

Es el servidor Web más utilizado en el mundo con un 57 % de cuota de mercado, frente al 20 % de Microsoft IIS y el 7 % de Netscape. Dentro de sus puntos fuertes se encuentran:

- Tiene interfaz con todos los sistemas de autenticación.
- Facilita la integración como "plug-ins" de los lenguajes de programación de páginas Web dinámicas más comunes.
- Tiene integración en estándar del protocolo de seguridad SSL.(más utilizado).
- Provee interfaz a todas las bases de datos.

1.11 Arquitectura a utilizar.

Arquitectura

Una arquitectura es el conjunto de decisiones significativas sobre la organización del SS, la selección de los elementos estructurales y sus interfaces, con los que se compone el

sistema, junto con su comportamiento tal como se especifica en las colaboraciones entre esos elementos, la composición de esos elementos estructurales y de comportamiento en subsistemas progresivamente más amplios, y el estilo de arquitectura que guía esta organización, estos elementos y sus interfaces, sus colaboraciones, y su composición. Ej.: los Patrones de diseño relacionados con el diseño de los objetos y frameworks de pequeña y mediana escala, que son aplicables al diseño de una solución para conectar los elementos de gran escala que se definen mediante los patrones de arquitectura, y durante el trabajo de diseño detallado para cualquier aspecto del diseño local. También se conocen como patrones de micro-arquitectura. El patrón Fachada, que se puede utilizar para proporcionar la interfaz de una capa a la siguiente.

La Arquitectura en Capas o Arquitectura n capas es la que se ha seleccionado para darle solución a la problemática planteada anteriormente, debido a las facilidades presentadas anteriormente.

1.12 Herramientas a utilizar en la propuesta de solución.

Zend Studio v5.5

Zend Studio 5.5 es un Integrated Development Environment (IDE) Incluye un poderoso editor de código con soporte para todos los formatos de la Web. Es un entorno de desarrollo para PHP, que integrando el uso y completado de código personalizado de Zend Framework y vista de la lista de las funciones del framework desde la Visualización de funciones PHP; utiliza una fuerte herramienta para la documentación de código llamada PHPDocumentor , lo cual proporciona que el código este mas entendible.[18]

Zend Platform

Zend Platform servidor de aplicación para PHP compatible con el rendimiento, administración, integración, utilizando PHP para sus aplicaciones de negocios críticas. Mediante la inclusión de capacidades que hacen más eficientes el desarrollo y despliegue, Zend Platform mejora la experiencia del usuario, la respuesta de la aplicación, integración a su infraestructura existente y aumenta la fiabilidad y escalabilidad de la aplicación.[19]

Dreanweaver 8

Dreamweaver 8 es la opción profesional para la creación de sitios y aplicaciones web.

Proporciona una combinación potente de herramientas visuales de disposición, permite desarrollo de aplicaciones y soporte para la edición de código. Gracias a las robustas características para la integración y diseño basado en CSS. Incluye potentes controles basados en normas para asegurar un diseño de alta calidad. Un entorno de diseño construido en torno a las hojas de estilo en cascada (CSS) hace posible un desarrollo más rápido y más eficiente de sitios profesionales creados con código limpio. Es un entorno de desarrollo integrado para desarrollar sitios web de HTML, XHTML, XML, ASP, ASP.NET, JSP, PHP y Macromedia ColdFusion.[20]

Fireworks 8

Fireworks es un programa práctico, flexible e innovador que incorpora todo tipo de herramientas para el tratamiento de imágenes de mapa de bits y gráficos vectoriales. Incluye nuevas funciones y mejoras de diseño y desarrollo, máxima integración con la Suite de Macromedia 8 y otros programas de diseño y editores HTML. Destacan la creación y exportación de menús emergentes utilizando CSS, dando completo control sobre el código sin necesidad de volver a programarlo, mejoras en el flujo de trabajo con los nuevos paneles de acceso rápido de edición de imágenes e inserción de caracteres y nuevos formatos de imagen admitidos para guardar e importar.[21]

Rational Rose Enterprise

Rational Rose Enterprise proporciona un lenguaje de modelado común para permitir la creación más rápida de la calidad de software. Incluye Unified Modeling Language (UML) de apoyo y es una de la más completa de productos de la familia Rational Rose. Proporciona la base de datos para el modelado UML diseños, con la capacidad de representar a la integración de los datos y los requisitos de las aplicaciones a través de diseños físicos y lógicos. Crea definiciones XML de tipo de documento (DTD) para uso en su aplicación. Los sistemas operativos soportados: HP Unix, Linux, Windows. Rational Rose ayuda a las empresas a superar la paradoja de software por la unificación de las comunidades que

desarrollan software con un lenguaje visual que acelera la creación de aplicaciones robustas.[22]

Xampp

Es un servidor independiente de plataforma, software libre, incluye el servidor Web Apache, los servidores de datos MySQL y SQLite, sus respectivos gestores phpMyAdmin y phpSQLiteAdmin, el intérprete del lenguaje homónimo PHP con los extras incluidos en PEAR, el intérprete del lenguaje Perl, servidores de FTP como ProFTPD ó FileZilla FTP Server, las estadísticas Webalizer y OpenSSL, eAccelerator, Freetype2, libjpeg, libpng, zlib, Ming, etc. entre otros. El programa esta liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP esta disponible para Microsoft Windows, GNU/Linux, Solaris y MacOS X.

Oficialmente, los diseñadores de XAMPP solo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios Webs y programadores testar su trabajo en sus propios ordenadores sin ningún acceso a Internet. En la práctica sin embargo, XAMPP es utilizado actualmente para servidor de sitios Web, y con algunas modificaciones es generalmente lo suficientemente seguro para serlo. Con el paquete se incluye una herramienta especial para proteger fácilmente las partes más importantes.

Ventajas:

- Incluye Chequeo de seguridad.
- Contiene un Panel de control.
- Incluye herramientas extras (Webalizer, Mercury Mail, Conmutador PHP).[23, 24]

Embarcadero ER/Studio 6.0.1

Es una herramienta de modelado de datos, se usa para el diseño y la construcción lógica y física de bases de datos. Su ambiente es de gran alcance y multinivel. ER/Studio se diseña para hacer más fácil de entender el estado actual de los datos de las empresas. Simple y fácil al usuario, ayuda a organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia y alcanza en última instancia usos de más alta calidad que entreguen datos más eficientes y exactos a la empresa. [25, 26]

- **Potencial de ER/Studio:** Si se está comenzando un nuevo diseño o está manteniendo una base de datos existente, ER/Studio se combina con las características para ayudarle a conseguir el trabajo hecho con eficacia. Con el potencial y la facilidad de empleo de ER/Studio's que modela el ambiente, será productivo rápidamente y podrá casi demostrar resultados inmediatamente después de la instalación.
- **Diagramas:** La creación de diagramas es clara y rápida. Tiene la posibilidad de realizar diagramas con desempeño rápido. También es posible cambiar el estilo de las líneas, los colores, tipos de letra, niveles de acercamiento, y modelos de despliegue. Es posible crear subvistas para separar y manejar áreas importantes. ER/Studio automáticamente mantiene todas las dependencias entre subvistas y el diagrama completo. El Explorer Navigation facilita el trabajo hasta con los diagramas más grandes. Si se está trabajando con un modelo largo de Datos, ER/Studio ofrece un aumento en la ayuda y fácil navegación en sus modelos. La Apreciación global (overview). Se usa el browser Explorer para encontrar y seleccionar entidades. Un solo clic inmediatamente enfoca una ventana de diagrama. [26]

Ventajas:

- Soporta el proceso de diseño interactivo inherente en el ciclo de vida de la aplicación.
- ER/Studio puede documentar automáticamente un diagrama entero, generando un conjunto integrado de reportes HTML sofisticados que múltiples usuarios pueden compartir en Internet.
- Calidad de presentación en los reportes. Además de los reportes de HTML, ER/Studio puede generar reportes de alta calidad con un formato de texto amplio que esta disponible para presentaciones profesionales.
- Soporta metodología de Yourdon, con diagramas relación-entidad y modelos IDEF1.

1.13 Metodología a utilizar para la propuesta de solución.

Después de realizar un análisis e investigación de estas tres metodologías, se elige utilizar RUP para el desarrollo del software, ya que es la más completa y abarcadora, pues como señalan algunos autores, las otras metodologías son casos particulares de esta. Además, XP y MSF presentan algunas debilidades, lo que representa riesgos considerables, como dificultades a la hora de una buena obtención de requisitos para el sistema. RUP es adaptable, según el tipo de proyecto, así serán las características del mismo, haciéndose énfasis en aquellos flujos de trabajo durante la vida del software, que reporten más importancia y sean indispensables. RUP permite trazarse planes de riesgos y pruebas durante el ciclo de vida. Contiene artefactos que son diseñados durante las diferentes fases, los cuales describen detalladamente las características del software desde que se realiza el análisis del problema hasta la entrega final del producto.

UML

El Proceso Unificado RUP (Racional Unified Process) utiliza el Lenguaje de Modelado UML (Unified Modeling Language) para preparar todos los esquemas de un sistema software. UML es una parte esencial del Proceso Unificado, esta consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

UML recomienda 9 diagramas que son modelados durante las diferentes fases de desarrollo del software, es decir, representan las siguientes vistas del sistema:

- **Diagrama de Casos de Uso:** modela la funcionalidad del sistema agrupándola en descripciones de acciones ejecutadas por un sistema para obtener un resultado.
- **Diagrama de Clases:** muestra las clases que componen el sistema y cómo se relacionan entre sí.
- **Diagrama de Objetos:** muestra una serie de objetos y sus relaciones.
- **Diagrama de Secuencia:** enfatiza la interacción entre los objetos y los mensajes que intercambian entre sí junto con el orden temporal de los mismos.

- **Diagrama de Colaboración:** igualmente, muestra la interacción entre los objetos resaltando la organización estructural de los objetos en lugar del orden de los mensajes intercambiados.
- **Diagrama de Estados:** modela el comportamiento de acuerdo con eventos.
- **Diagrama de Actividades:** simplifica el Diagrama de Estados modelando el comportamiento mediante flujos de actividades.
- **Diagrama de Componentes:** muestra la organización y las dependencias entre un conjunto de componentes.
- **Diagrama de Despliegue:** muestra la organización y las dependencias entre un conjunto de componentes. [1]

1.14 Conclusiones.

En este capítulo quedaron reflejados todos los conceptos relacionados con el problema, ampliando así los conocimientos para comprender mejor los procesos de negocio que ocurren en la entidad. Se hizo un análisis del estado del arte, las tecnologías y herramientas informáticas, procesos de desarrollo de software, lenguajes de programación y tendencias que fundamentan la solución propuesta.

Capítulo 2

Características del Sistema.

2.1 Introducción.

Antes de adentrarse en el desarrollo de un sistema, el primer paso en el proceso de elaboración de un software, es precisamente alcanzar cierto nivel de conocimiento sobre el problema en cuestión. Se debe ser capaz de comprender la estructura y la dinámica de la organización en la cual se va a implementar nuestro sistema, comprender los problemas actuales de la organización e identificar las mejoras potenciales, asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización y derivar los requerimientos del sistema que va a soportar la organización. Para poder lograr esos propósitos, el proceso de modelado permite obtener una visión de la organización que permita definir los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos. Se debe ser capaz de analizar el estado del negocio, es decir, determinar si los procesos involucrados están claramente definidos.

2.2 Descripción de los procesos del negocio propuesto.

Para describir los procesos del negocio que se relacionan con el campo de acción de este trabajo, es necesario centrar la atención en los procesos que se llevan a cabo en la UEB y sus necesidades actuales.

Para entender mejor como se lleva a cabo los procesos antes mencionados dentro de la UEB Organización Empresarial se impone el conocimiento del uso de las diversas metodologías de desarrollo de software, en el caso que compete, la metodología RUP. Los servicios de la UEB para su realización pasan por **fases**:

Cuando un cliente presenta problemas en su empresa y desea mejorar el rendimiento de la misma, condiciones y calidad de su trabajo, **solicita** que se le preste algún servicio de la

UEB Organización Empresarial, este debe realizar la solicitud a la Dirección de producción de la Empresa, que es la encargada de la contratación de servicios o directamente a la UEB Organización Empresarial. Luego de tramitar la solicitud se realiza una **entrevista** en la cual se define:

- El tipo de servicio.
- La UEB en la que dicha empresa desea realizar el servicio.
- Los puntos y objetivos que el cliente necesita comprobar.

Una vez definidos los puntos anteriores la dirección de la UEB Organización Empresarial redacta y presenta una **oferta** en correspondencia a las necesidades del cliente.

En caso de que el cliente este de acuerdo con esta propuesta prosigue la fase **Presentación** el Especialista principal designará al Jefe de Grupo, especialistas y técnicos que realizaran el servicio. La siguiente fase seria la de **Familiarización**, durante esta fase el equipo de trabajo realiza lo siguiente:

- Recogida de datos.
- Involucramiento del personal.
- Estudio preliminar de los puestos y áreas.
- Recogida de plantilla u otros documentos necesarios para la ejecución del servicio.
- Identificar, verificar, proteger y salvaguardar los bienes que son propiedad del cliente.

Después de este periodo que tiene un tiempo promedio de tres días de duración se pasa a **Definir y revisar los métodos y las técnicas a utilizar**, una vez concluida esta fase se pasa a la siguiente, **Elaboración del cronograma de trabajo** en la cual se define el tiempo que necesita cada tarea y la fecha de entrega del servicio. Una vez realizadas estas fases pasan a ejecutarse las siguientes fases:

- **Aplicación de métodos y técnicas escogidas.**
- **Reportar incidencias en el registro.**
- **Realización del despacho de los servicios.**
- **Elaboración del informe.**
- **Revisión del informe.**
- **Declaración de conformidad.**
- **Discusión del informe al cliente.**

El objetivo primordial de esta empresa es:

- Establecer el procedimiento de trabajo a seguir en los servicios técnicos organizativos que se ejecutan en el Grupo de Organización.

2.3 Descripción de los Actores y Trabajadores del Negocio.

2.3.1 Actores del Negocio.

Tabla 2.1: Actores del negocio.

Actores del Negocio	Justificación.
Cliente	Persona que solicita un servicio.

2.3.2 Trabajadores del Negocio.

Tabla 2.2: Trabajadores del negocio.

Trabajadores del Negocio	Justificación.
Especialista Principal	Persona encargada de elaborar la oferta y crear el equipo de trabajo.
Jefe de Equipo	Persona encargada de la dirección de los equipos de trabajo de la realización de los servicios.
Dirección UEB	Grupo de personas encargadas de entrevistarse con los clientes.
Equipo de Trabajo	Persona o Grupo de personas encargadas de realizar los servicios en las empresas.

2.4 Modelo de Casos de Uso del Negocio.

El modelo del negocio describe el proceso de negocio en términos de casos de uso, que corresponde a lo que generalmente se le llama procesos. El modelo de casos de uso del negocio describe los procesos de un negocio (casos de uso del negocio) y su interacción con elementos externos (actores), tales como socios y clientes.

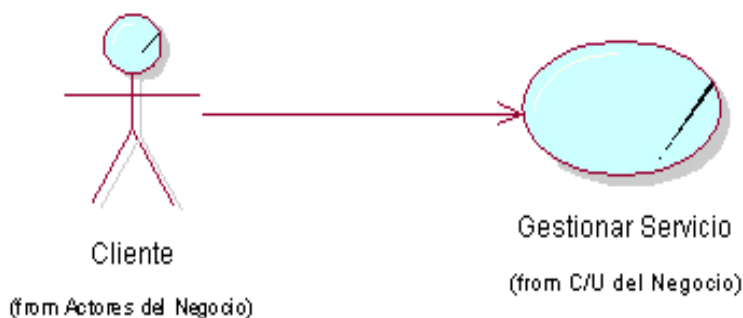


Figura 2.1: Modelo de C/U del Negocio.

2.4.1 Descripción de los casos de uso del Negocio.

Tabla 2.3 Descripción del caso de uso Solicitar Servicio.

Caso de uso:	Gestionar Servicio
Actores del negocio:	Cliente.
Trabajadores del negocio:	Especialista Principal, Jefe de equipo, Dirección, Equipo de trabajo.
Resumen:	El caso de uso se inicia cuando el cliente se dirige a la UEB a solicitar un servicio determinado. El caso de uso finaliza cuando se entrega el producto final.
Acción del actor	Respuesta del proceso de negocio
1- El Cliente solicita el servicio.	1.1- La Especialista principal, la Dirección de la UEB o la Especialista en Marketing reciben la solicitud.
	1.2- El Especialista principal se entrevista con el cliente.
	1.3- La Dirección de la UEB le realiza una Oferta al cliente.
	1.4- Si el cliente acepta la oferta el Especialista Principal conforma el equipo de trabajo, sino Sesión

	1.
	1.5- El Equipo de trabajo pasa a la Familiarización con el área donde se prestará el servicio.
	1.6- El Equipo de trabajo define los métodos y técnicas a utilizar.
	1.7- El Especialista a cargo del equipo realiza el cronograma de trabajo.
	1.8- El Equipo aplica los métodos y técnicas definidas.
	1.9- El Jefe de equipo a cargo reporta las incidencias ocurridas.
	1.10- El Especialista principal realiza un despacho con el cliente para explicarle en que etapa se encuentra el servicio.
	1.11- El Equipo de trabajo redacta el informe.
	1.12- El Especialista principal revisa el informe, si no cumple con las condiciones necesarias se pasa a Sesión 2 .
	1.13- El especialista principal redacta la Declaración de conformidad y la discute con el cliente. Si el cliente no esta de acuerdo se pasa a Sesión 3 .
	1.14- El Equipo de trabajo realiza la discusión del Informe con el cliente en el área donde se trabajo.
Mejoras:	Con la automatización de la captación del producto final de los servicios, se actualizan los datos, para realizar revisiones y comparaciones posteriores para determinar si se han

	cumplido con los objetivos propuestos.
Cursos Alternos:	
Sesión 1. El Especialista principal trata de buscar soluciones. 1.1 Reestructura la oferta o se niega a prestar el servicio.	
Sesión 2. El Especialista principal orienta redactar nuevamente el informe. 2.1 Si una vez redactado cumple con los requisitos, se pasa a la Declaración de Conformidad.	
Sesión 3. El Cliente se queja a la Dirección de la UEB. 3.1 Se realizan la revisión de causa raíz.	

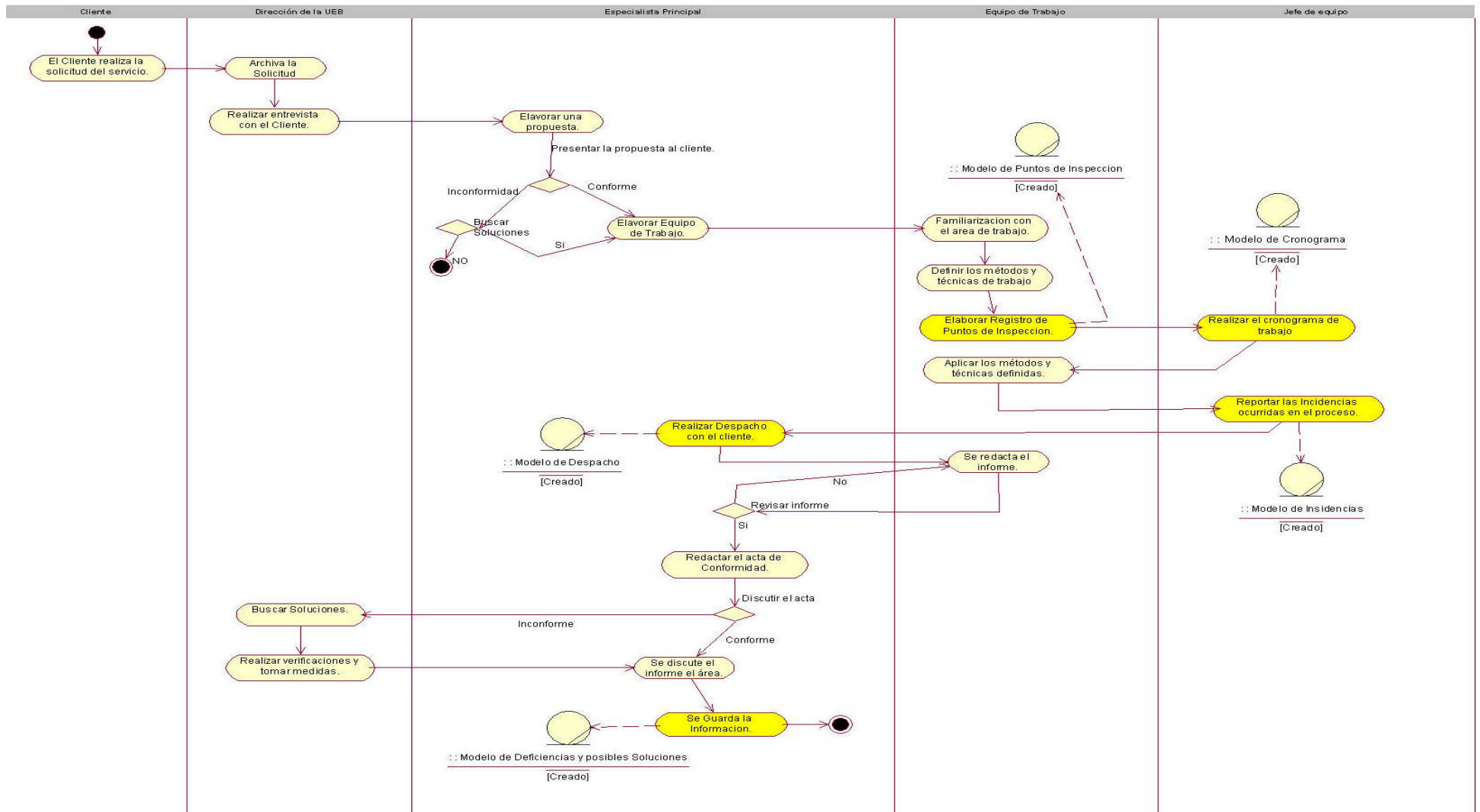


Figura 2.2: Diagrama de Actividades.

2.5 Captura de Requisitos.

Los requerimientos son una descripción de las necesidades o deseos de un cliente o un producto, su meta es identificar y documentar de una forma clara para el cliente y para el equipo de desarrollo lo que en realidad se necesita. Para la captura de requisitos de este proyecto se contó con la colaboración de los usuarios finales, la dirección y el cliente.

2.5.1 Requisitos funcionales.

Una vez conocidos los conceptos que rodean al objeto de estudio, se puede comenzar a analizar ¿Qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo?, para ello enumeramos a través de requerimientos funcionales las funciones 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:

R1 Autenticarse.

R2 Registrar Técnicas y Métodos.

- 2.1 Registrar el nombre de una Técnica o Método.

R3 Registrar un Servicio.

- 3.1 Registrar el nombre del servicio.

R4 Registrar Empresa.

- 4.1 Registrar el nombre de la empresa.
- 4.2 Registrar el código de la empresa.
- 4.3 Registrar el nombre del área de la empresa.

R5 Gestionar Información.

- 5.1 Permitir a los clientes tener acceso a los servicios que se le han prestado.
- 5.2 Mostrar un resumen de un servicio determinado.

R6 Gestionar el cronograma de trabajo del servicio.

- 6.1 Registrar la fecha en la que se realiza.
- 6.2 Registrar el tipo de servicio que se va a realizar.
- 6.3 Registrar la empresa a la que se le a prestar el servicio.
- 6.4 Registrar las actividades que se realizaran.
- 6.5 Registrar las fechas de inicio y fin de cada actividad.
- 6.6 Registrar una aclaración en caso de dudas.
- 6.7 Adicionar un Cronograma de Trabajo.
- 6.8 Modificar un Cronograma de Trabajo.

R7 Gestionar Modelo Deficiencias y Posibles Soluciones.

- 7.1 Registrar el nombre del servicio que se presta.
- 7.2 Registrar el nombre de la empresa a la que se le presta el servicio.
- 7.3 Registrar cantidad de trabajadores abarcados en el estudio.
- 7.4 Registrar los principales resultados obtenidos.
- 7.5 Registrar las deficiencias detectadas en el servicio.
- 7.6 Registrar las posibles soluciones.
- 7.7 Registrar el nombre de la persona que realiza el servicio.
- 7.8 Adicionar un Modelo Deficiencias y Posibles Soluciones.
- 7.9 Eliminar un Modelo Deficiencias y Posibles Soluciones.

R8 Gestionar Control de las Incidencias de los Servicios.

- 8.1 Registrar el nombre del cliente.
- 8.2 Registrar el nombre del servicio.
- 8.3 Registrar las actividades realizadas en la semana.
- 8.4 Registrar las dificultades en el desarrollo de las actividades.
- 8.5 Registrar la fecha en la que se realiza.
- 8.6 Registrar el nombre del responsable de informar.
- 8.7 Registrar el nombre del que receptiona la información.
- 8.8 Adicionar el Control de las Incidencias de los servicios.

8.9 Modificar el Control de las Incidencias de los servicios.

R9 Gestionar Acta del Despacho.

- 9.1 Registrar la fecha de realización del despacho.
- 9.2 Registrar el nombre de la empresa con la que se hace el despacho.
- 9.3 Registrar el lugar donde se ejecuta.
- 9.4 Registrar los nombre, cargo y empresa de las personas que participan.
- 9.5 Registrar aspectos a analizar.
- 9.6 Registrar situación del despacho (Cumpliéndose S/N).
- 9.7 Registrar dificultades en el desempeño de las actividades.
- 9.8 Registrar los acuerdos tomados.
- 9.9 Registrar el código de la empresa.
- 9.10 Adicionar Acta del despacho.
- 9.11 Modificar Acta del despacho.

R10 Gestionar Registro de los Puntos de Inspección.

- 10.1 Registrar el código de la empresa.
- 10.2 Registrar el nombre de la empresa.
- 10.3 Registrar el nombre del servicio.
- 10.4 Registrar el nombre del especialista que realiza el servicio.
- 10.5 Registrar la información recibida.
- 10.6 Clasificar la información en (B/R/M).
- 10.7 Registrar las técnicas escogidas.
- 10.8 Evaluar las técnicas escogidas (B/R/M).
- 10.9 Registrar los nombres de las personas que participan.
- 10.10 Registrar la fecha en que se realiza.
- 10.11 Registrar el nombre de quien informa.
- 10.12 Registrar el nombre de quien evalúa.
- 10.13 Adicionar Registro de los Puntos de Inspección.
- 10.14 Modificar Registro de los Puntos de Inspección.

R11 Conformar un Servicio a Empresa.

- 11.1 Registrar fecha de inicio del servicio.
- 11.2 Registrar el servicio.
- 11.3 Registrar la empresa.
- 11.4 Registrar el área.
- 11.5 Registrar el monto del servicio en CUC y en CUP.

R12 Guardar Documento.

- 12.1 Permitir guardar los documentos en el servidor.

2.5.2 Requisitos no funcionales.

Los requerimientos no funcionales como su nombre indica, son propiedades o cualidades que el producto debe tener. Estas propiedades se ven como las características que hacen al producto agradable, usable, rápido y confiable. A continuación se definen los siguientes:

Apariencia o interfaz externa.

- La interfaz a implementar debe ser sencilla, para que los usuarios que no son personas expertas en la rama de la informática, no necesiten mucho tiempo de adiestramiento.
- Por el uso diario y constante que tendrá el software, la interfaz debe ser agradable, que favorezca el estado de ánimo del cliente y que combine correctamente los colores, tipo de letra y tamaño.
- Deben utilizarse plantillas con el mismo estilo.

Usabilidad.

- El sistema debe ser de fácil manejo para los usuarios que tengan niveles básicos sobre la computación o trabajo con la Web.
- Debe tener una opción de ayuda sobre las principales operaciones que se realizan, para lograr un menor tiempo de adaptación.

Rendimiento.

- La aplicación debe estar concebida para el consumo mínimo de recursos.
- El sistema debe ser capaz de formular la respuesta lo más rápido posible.

Soporte.

Para el servidor de aplicaciones:

- Se requiere que este instalado un interprete de ficheros PHP y con las últimas actualizaciones del lenguaje.

Para el servidor de base de datos:

- Se requiere que esté instalado un gestor de bases de datos que soporte grandes volúmenes de datos y velocidad de procesamiento.

Para el cliente:

- Se requiere que esté instalado un navegador que interprete Javascript y versiones HTML 3.0 o superior.

Portabilidad

- El sistema debe ser compatible con los SO más usados como UNIX (Linux), Windows (Versiones 2000 y XP), siendo además accesible principalmente en Intranet con el navegador Mozilla.

Hardware

Para las PC del cliente:

- Se requiere tengan tarjeta de red.
- Se requiere tengan al menos 32 MB de memoria RAM.
- No se requiere disco duro.
- Se requiere procesador de 333 MHz como mínimo.

Para los servidores:

- Se requiere tarjeta de red.
- Se requiere tenga al menos 128 MB de RAM.
- Se requiere al menos 1 GB de disco duro.

- Procesador 333 MHz como mínimo.

Software

- Se utilizará un servidor con el SO instalado Windows 2000 o superior, o con SO UNIX (Linux) preferiblemente.
- Se necesita tecnología Apache 2.0 o superior para el servidor.
- En las PC de los clientes se requiere de un navegador (Mozilla Firefox 1.5 o superior).

Seguridad

- El sistema debe comunicarse usando un protocolo seguro (HTTPs).
- Brindar servicio de autenticación.
- Garantizar que la información sea vista solo por quien tiene derecho a verla.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de acceso del usuario que este activo.
- Mantener la integridad de la información, es decir que no se pierda durante su transporte o almacenamiento.

Disponibilidad

- El sistema deberá estar disponible las 24 horas del día para todos los usuarios.

Confiabledad

- La información manejada por el sistema debe estar protegida de accesos no autorizados y divulgación.

2.6 Modelo de C/U del sistema.

Una de las situaciones más difíciles a la hora de construir un sistema es precisamente saber qué construir. Ninguna otra parte del trabajo conceptual es tan compleja como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con las personas, maquinas y otros sistemas. Este establecimiento de los requisitos es el punto de partida para la identificación de los principales casos de uso y actores del sistema en cuestión. Los casos

de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre los clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema. Por lo tanto podemos plasmar todos los requerimientos en un modelo de casos de uso del sistema y a su vez identificar los principales actores que interactúan con el mismo. Un actor no es parte del sistema, sino un rol que se juega dentro del sistema, que puede intercambiar información o puede ser un recipiente pasivo de información y representa a un ser humano, a un software o a una maquina que interactúa con el sistema.

Utilizando las facilidades que nos brinda el UML, vamos a capturar los requisitos funcionales del sistema y a representarlos mediante un diagrama de casos de uso. Para ello tenemos que definir cuales serían los actores que van a interactuar con el sistema y los casos de uso que van a representar las funcionalidades.

2.6.1 Actores del sistema para automatización.

ACTORES	JUSTIFICACIÓN
Cliente	Representa el rol de todos los clientes que pudieran entrar en la aplicación en busca de información sobre un servicio.
Especialista Principal	Representa el rol de la persona que esta encargada de todos los equipos de trabajo.
Jefe de Equipo	Representa el rol de la persona encargada de dirigir y asistir el equipo de trabajo.
Director UEB	Representa el rol de la persona encargada de dirigir y controlar los servicios que se prestan.

Capítulo 2 Características del Sistema.

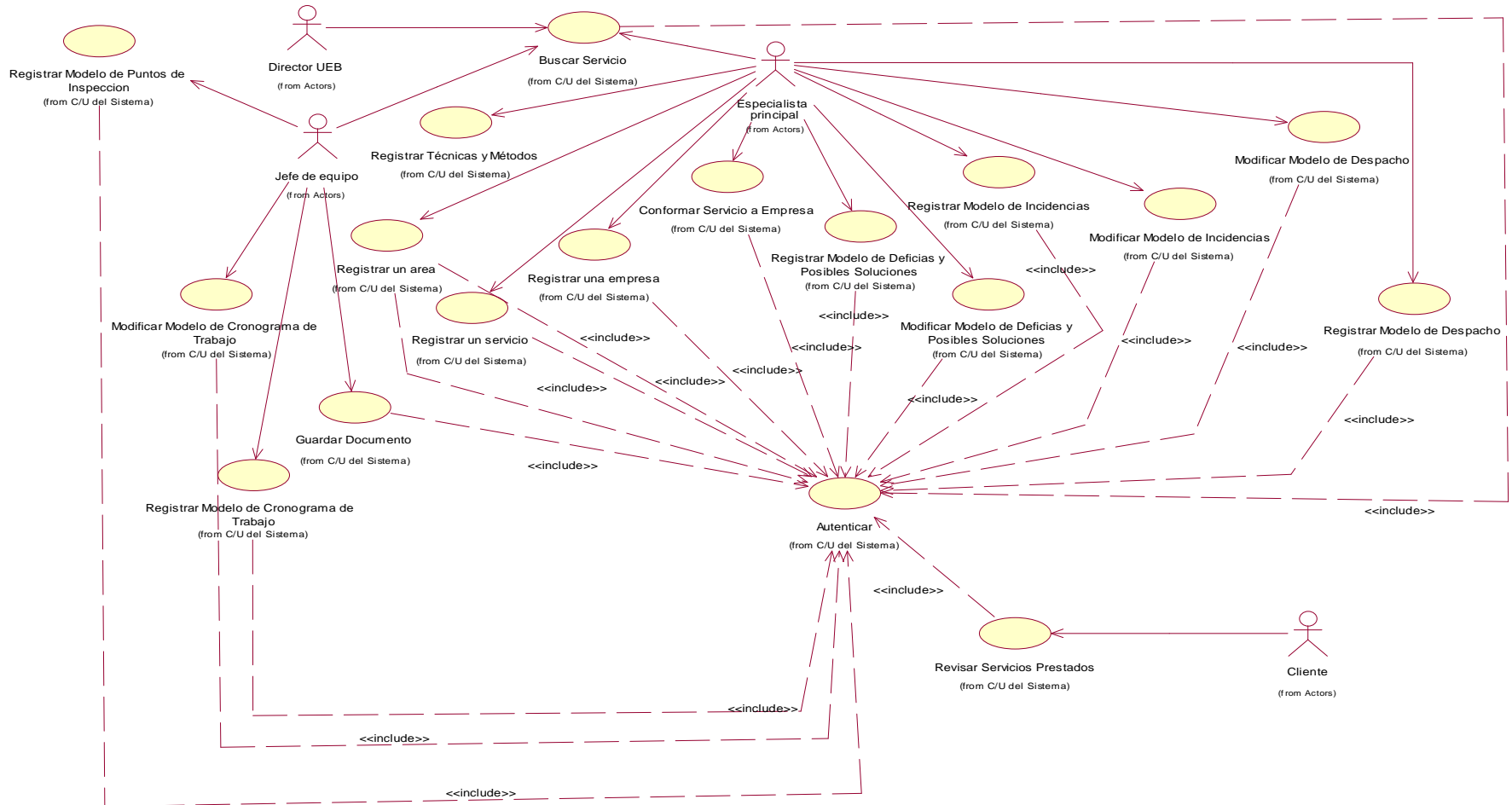


Figura 2.3: Diagrama de casos de uso del Sistema.

2.6.2 Descripción de los casos de uso del Sistema.

Caso de uso del sistema:	Registrar Modelo de Cronograma de Trabajo.
Actores del sistema:	Jefe de Equipo (Inicia)
Propósito:	Permitir que el cronograma este almacenado y a disposición de los miembros del equipo de trabajo.
Resumen:	El caso de uso se inicia cuando el Jefe de Equipo abre la aplicación se dirige a la opción de Registrar Modelo de Cronograma de Trabajo. El caso de uso termina cuando oprime el botón de registrar y cierra la aplicación.
Referencia:	R6.
Precondiciones:	
Postcondiciones:	El cronograma debe estar realizado.
Prioridad:	

Caso de uso del sistema:	Registrar Modelo de deficiencias y Posibles Soluciones.
Actores del sistema:	Especialista Principal. (Inicia)
Propósito:	Permitir al cliente una vez terminado el producto poder definir sus deficiencias y ver las posibles soluciones de forma concreta.
Resumen:	El caso de uso se inicia cuando el Especialista Principal se dirige a la opción Registrar Modelo de deficiencias y Posibles Soluciones. El caso de uso finaliza cuando oprime el botón registrar y cierra la aplicación.
Referencia:	R7.
Precondiciones:	El Especialista Principal debe haberse logueado.
Postcondiciones:	El modelo fue registrado.
Prioridad:	

Caso de uso del sistema:	Registrar Modelo de Incidencias.
Actores del sistema:	Especialista Principal. (Inicia)
Propósito:	Permitir al Especialista Principal tener un control de las incidencias en un servicio determinado.
Resumen:	El caso de uso se inicia cuando el Especialista Principal se dirige a la opción Registrar Modelo de Incidencias. El caso de uso finaliza cuando oprime el botón registrar y cierra la aplicación.
Referencia:	R8.
Precondiciones:	El Especialista Principal debe haberse logueado.

Postcondiciones:	El modelo fue registrado.
Prioridad:	
Caso de uso del sistema:	Registrar Modelo de Puntos de Inspección.
Actores del sistema:	Jefe de Equipo. (Inicia)
Propósito:	Permitir al Jefe de Equipo poder realizar un control del trabajo..
Resumen:	El caso de uso se inicia cuando el efe de Equipo se dirige a la opción Registrar Modelo de Puntos de Inspección. El caso de uso finaliza cuando oprime el botón registrar y cierra la aplicación.
Referencia:	R10.
Precondiciones:	El Jefe de Equipo debe haberse logueado.
Postcondiciones:	El modelo fue registrado.
Prioridad:	

Caso de uso del sistema:	Revisar Servicios Prestados.
Actores del sistema:	Usuario. (Inicia)
Propósito:	Permitir al Usuario poder buscar servicios anteriores para su estudio.
Resumen:	El caso de uso se inicia cuando el Usuario se dirige a la opción Buscar Servicios Prestados. El caso de uso finaliza cuando cierra la aplicación.
Referencia:	R5.
Precondiciones:	El Usuario debe haberse logueado.
Postcondiciones:	
Prioridad:	

Caso de uso del sistema:	Buscar Servicio.
Actores del sistema:	Usuario, Jefe de equipo, Especialista Principal. (Inicia)
Propósito:	Permitir al Usuario poder buscar servicios anteriores para su estudio o poder hacer una búsqueda de documentación, en caso de un trabajador poder buscar un servicio para modificarlo.
Resumen:	El caso de uso se inicia cuando el Usuario abre la aplicación y se dirige a la opción Buscar Servicios Prestados o a otra información. El caso de uso finaliza cuando cierra la aplicación.
Referencia:	R5.
Precondiciones:	El Usuario debe haberse logueado.
Postcondiciones:	
Prioridad:	

Caso de uso del sistema:	Conformar Servicio a Empresa.
Actores del sistema:	Especialista Principal. (Inicia)
Propósito:	Permitir al Especialista Principal poder conformar un servicio al que se le agregara información posterior.
Resumen:	El caso de uso se inicia cuando el Usuario abre la aplicación y se dirige a la opción Servicio a empresa. El caso de uso finaliza cuando se haya registrado el nuevo servicio.
Referencia:	R11.
Precondiciones:	El Usuario debe haberse logueado.
Postcondiciones:	
Prioridad:	

2.7 Conclusiones.

En este capítulo se comenzó a desarrollar la propuesta de solución, obteniéndose a partir del análisis de los procesos del negocio, un listado con las principales funcionalidades que debe tener el sistema y los requisitos adicionales. Se presentaron los Diagramas de Casos de Uso del Sistema y finalmente se describieron las acciones de los actores del mismo con los casos de uso con los que interactúan.

Capítulo 3

Análisis y Diseño

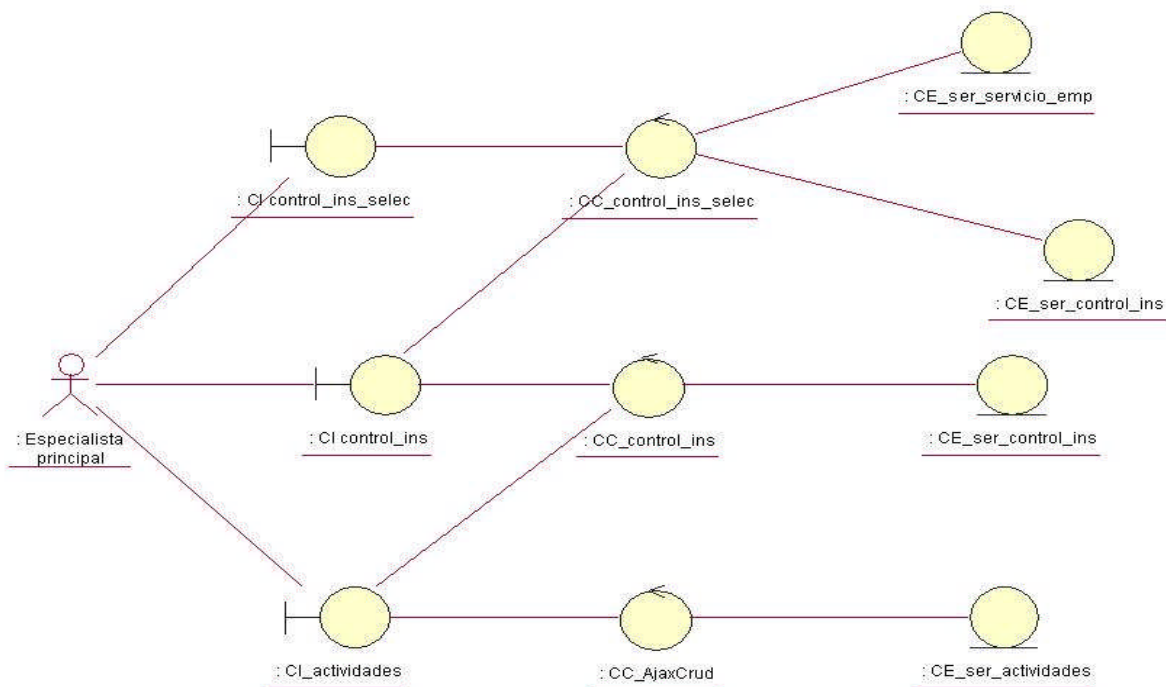
3.1 Introducción.

Este capítulo tiene el objetivo de plantear la concepción general del análisis y diseño del sistema propuesto y cómo se desarrollará. Para ello se modelan los artefactos que ayudan a manejar las complicaciones que implican la construcción de aplicaciones Web, donde los componentes de la aplicación se tratan como clases y utilizando las extensiones del UML, podemos presentarlas a través de diagramas de clases Web. Además se presenta el modelo de datos que es la base para construir finalmente la base de datos que soportará el sistema.

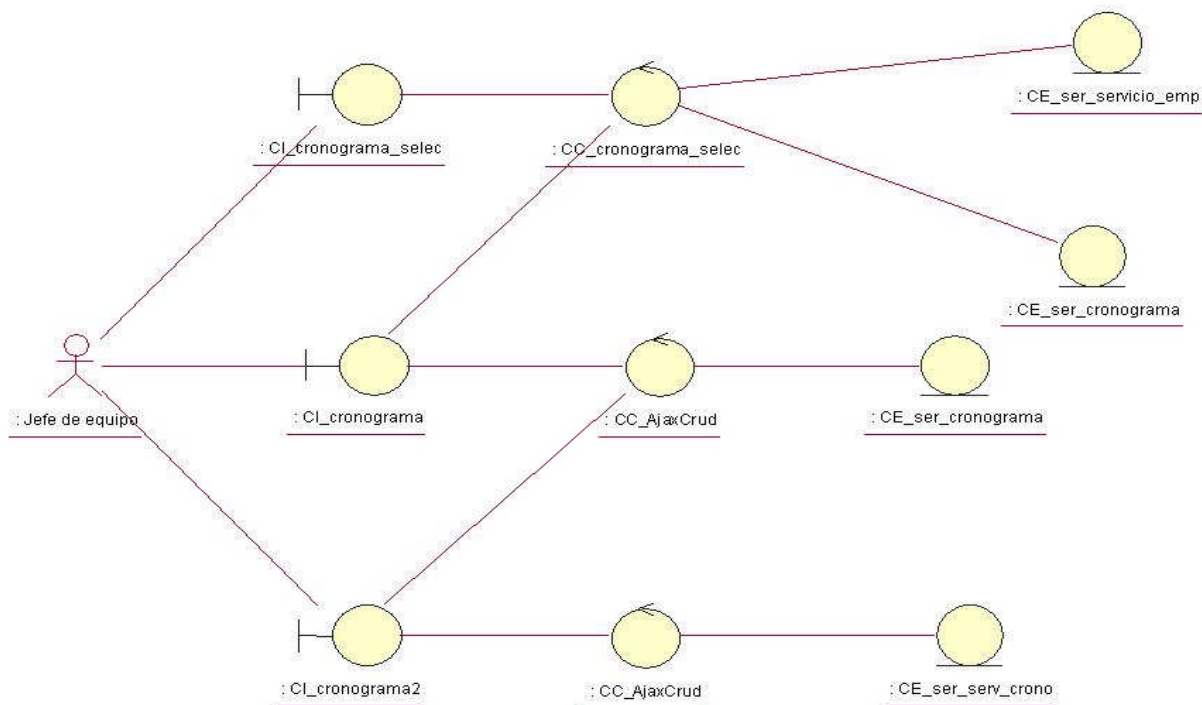
3.2 Análisis

3.2.1 Diagramas de clases del análisis.

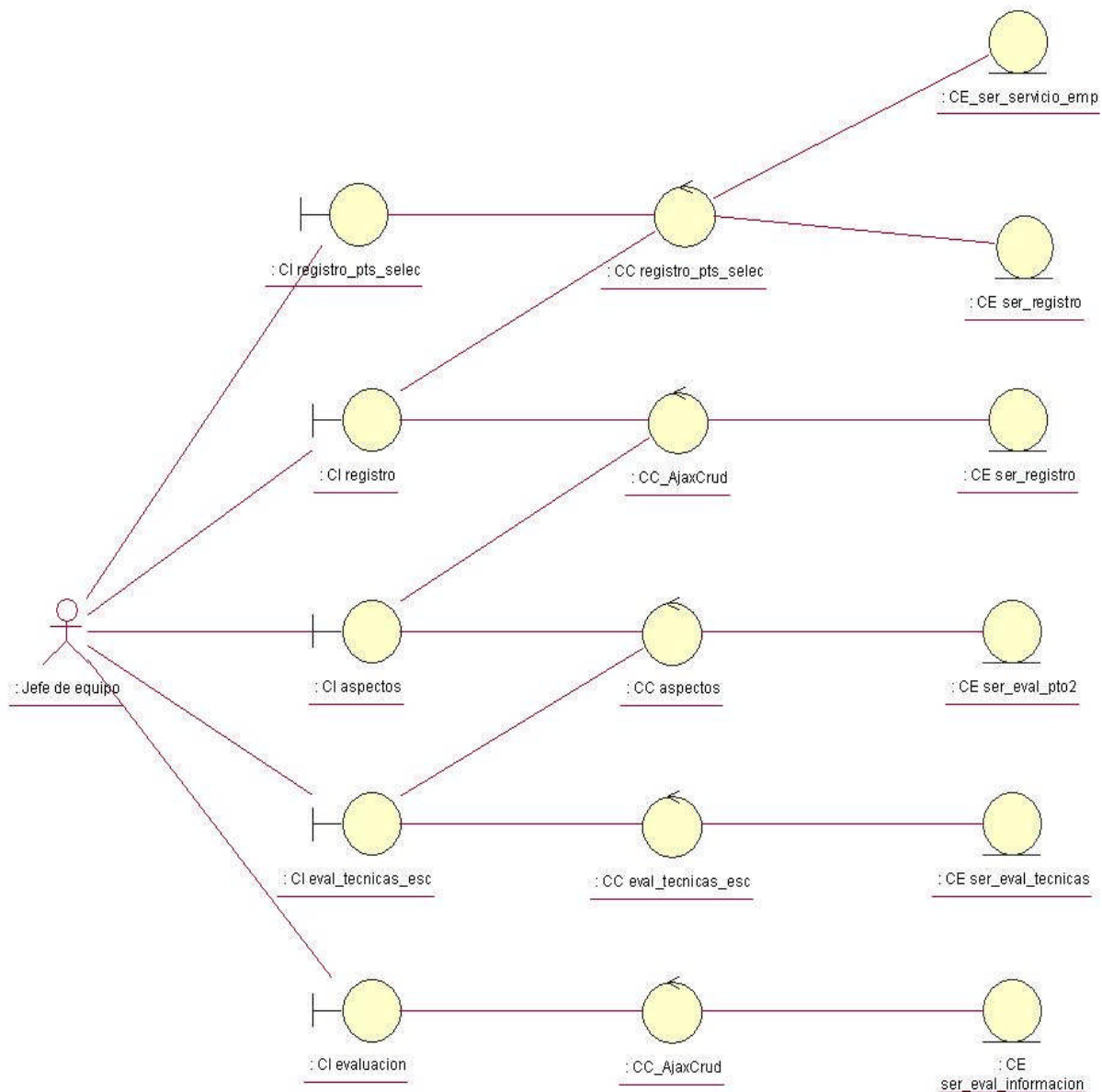
Un diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas.



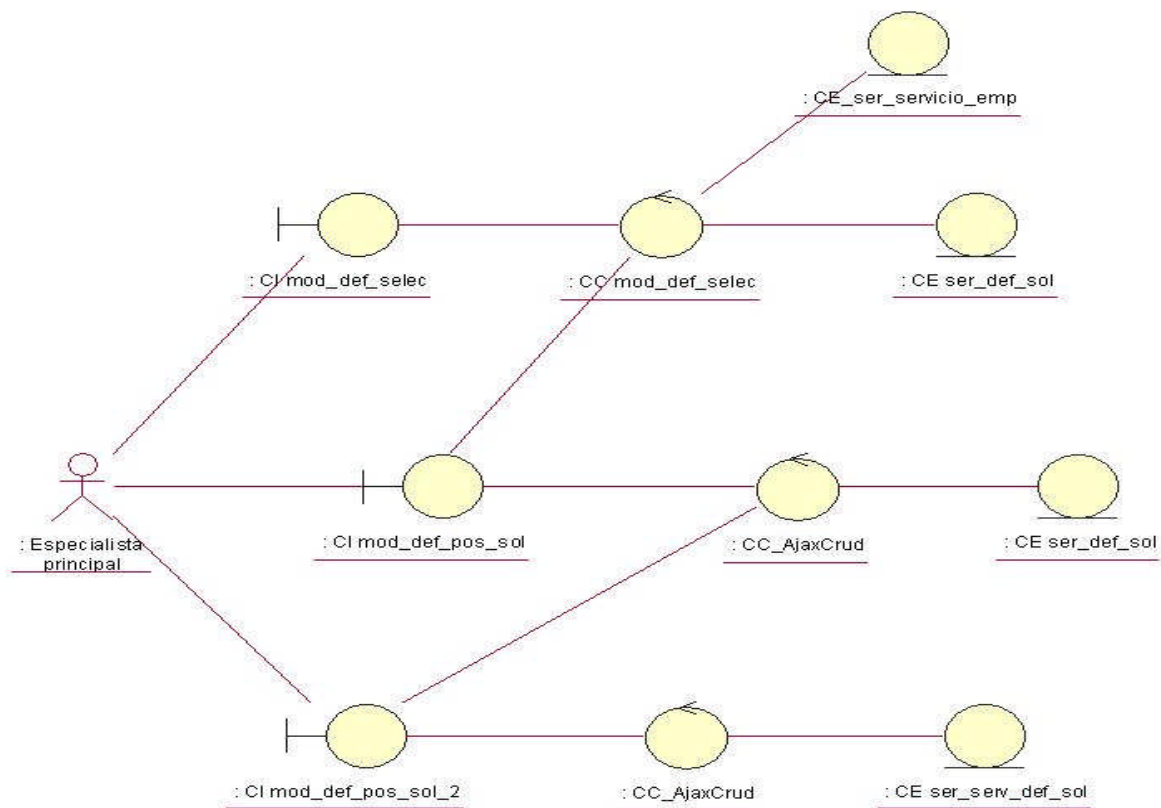
CU Registrar Control de Incidencias en los Servicio.



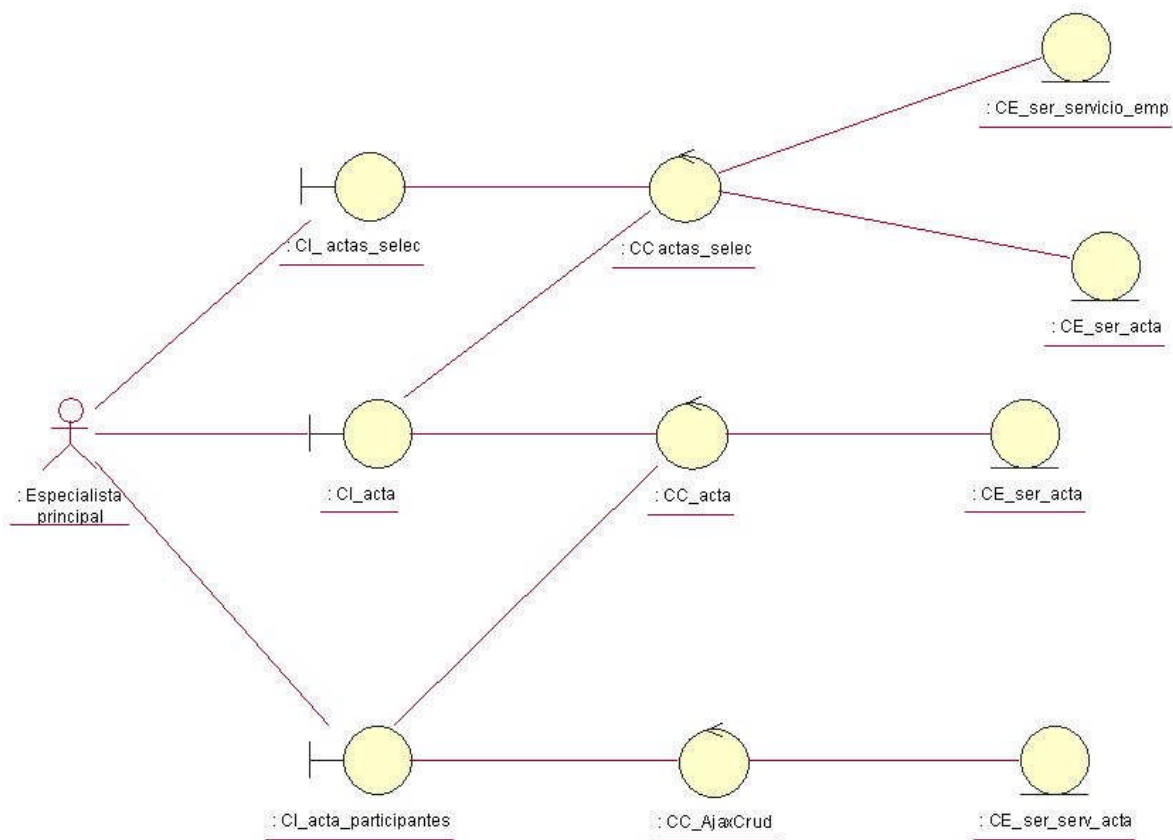
CU Registrar Cronograma de trabajo.



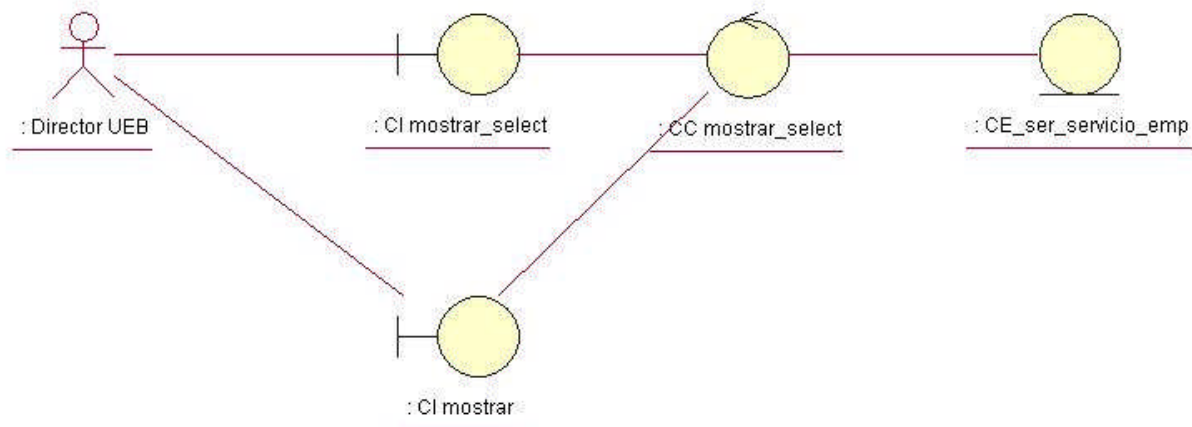
CU Registrar Modelo Registro de Puntos de Inspección.



CU Registrar Modelo de Deficiencias y Posibles Soluciones.



CU Registrar Acta del Despacho.



CU Mostrar Servicio.

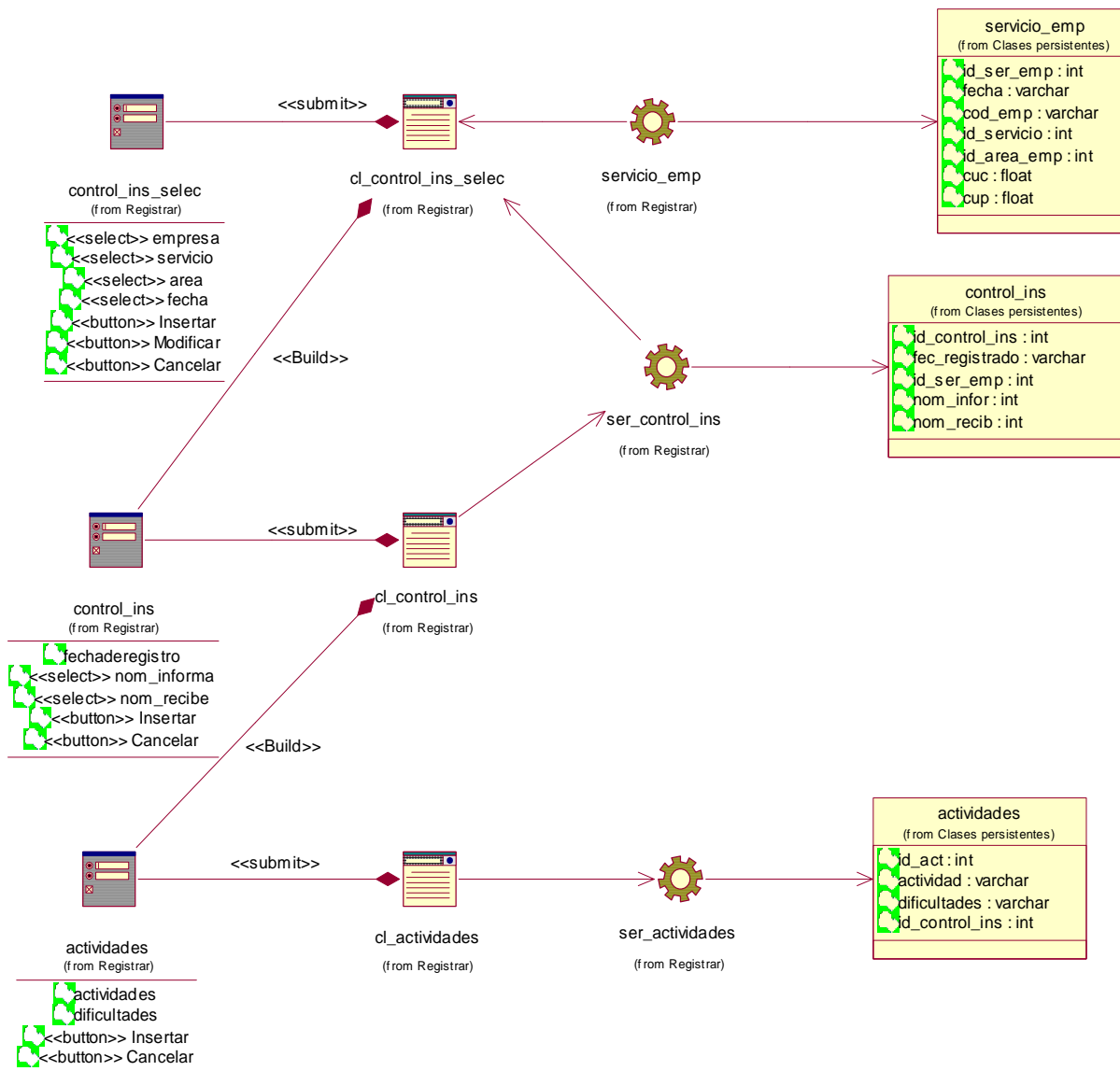


CU Conformar Servicio a Empresa.

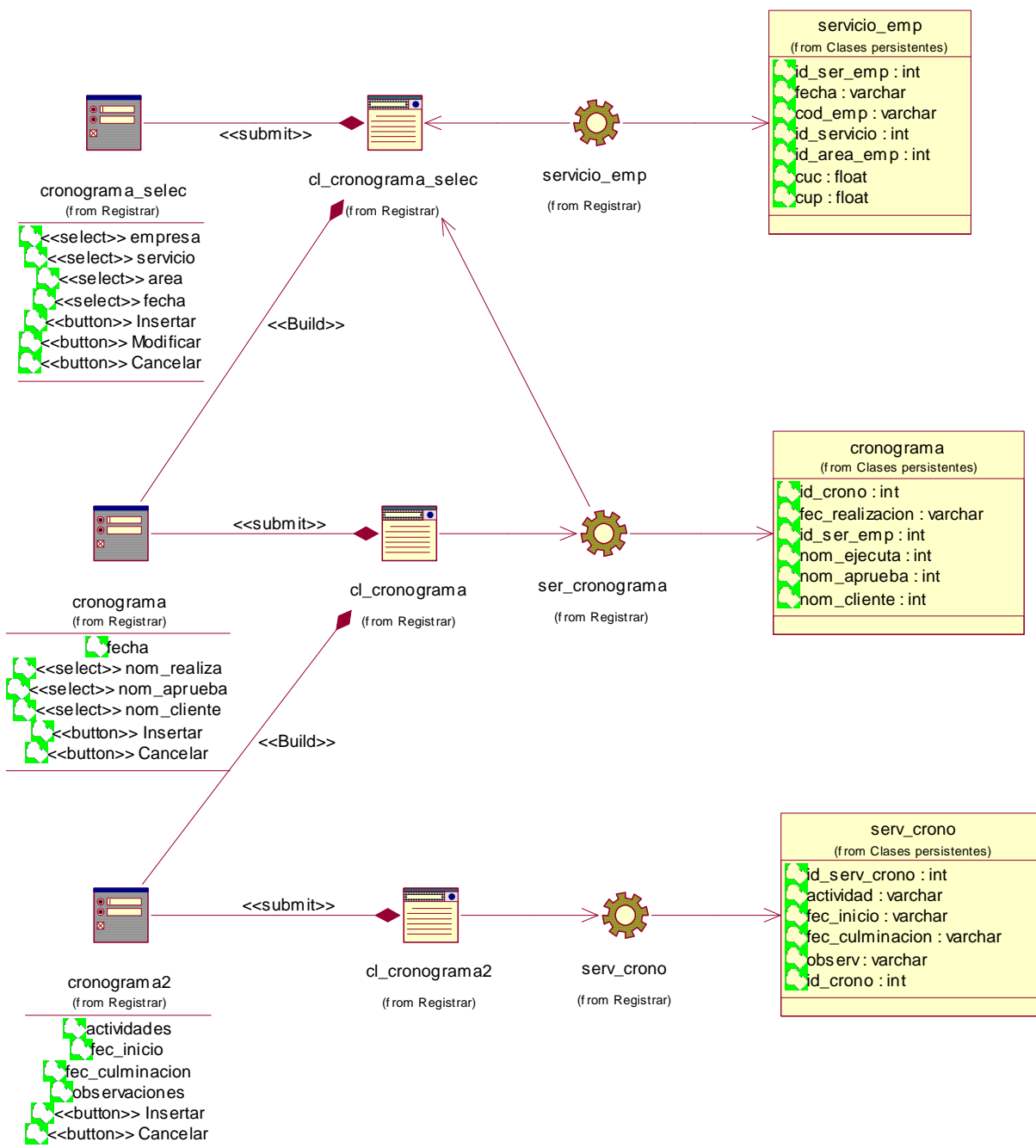
3.3 Diseño.

3.3.1 Diagramas de clases del diseño.

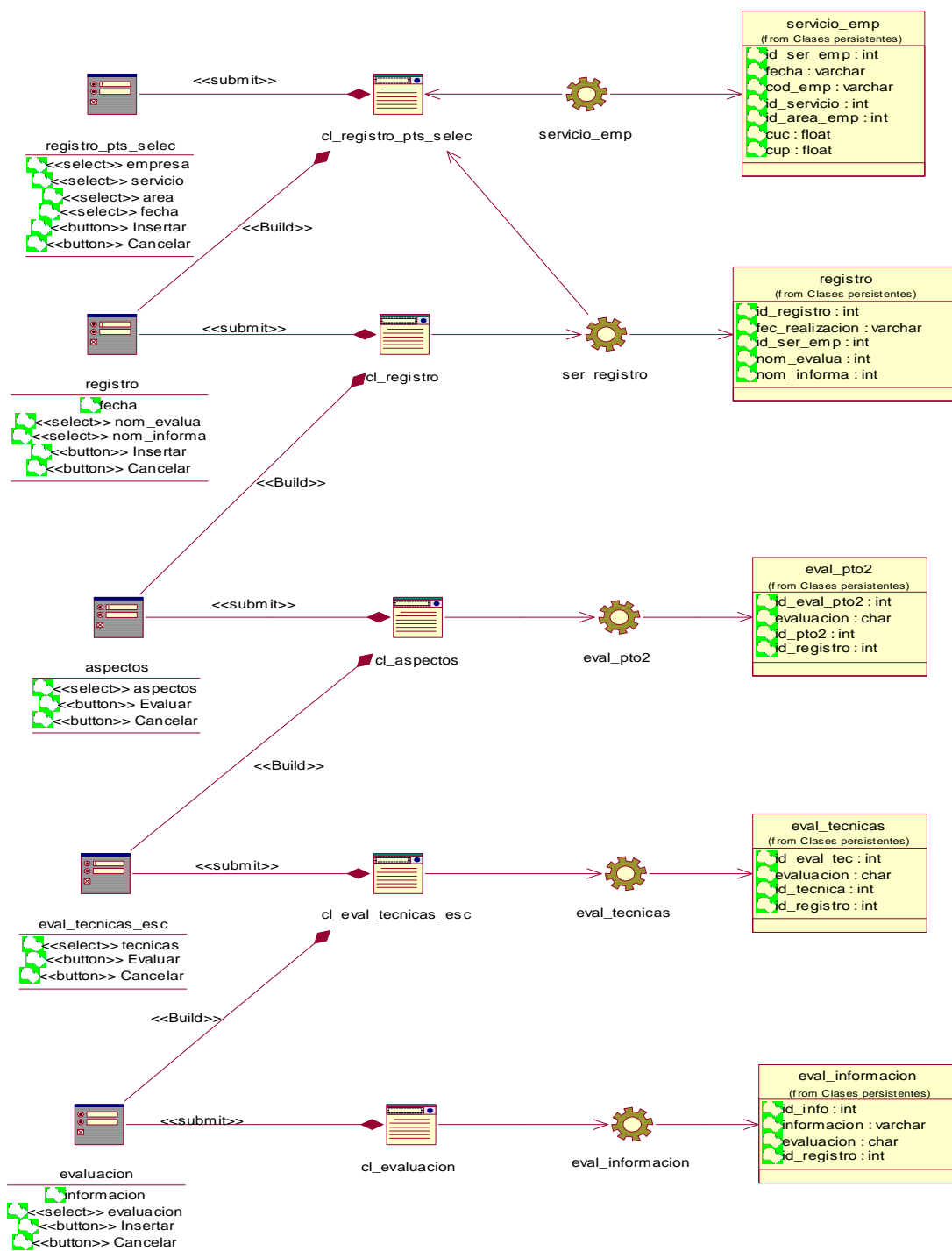
El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Esto contribuye a una arquitectura estable y sólida, y crear un plano del modelo de implementación. Durante la fase de construcción, cuando la arquitectura es estable y los requisitos están bien entendidos, el centro de atención se desplaza a la implementación. El modelo de diseño está muy cercano al de implementación, lo que es natural para guardar y mantener el modelo de diseño a través del ciclo de vida completo del software.



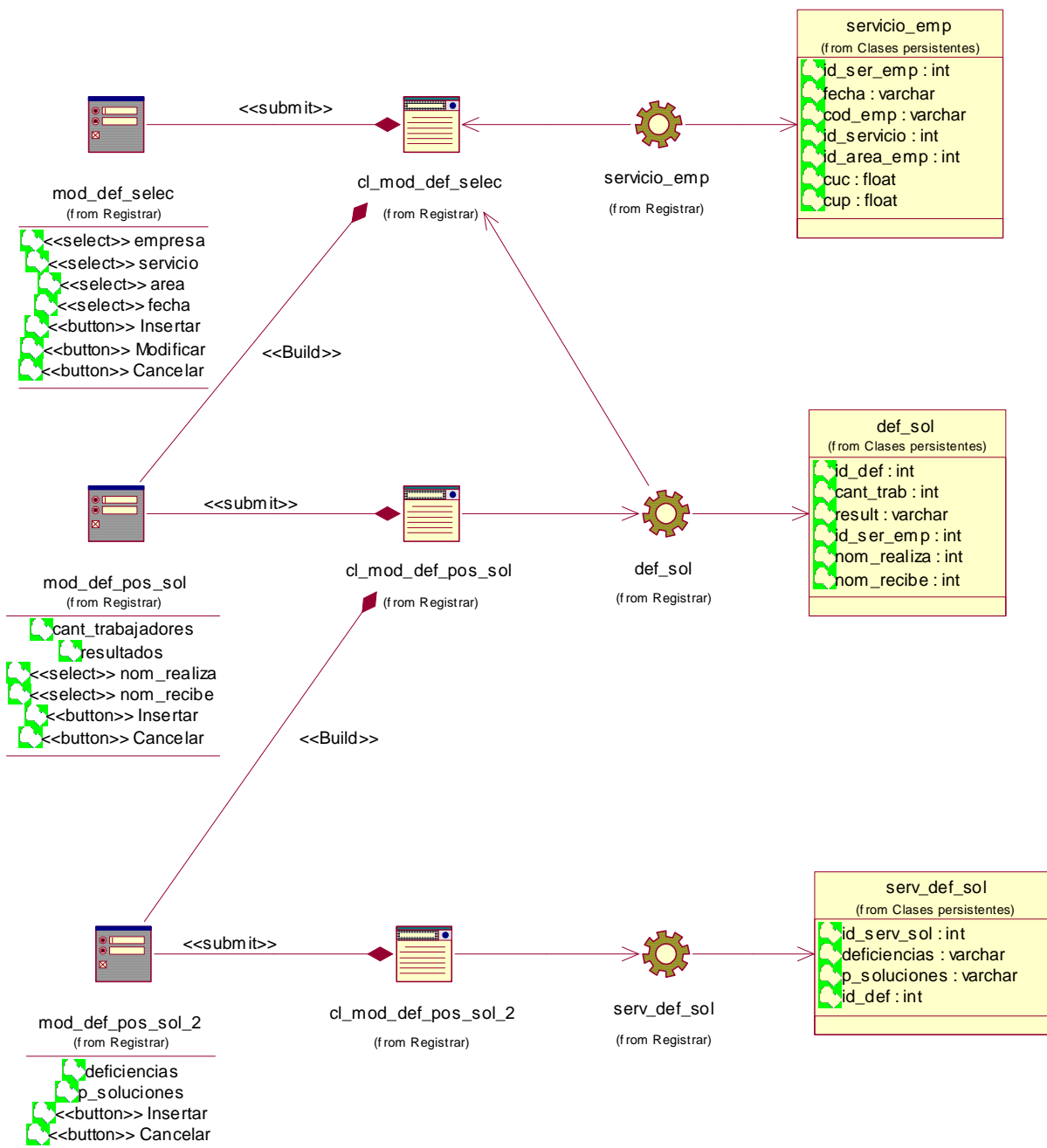
CU Registrar Control de incidencias.



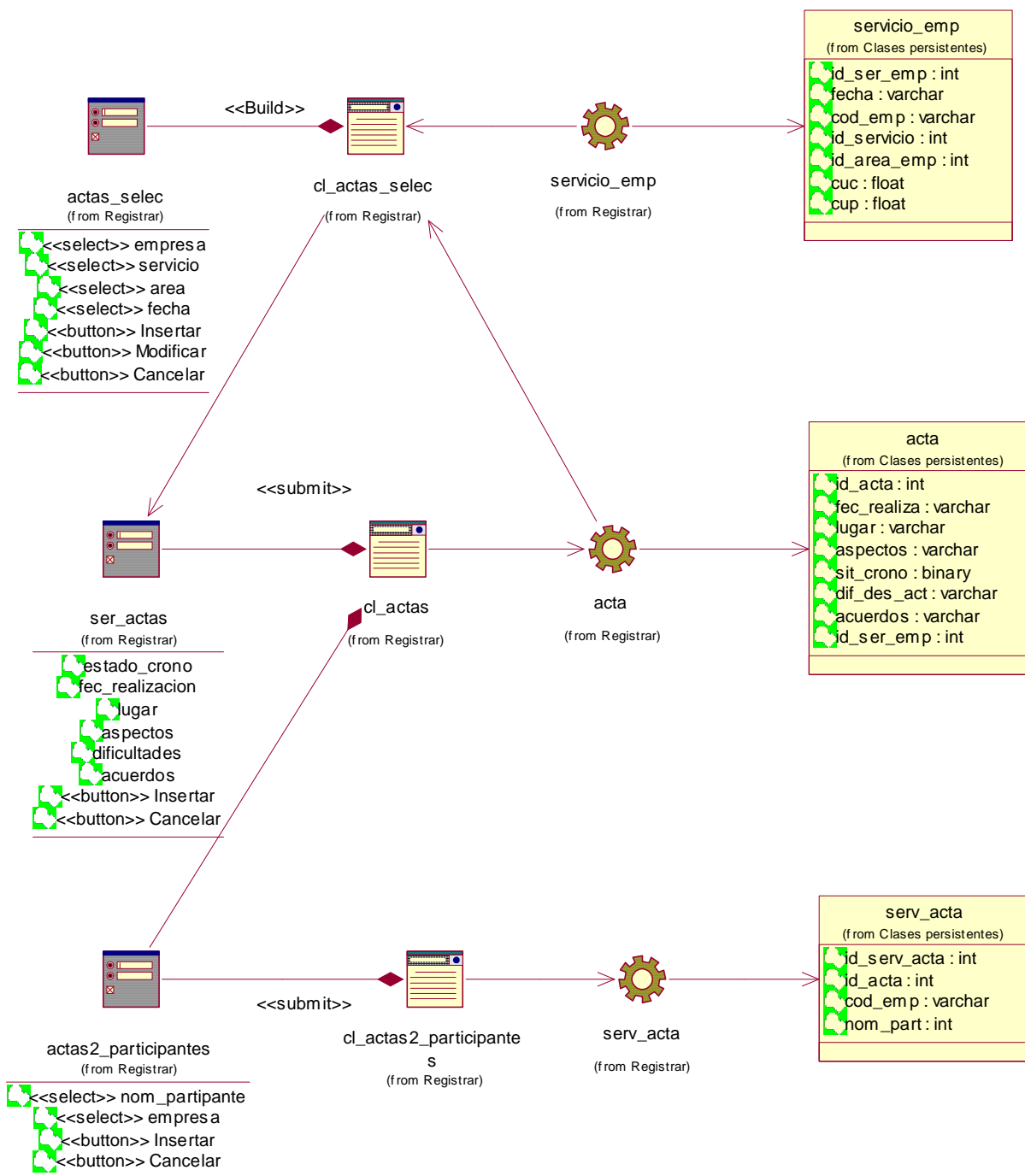
CU Registrar Cronograma de Trabajo.



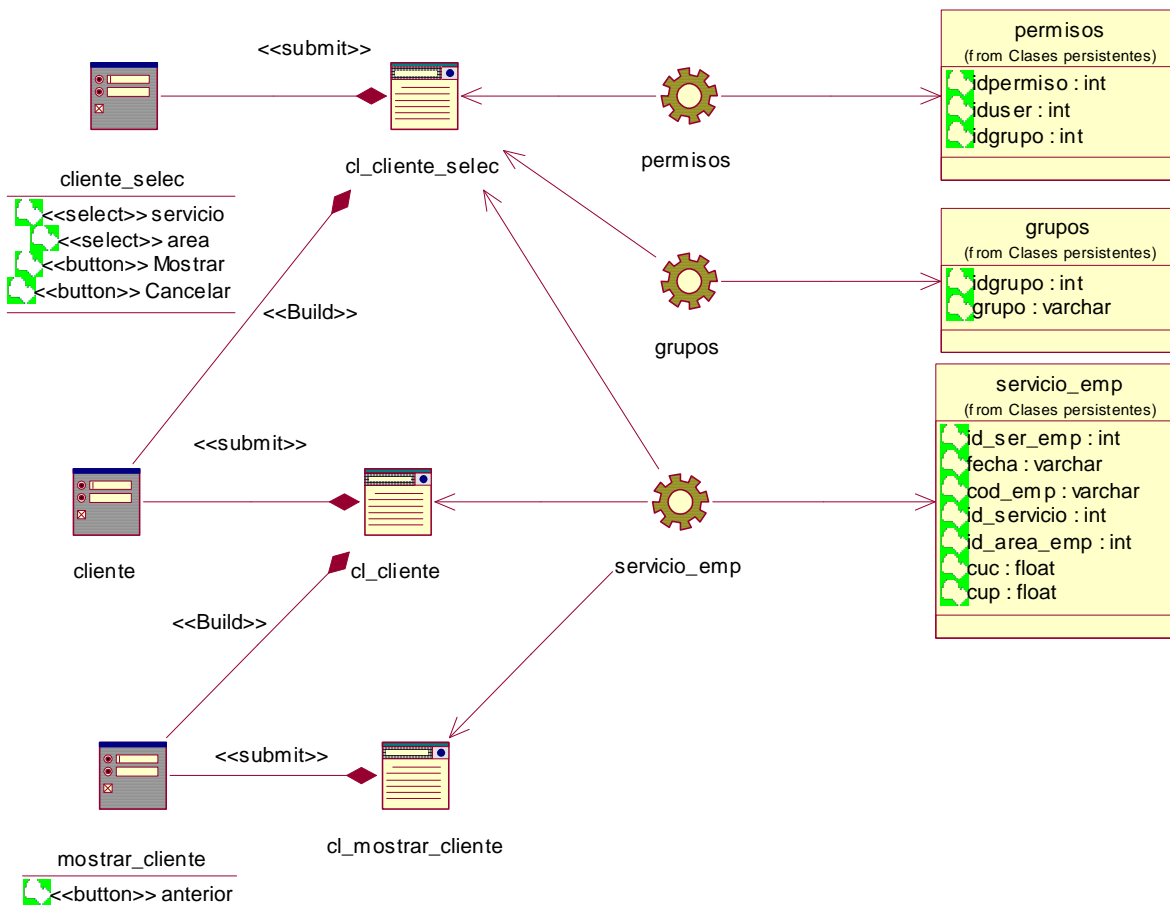
CU Registrar Modelo de Registro de Puntos de Inspección.



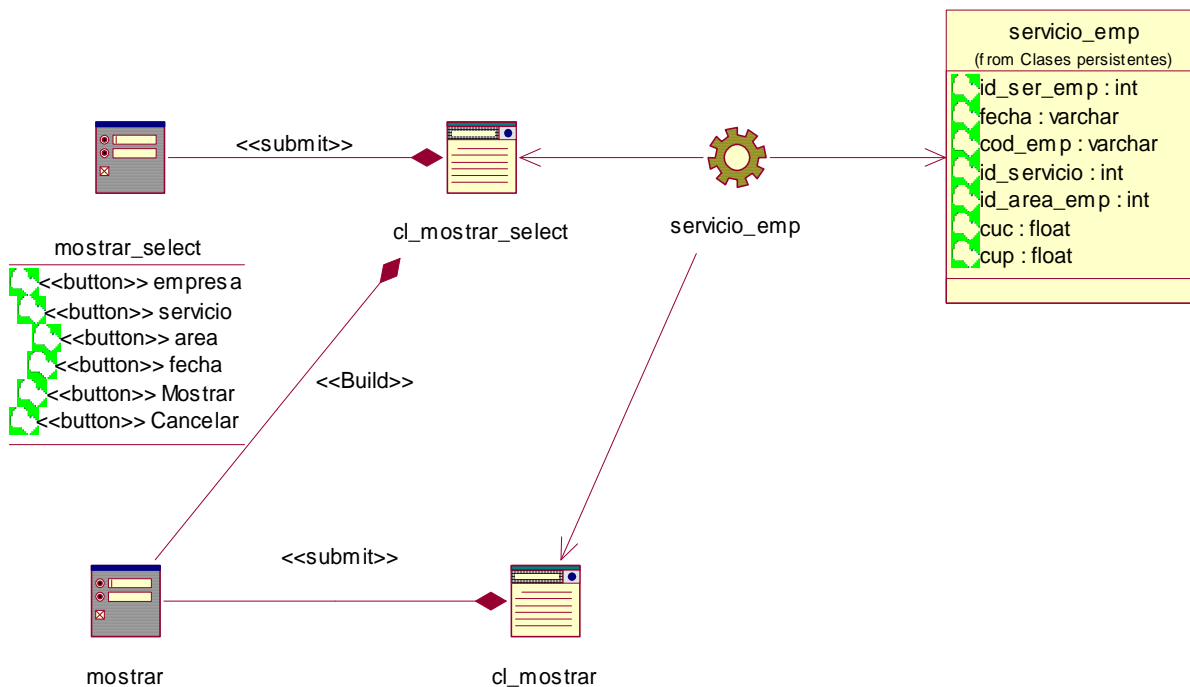
CU Registrar Modelo de Deficiencias y Posibles Soluciones.



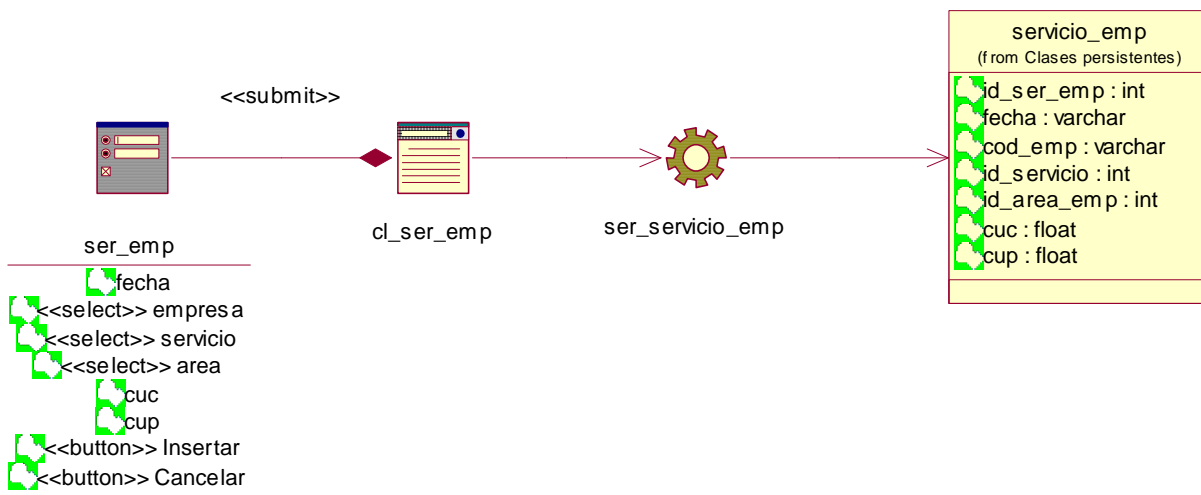
CU Registrar Acta del Despacho.



CU Revisar Servicios Prestados.

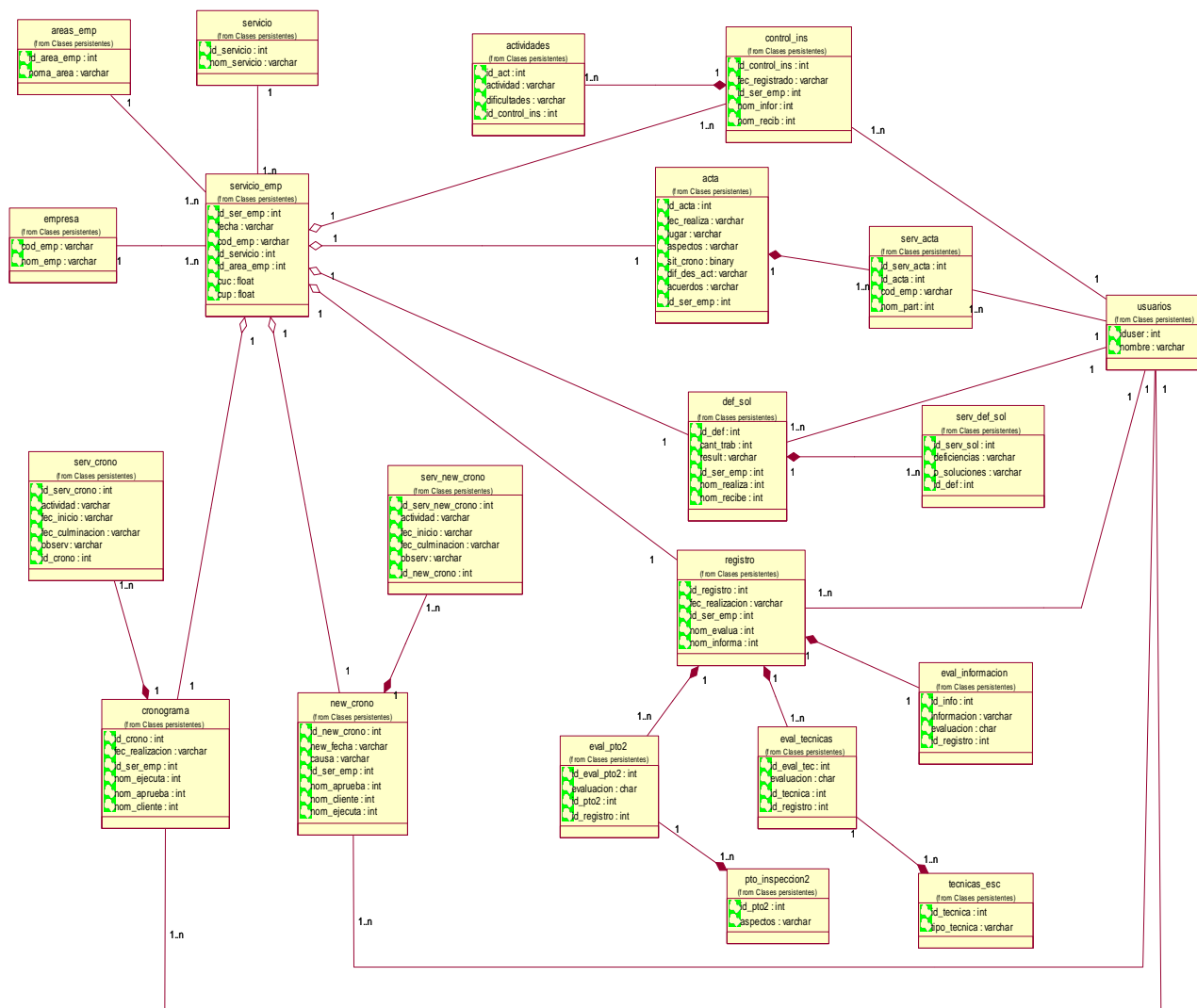


CU Mostrar Servicio.

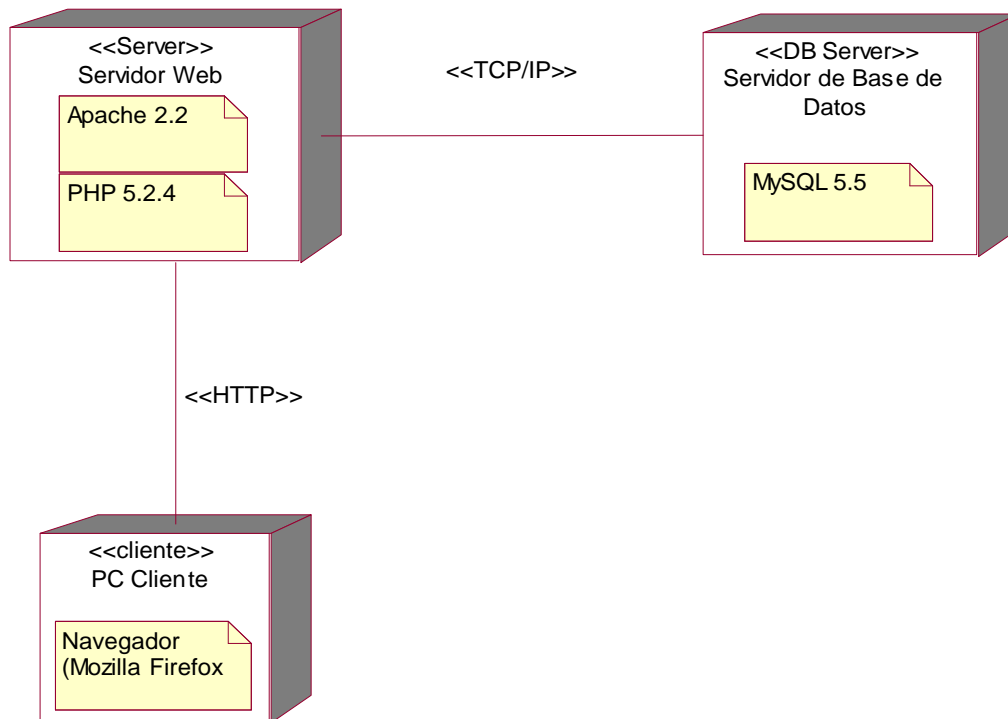


CU Conformar Servicio a Empresa.

3.4 Diagrama de clases persistentes.



3.5 Diagrama de despliegue.



3.6 Conclusiones.

En este capítulo se ha realizado una modelación del sistema para la implementación del sistema. Se utilizaron diagramas de clases Web para explicar la lógica del negocio del sistema, y se diseñaron las clases persistentes. En este momento, ya se tiene confeccionada completamente la propuesta que trae este trabajo.

4.1 Introducción.

El desarrollo de cualquier producto es muy probable que este plagado de escasez de recursos y de fechas de entrega difíciles, por lo que se hace necesario y prudente evaluar la viabilidad de un proyecto cuanto antes. Se pueden evitar meses o años de esfuerzo, miles de pesos si se reconoce un sistema mal concebido en la pronta fase de definición. Los productos informáticos no están exentos de estos problemas, por lo que estimar el esfuerzo humano, el tiempo de desarrollo que se requiere para la ejecución del mismo y también su costo son de vital importancia. Para realizar la estimación existen dos métodos:

1. **Análisis de Puntos de Casos de Uso:** Existe una posibilidad de predecir el tamaño de un sistema a partir de las características de sus requisitos, expresados en los casos de uso. (para proyectos no mayores de 20 CU).
2. **Análisis de Puntos de Función y COCOMO II:** El SEI (del inglés, Software Engineering Institute) propone desde hace algunos años un método para la estimación del esfuerzo llamado COCOMO II. Éste método está basado en ecuaciones matemáticas que permiten calcular el esfuerzo a partir de ciertas métricas de tamaño estimado, como el Análisis de Puntos de Función y las líneas de código fuente (en inglés SLOC, Source Line Of Code).

4.2 Estudio de Factibilidad.

En el presente trabajo se utilizará el método de Puntos de Casos de Uso, puesto que el sistema cuenta con 18 casos de uso, siendo este un método sencillo que cuenta con un software (Estimac) para facilitar su uso.

Factor de Peso de los Actores sin Ajustar.

Tipo	Peso	Valor	Peso * Valor
Simple	1	0	0
Medio	2	0	0
Complejo	3	4	12
			UAW: 12

Factor de Peso de los Casos de Uso sin Ajustar.

Tipo	Peso	Valor	Peso * Valor
Simple	5	5	25
Medio	10	2	20
Complejo	15	10	150
			UUCW: 195

Puntos de Casos de Uso sin Ajustar

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

$$\text{UUCP} = 12 + 195$$

$$\text{UUCP} = 207$$

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

Factor de Complejidad Técnica

Factor	Peso	Valor	Peso * Valor
T1	2	0	0
T2	1	1	1
T3	1	3	3
T4	1	1	1
T5	1	0	0
T6	0.5	1	0.5
T7	0.5	3	1.5
T8	2	1	2
T9	1	3	3
T10	1	0	0
T11	1	3	3
T12	1	5	5
T13	1	1	1
		Total:	21
			TCF: 0.81

$$TCF = 0.6 + 0.01 \times \Sigma (\text{Peso} \times \text{Valor asignado})$$

$$TCF = 0.6 + 0.01 \times 21$$

$$TCF = 0.81$$

Factor de Ambiente

Factor	Peso	Valor	Peso * Valor
E1	1.5	4	6
E2	0.5	4	2
E3	1	5	5
E4	0.5	4	2
E5	1	5	5
E6	2	3	6
E7	-1	0	0
E8	-1	2	2
		Total:	28
			EF: 0.68

$$EF = 1.4 - 0.03 \times \Sigma (\text{Peso} \times \text{Valor asignado})$$

$$EF = 1.4 - 0.03 \times 28$$

$$EF = 0.68$$

Puntos de casos de uso Ajustados.

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 207 \times 0.81 \times 0.68$$

$$UCP = 114.0156$$

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

Factor de conversión (CF): 20 horas-hombre/Punto de caso de uso.

Estimación del Esfuerzo Total del Proyecto

$$E = UCP \times CF$$

Actividad	% Esfuerzo	Valor Esfuerzo (horas/hombre)
Análisis	10	570.078
Diseño	20	1140.156
Implementación	40	2280.312
Prueba	15	855.117
Sobrecarga	15	855.117
Total	100%	5700.78

Estimacion de Proyectos Basada en el Análisis de Puntos de Casos de Uso

P 1 : Puntos de CU sin Ajustar Paso 2 : Puntos de CU Ajustados Paso 3 : Calculo del esfuerzo de Implementacion - Estimacion Final

Factor de Conversion (CF) - Esfuerzo de Implementacion (E)

Valor: horas-hombre Esfuerzo de Implementacion: horas-hombre

Estimacion del Esfuerzo Total del Proyecto

Actividad	% Esfuerzo	Valor Esfuerzo
Análisis	10	570,078
Diseño	20	1140,156
Implementación	40	2280,312
Prueba	15	855,117
Sobrecarga	15	855,117
Total	100%	5700,78

Formato de los resultado en números reales

 Editar % del Esfuerzo Total

Si trabajan personas, horas como promedio en un/una

 el proyecto se puede terminar en aproximadamente Mes

Los valores obtenidos indican que el proyecto necesitaría 4 hombres para su realización en aproximadamente 9 meses.

Costo por Mes (CM) y Costo General (CG):

CM = Salario Total

CM = Salario * cantidad de obreros

CM = 225 * 4

CM = 900 Pesos por mes.

CG = CM * TDEV

CG = 900 * 8.90

CG = 8010 Pesos

El desarrollo del sistema tendrá un costo estimado de \$ 8010.

4.3 Análisis de Sostenibilidad.

Al ser implantado un sistema informático este repercute de forma positiva y negativa sobre los usuarios finales, por lo que es necesario realizar un análisis de su impacto en las dimensiones administrativas, socio-humanista, ambiental y tecnológico, para así contribuir al desarrollo sostenible del producto informático. Se ha afirmado con frecuencia que las nuevas Tecnologías informáticas tendrán un impacto económico tan fuerte como el de la revolución industrial. De hecho, gracias a los avances de las Tecnologías informáticas y de la comunicación, la forma en que los directivos gestionan la información en sus empresas ha cambiado completamente, pasando de los informes impresos al uso de productos informáticos que le brinden la información que necesitan.

El desarrollo de este sistema no supone grandes gastos de recursos, ni tampoco de tiempo; la base de datos que contiene puede ser alojada en los sistemas de Gestión existentes en la empresa, ya que los mismos tienen buenas prestaciones y acceso rápido. La tecnología utilizada para el desarrollo del sistema es totalmente libre, por lo que no hay que incurrir en gastos en el pago de licencias de uso. El sistema es portable por lo que un cambio de plataforma para la implantación del mismo es viable y factible, y no hay que incurrir en muchos cambios.

Aún cuando existe un gasto de electricidad asociado al consumo de las maquinas y el servidor del sistema que debe mantenerse encendido 24 horas al día, el mismo no aumentará el consumo actual pues esta situación ya se presenta.

Con la explotación del sistema que se desarrolla, se obtienen beneficios directos al medio ambiente, ya que se disminuye el volumen de papel y tinta antes utilizado por parte de los técnicos, eliminando así el uso excesivo de estos recursos.

Los beneficios inmediatos son por lo general intangibles:

1. Ahorro de tiempo en la búsqueda de información de un servicio prestado.
2. Los clientes pueden revisar el estado de un servicio en todo momento desde sus empresas.
3. Seguridad al guardar la información.

Con el sistema se entregará un manual de usuario donde los usuarios podrán consultar a detalle el funcionamiento del sistema. Los tiempos de respuestas del sistema son mínimos, como máximo dos segundos y los principales retrasos están dados por el tráfico de la red, por lo que los usuarios finales realizarán de forma rápida su trabajo. Con la implantación de la aplicación no se incide sobre el deterioro del medio ambiente.

Existe una estrecha comunicación entre los desarrolladores de la aplicación y los usuarios finales, logrando que el sistema que se desarrolle, satisfaga las necesidades y expectativas de los clientes finales.

Luego de un análisis realizado del producto informático se puede llegar a las siguientes conclusiones:

- El producto informático desarrollado da respuesta a la necesidad que lo origina.
- Los recursos para desarrollar el sistema son mínimos ya que se realizó con los que se disponían y las tecnologías sobre las que funciona están basadas en software libre.
- La solución que se propone es flexible a cambios.
- El sistema se desarrolló con la filosofía de trabajo con la que se desarrollan todos los sistemas en la empresa, por lo que se facilitará su mantenimiento y futuras actualizaciones.
- Está previsto en el sistema que se incluyan otros módulos del proceso de los servicios, garantizando así su continuidad.

Conclusiones Generales.

El sistema se desarrolló siguiendo la metodología RUP, y se utilizaron representaciones para la modelación de todas las fases del proyecto. El sistema resultante está provisto de un ambiente cómodo, fácil de entender, que cumple los estándares de diseño y utiliza técnicas modernas de programación orientada a objetos, para esto se siguieron los siguientes pasos:

- Se realizó un estudio de los servicios que se prestan en la UEB para tener un mejor conocimiento del negocio, a demás de investigar como se guarda el producto final.
- Se desarrolló un estudio de las diferentes tecnologías y herramientas para la confección del sistema, realizando una selección de acuerdo con las especificaciones del cliente.
- Se elaboró un modelo de negocio y levantamiento de requisitos que permitió entender el problema en cuestión y facilitó el análisis, diseño e implementación del sistema automatizado, con lo que se resolvió las anteriores dificultades:
 1. El cliente ya cuenta con un sistema capaz de guardar grandes volúmenes de información de forma segura, eliminando los inconvenientes de que no se pierdan o deterioren ni que se acumulen de forma excesiva.
 2. Las empresas ya cuentan con la disponibilidad de todos los servicios prestados, con mayor seguridad y confiabilidad.

Por todo lo anterior se concluye que los objetivos propuestos para el presente proyecto han sido cumplidos satisfactoriamente.

Recomendaciones.

A modo general los objetivos trazados al inicio de este trabajo han sido logrados, pero al mismo tiempo, a lo largo del proceso de desarrollo, ha quedado claro que la propuesta es sólo la primera fase de un proyecto que puede ser mucho más ambicioso. Por tanto se hacen las siguientes recomendaciones:

- Aumentar la cantidad de objetivos.
- Mejorar y agregar nuevas funcionalidades a las búsquedas.
- Establecer nuevos métodos de reportes más detallados y específicos.
- Agregar otros módulos de los otros servicios que se prestan en la UEB.

Referencias Bibliograficas:

1. *Getting Started with UML*. January 02, 2007 [cited 2008 19 de Septiembre]; Available from: <http://www.omg.org/UML/>.
2. Sanchez, M.A.M., et al., *Metodologías De Desarrollo De Software*. 2005: Perú S.A.C.
3. *Open Workbench*. 2007 [cited; Available from: http://www.puntoorg.org/open_workbench_gesti_n_de_proyectos_para_windows .
4. *SGBD*. [cited 2008 23 de Septiembre]; Available from: <http://www.eubd.ucm.es/html/personales/enred/mantonia/docauto/tema5/tema5.htm>.
5. *Microsoft SQL Server*. [cited 2008 23 de Septiembre]; Available from: http://es.wikipedia.org/wiki/Microsoft_SQL_Server.
6. *Oracle* [cited 2008 23 de Septiembre]; Available from: <http://es.wikipedia.org/wiki/Oracle>.
7. *MySQL* [cited 2008 19 de Septiembre]; Available from: <http://es.wikipedia.org/wiki/MySQL>.
8. *HTML y XHTML*. [cited 2008 24 de Septiembre]; Available from: http://www.librosweb.es/xhtml/capitulo1/html_y_xhtml.html.
9. *CSS*. [cited 2008 24 de Septiembre]; Available from: http://www.librosweb.es/css/capitulo1/que_es_css.html.
10. *Javascript*. [cited 2008 23 de Septiembre]; Available from: <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
11. *ASP*. [cited 2008 23 de Septiembre]; Available from: <http://www.maestrosdelweb.com/editorial/aspintro/>.
12. *PERL*. [cited 2008 23 de Septiembre]; Available from: <http://www.desarrolloweb.com/articulos/541.php>.
13. *JSP*. [cited 2008 23 de Septiembre]; Available from: <http://www.desarrolloweb.com/articulos/831.php>.
14. *Python*. [cited 2008 23 de Septiembre]; Available from: <http://www.desarrolloweb.com/articulos/1325.php>.
15. *Internet Information Services* [cited 2008 19 de Septiembre]; Available from: <http://es.wikipedia.org/wiki/IIS>.
16. *Zope*. [cited 2008 19 de Septiembre]; Available from: <http://es.wikipedia.org/wiki/Zope>.
17. [cited 2008 24 de Septiembre]; Available from: <http://www.phppizza.com/es/>.
18. Company, T.P. *Zend Studio 5 LAS SOLUCIONES MÁS COMPLETAS PARA EL DESARROLLO DE PHP*. 2007 [cited 2008 19 de Septiembre]; Available from: <http://www.zend.com/en/products/studio/>.
19. Company, T.P. *Performance, Management, Integration, and Enterprise Scalability*. 2007 [cited 2008 19 de Septiembre]; Available from: <http://www.zend.com/en/products/platform/>.
20. *Acerca de Dreamweaver*. [cited 2008 24 de Septiembre]; Available from: http://www.tci.cl/cursos/acerca_de.php.
21. Incorporated, A.S. *Información sobre la integración de las funciones de Studio 8*. 2007 [cited 2008 19 de Septiembre]; Available from: <http://www.adobe.com/es/products/studio/integration/>.
22. *Rational Rose Enterprise*. [cited 2008 24 de Septiembre]; Available from: <http://www-306.ibm.com/software/awdtools/developer/rose/enterprise/index.html>.
23. *XAMPP*. [cited 2008 5 de Octubre]; Available from: <http://es.wikipedia.org/wiki/XAMPP>.

24. *XAMMP*. [cited 2008 5 de Octubre]; Available from: <http://www.genbeta.com/2005/09/27-xampp-servidor-web-correo-base-de-datos-y-mas-extras-en-pocos-minutos>.
25. *Embarcadero ER/Studio* [cited 2008 10 de Octubre]; Available from: http://bureaudeprensa.com/es/view.php?bn=bureaudeprensa_software&key=1153755965.
26. *ER/Studio*. [cited 2008 10 de Octubre]; Available from: <http://www.monografias.com/trabajos14/modelodebase/modelodebase.shtml>.

Anexos.

Anexo # 1 Descripción de casos de uso del sistema.

Caso de uso del sistema:	Registrar un Servicio
Actores del sistema:	Especialista Principal. (Inicia)
Propósito:	Permitir al Especialista Principal poder agregar un servicio al sistema
Resumen:	El caso de uso se inicia cuando el Usuario abre la aplicación y se dirige a la opción Servicios. El caso de uso finaliza cuando se haya registrado el servicio nuevo.
Referencia:	R3.
Precondiciones:	El Usuario debe haberse logueado.
Postcondiciones:	
Prioridad:	

Caso de uso del sistema:	Registrar una empresa
Actores del sistema:	Especialista Principal. (Inicia)
Propósito:	Permitir al Especialista Principal poder agregar una empresa al sistema
Resumen:	El caso de uso se inicia cuando el Usuario abre la aplicación y se dirige a la opción Empresas. El caso de uso finaliza cuando se haya registrado la nueva empresa.
Referencia:	R4.
Precondiciones:	El Usuario debe haberse logueado.
Postcondiciones:	
Prioridad:	

Caso de uso del sistema:	Registrar un área de empresa.
Actores del sistema:	Especialista Principal. (Inicia)
Propósito:	Permitir al Especialista Principal poder agregar un área de una empresa al sistema
Resumen:	El caso de uso se inicia cuando el Usuario abre la aplicación y se dirige a la opción Áreas. El caso de uso finaliza cuando se haya registrado la nueva área.
Referencia:	R4.
Precondiciones:	El Usuario debe haberse logueado.
Postcondiciones:	
Prioridad:	

Caso de uso del sistema:	Registrar Técnicas y Métodos.
Actores del sistema:	Especialista Principal. (Inicia)
Propósito:	Permitir al Especialista Principal poder registrar las técnicas y métodos que se pueden utilizar en los servicios.
Resumen:	El caso de uso se inicia cuando el Usuario abre la aplicación y se dirige a la opción Técnicas y Métodos. El caso de uso finaliza cuando se haya registrado la nueva técnica o método.
Referencia:	R2.
Precondiciones:	El Usuario debe haberse logueado.
Postcondiciones:	
Prioridad:	

Caso de uso del sistema:	Modificar Modelo de Cronograma de Trabajo
Actores del sistema:	Jefe de Equipo (Inicia)
Propósito:	Permitir realizar algún cambio en el cronograma debido a la falta de tiempo o mala elaboración.
Resumen:	El caso de uso se inicia cuando el Jefe de Equipo se dirige a la opción Modificar Modelo de Cronograma de Trabajo. El caso de uso finaliza cuando se oprime el botón de modificar y se cierra la aplicación.
Referencia:	R6.
Precondiciones:	El Jefe de Equipo debe haberse logueado.
Postcondiciones:	El cronograma fue modificado.
Prioridad:	

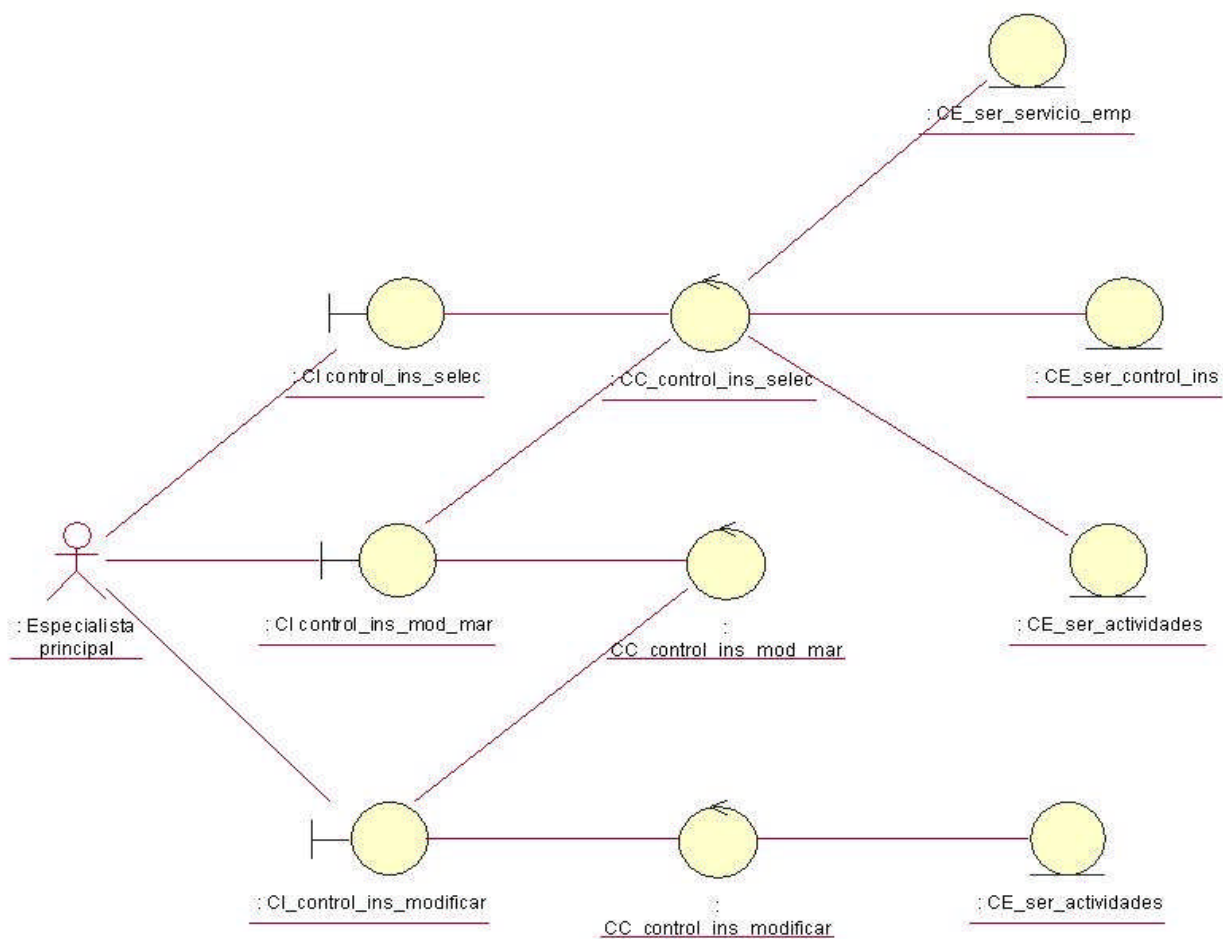
Caso de uso del sistema:	Modificar Modelo de deficiencias y Posibles Soluciones.
Actores del sistema:	Especialista Principal. (Inicia)
Propósito:	Permitir al Especialista Principal realizar algún cambio necesario en el modelo.
Resumen:	El caso de uso se inicia cuando el Especialista Principal se dirige a la opción Modificar Modelo de deficiencias y Posibles Soluciones. El caso de uso finaliza cuando oprime el botón modificar y cierra la aplicación.
Referencia:	R7.
Precondiciones:	El Especialista Principal debe haberse logueado.
Postcondiciones:	El modelo fue modificado.
Prioridad:	

Caso de uso del sistema:	Modificar Modelo de Incidencias.
Actores del sistema:	Especialista Principal. (Inicia)
Propósito:	Permitir al Especialista Principal puede realizar modificaciones en el modelo.
Resumen:	El caso de uso se inicia cuando el Especialista Principal se dirige a la opción Modificar Modelo de Incidencias. El caso de uso finaliza cuando oprime el botón modificar y cierra la aplicación.
Referencia:	R8.
Precondiciones:	El Especialista Principal debe haberse logueado.
Postcondiciones:	El modelo fue modificado.
Prioridad:	

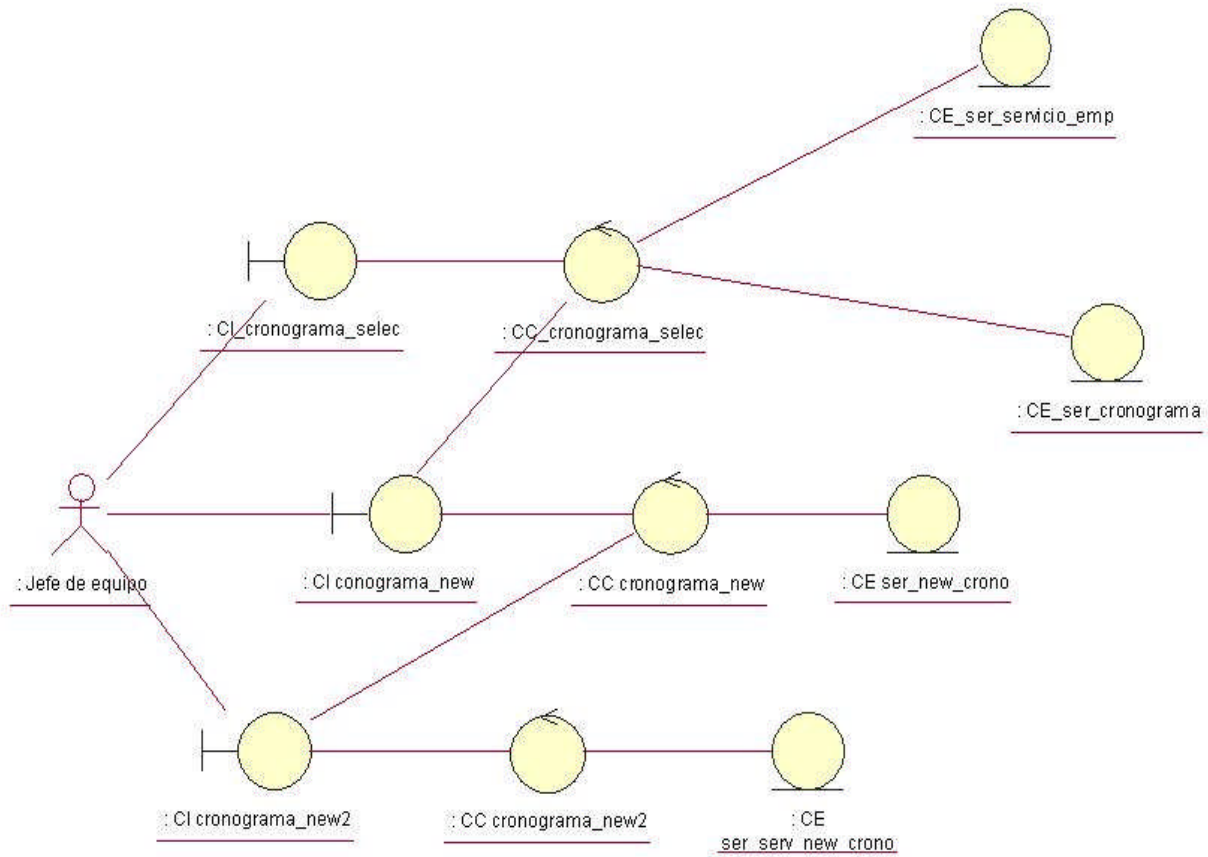
Caso de uso del sistema:	Modificar Modelo de Despacho.
Actores del sistema:	Especialista Principal. (Inicia)
Propósito:	Permitir al Especialista Principal poder realizar un cambio necesario.
Resumen:	El caso de uso se inicia cuando el Especialista Principal se dirige a la opción Modificar Modelo de Despacho. El caso de uso finaliza cuando oprime el botón modificar y cierra la aplicación.
Referencia:	R9.
Precondiciones:	El Especialista Principal debe haberse logueado.
Postcondiciones:	El modelo fue modificado.
Prioridad:	

Caso de uso del sistema:	Registrar Modelo de Despacho.
Actores del sistema:	Especialista Principal. (Inicia)
Propósito:	Permitir al Especialista Principal tener una constancia de la satisfacción del cliente.
Resumen:	El caso de uso se inicia cuando el Especialista Principal se dirige a la opción Registrar Modelo de Despacho. El caso de uso finaliza cuando oprime el botón registrar y cierra la aplicación.
Referencia:	R9.
Precondiciones:	El Especialista Principal debe haberse logueado.
Postcondiciones:	El modelo fue registrado.
Prioridad:	

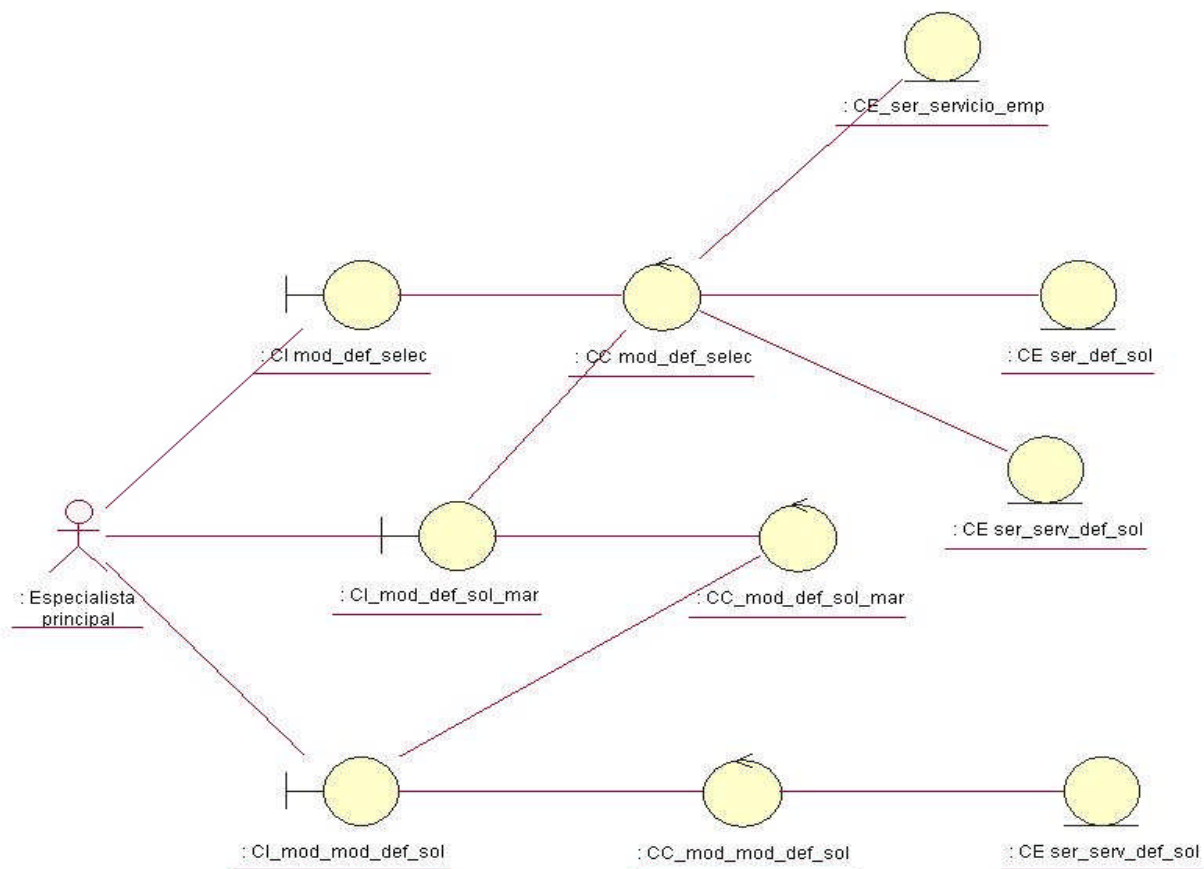
Anexo # 2 Clases del Análisis,



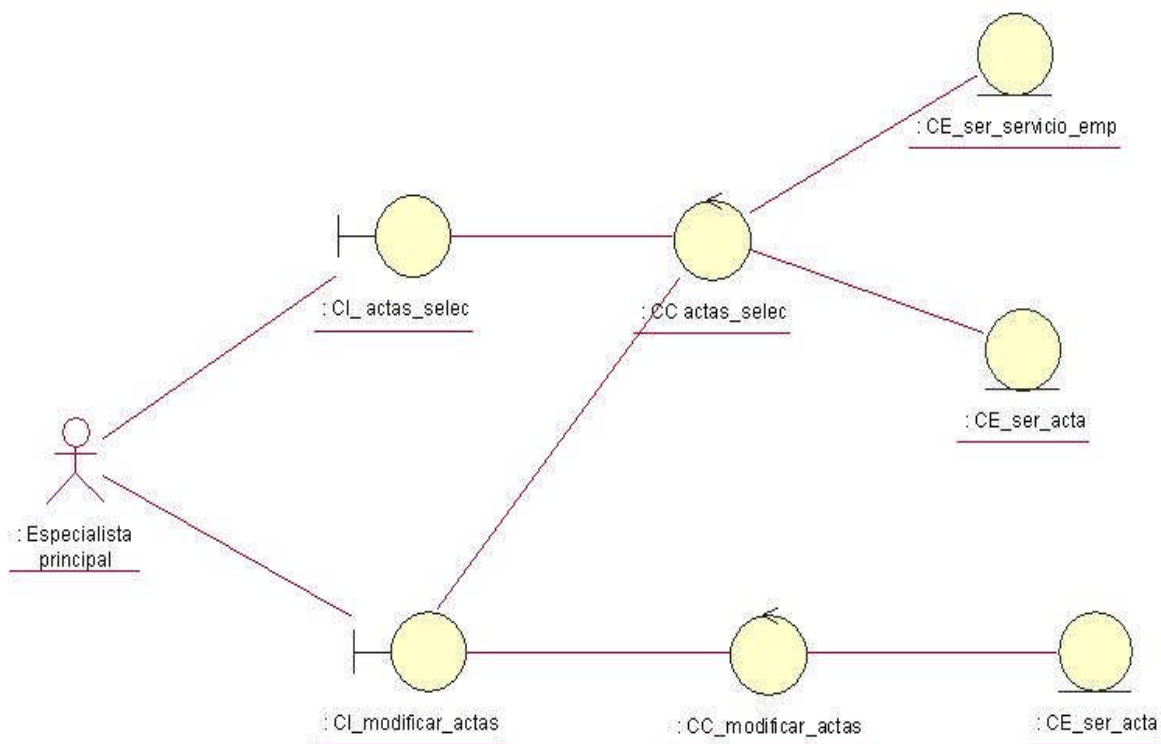
CU Modificar Control de Incidencias en los Servicio.



CU Modificar Cronograma de trabajo



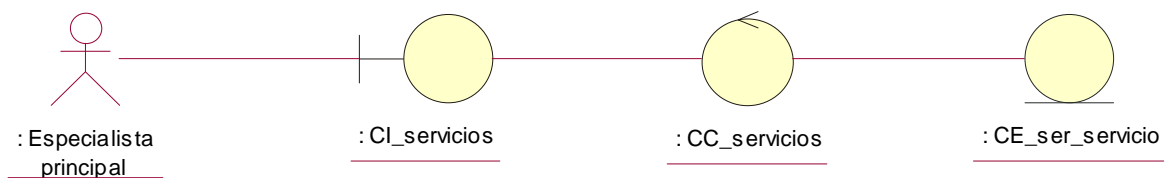
CU Modificar Modelo de Deficiencias y Posibles Soluciones.



CU Modificar Acta del Despacho.

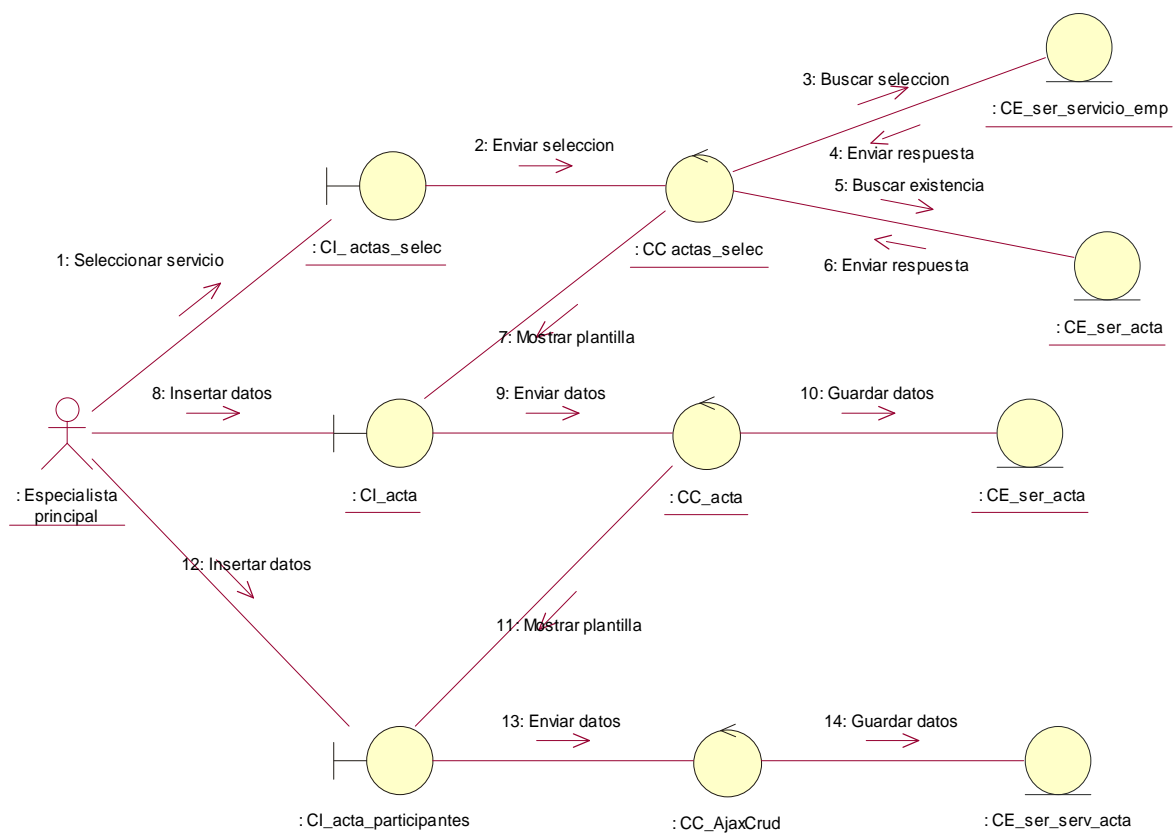


CU Registrar Empresa.

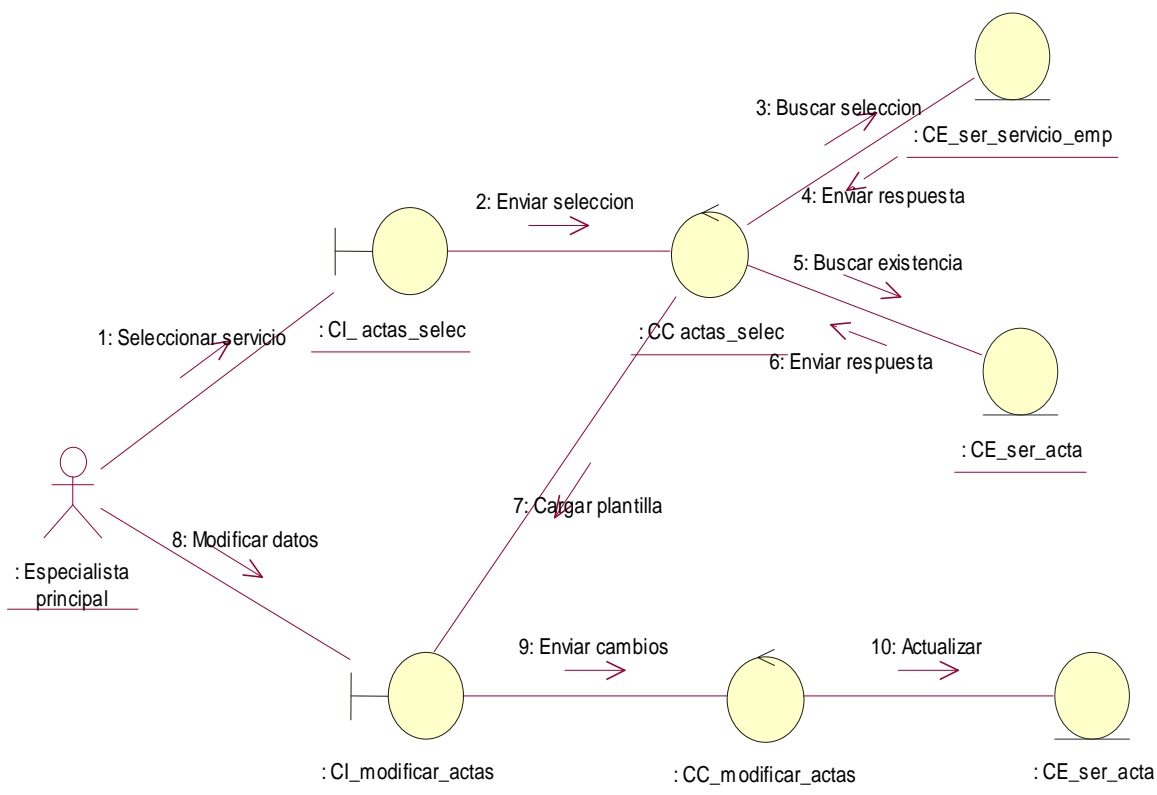


CU Registrar Servicio.

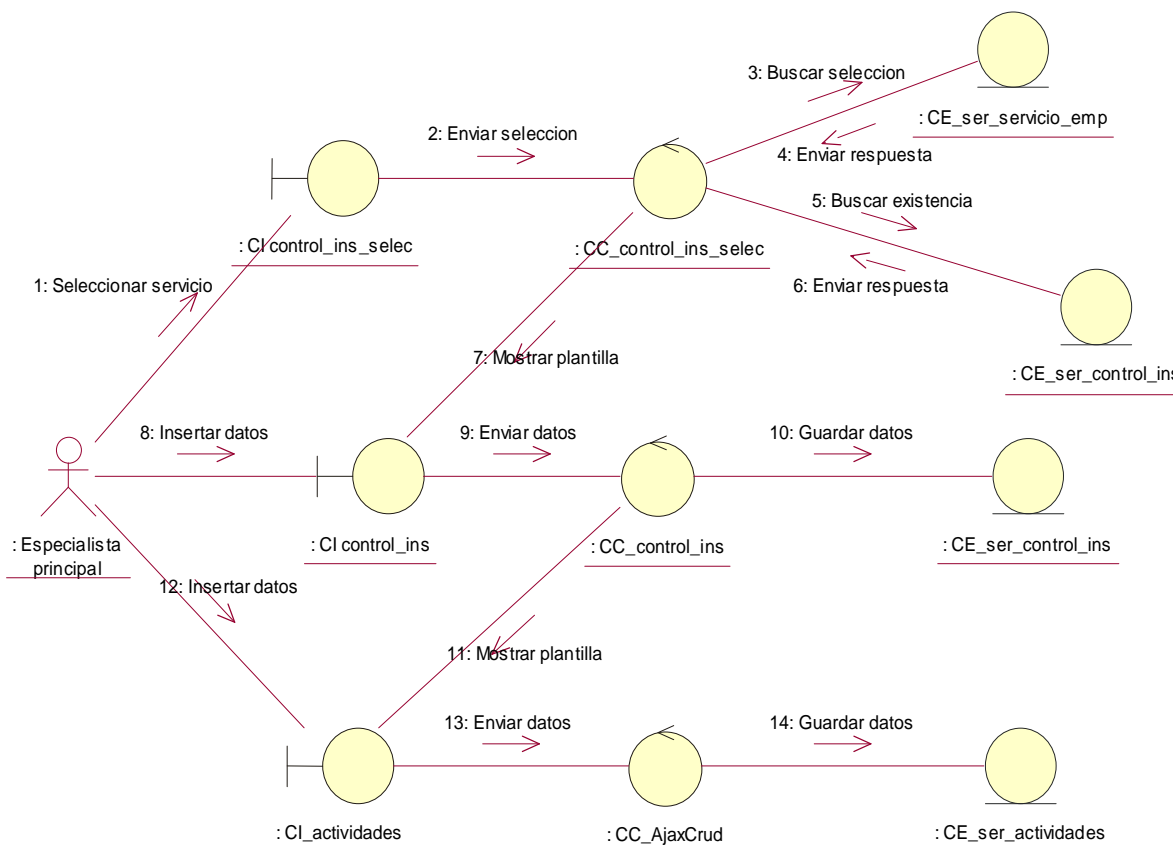
Anexo # 3 Diagramas de colaboración



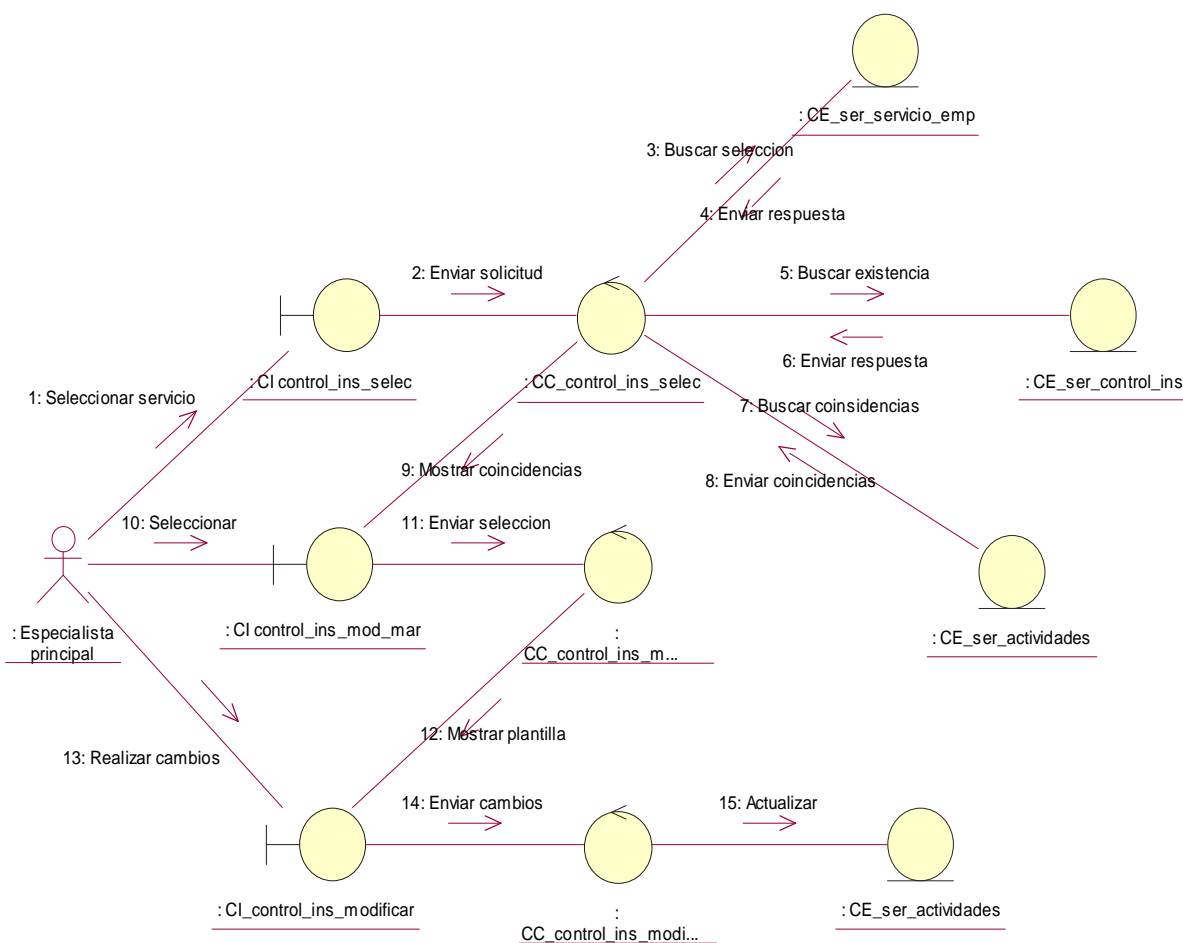
CU Registrar Acta del Despacho.



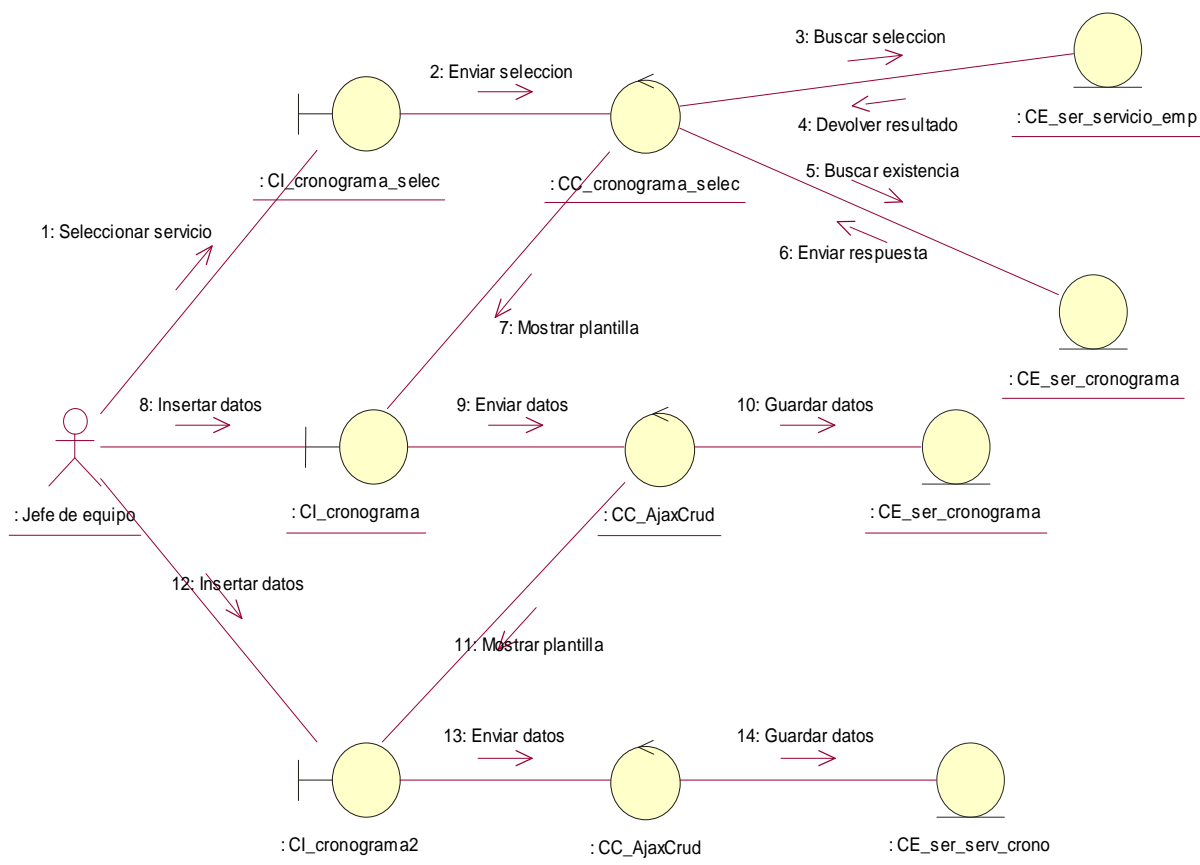
CU Modificar Acta del Despacho.



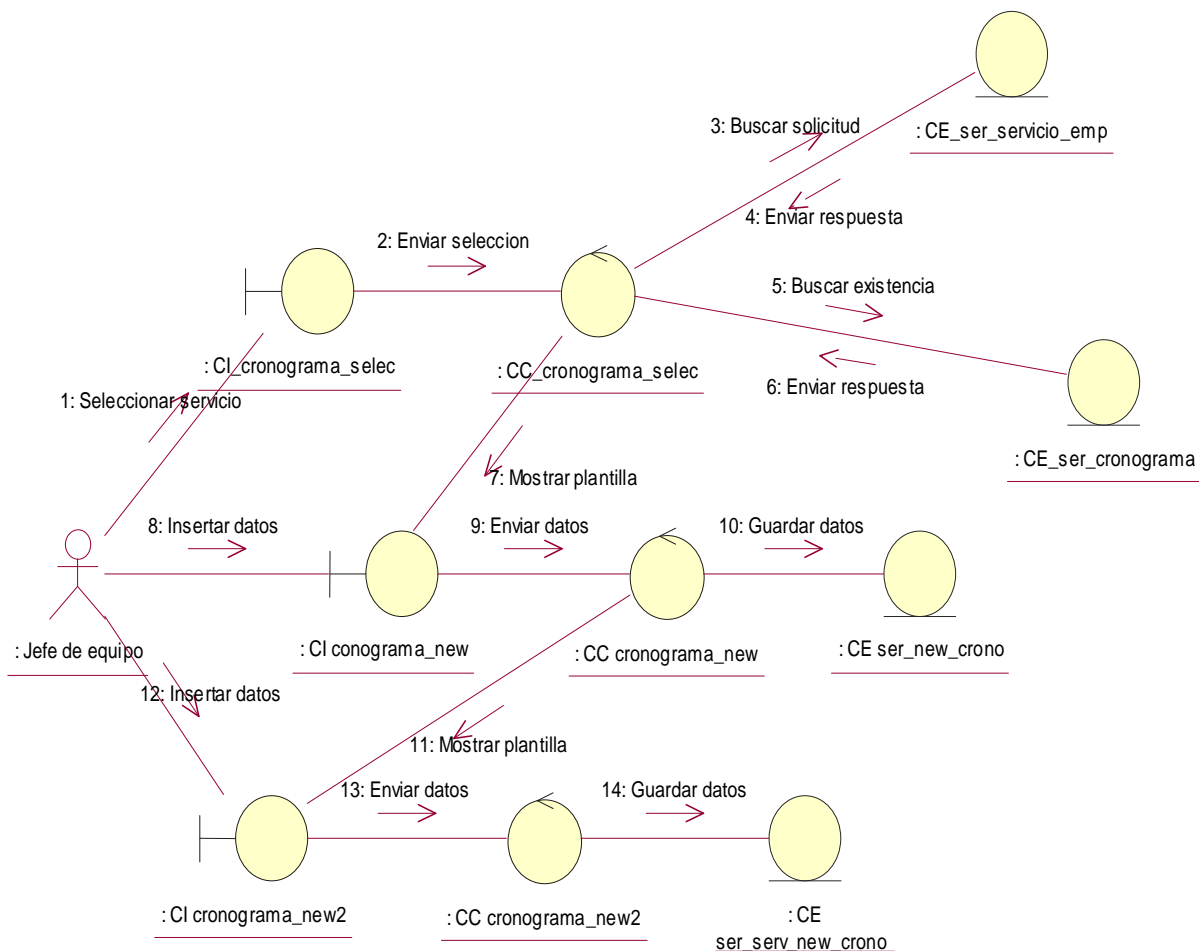
CU Registrar Control de Incidencias.



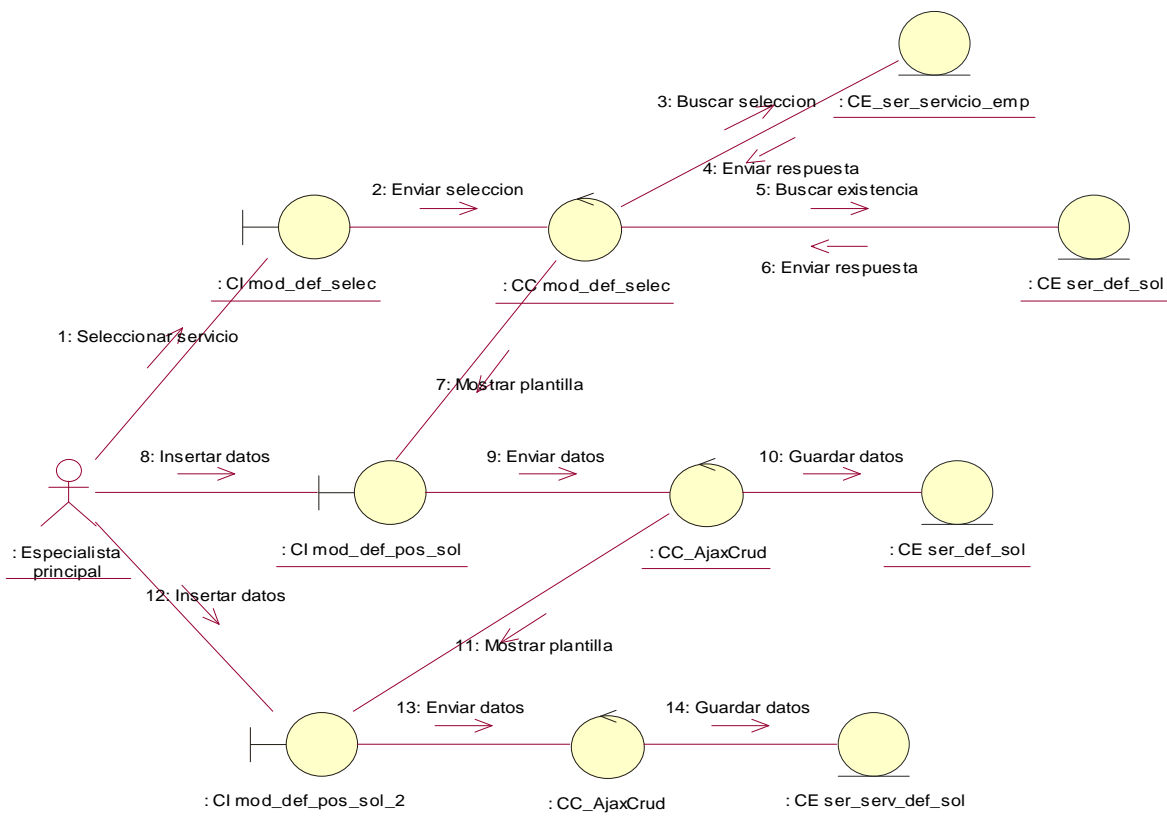
CU Modificar Control de Incidencias.



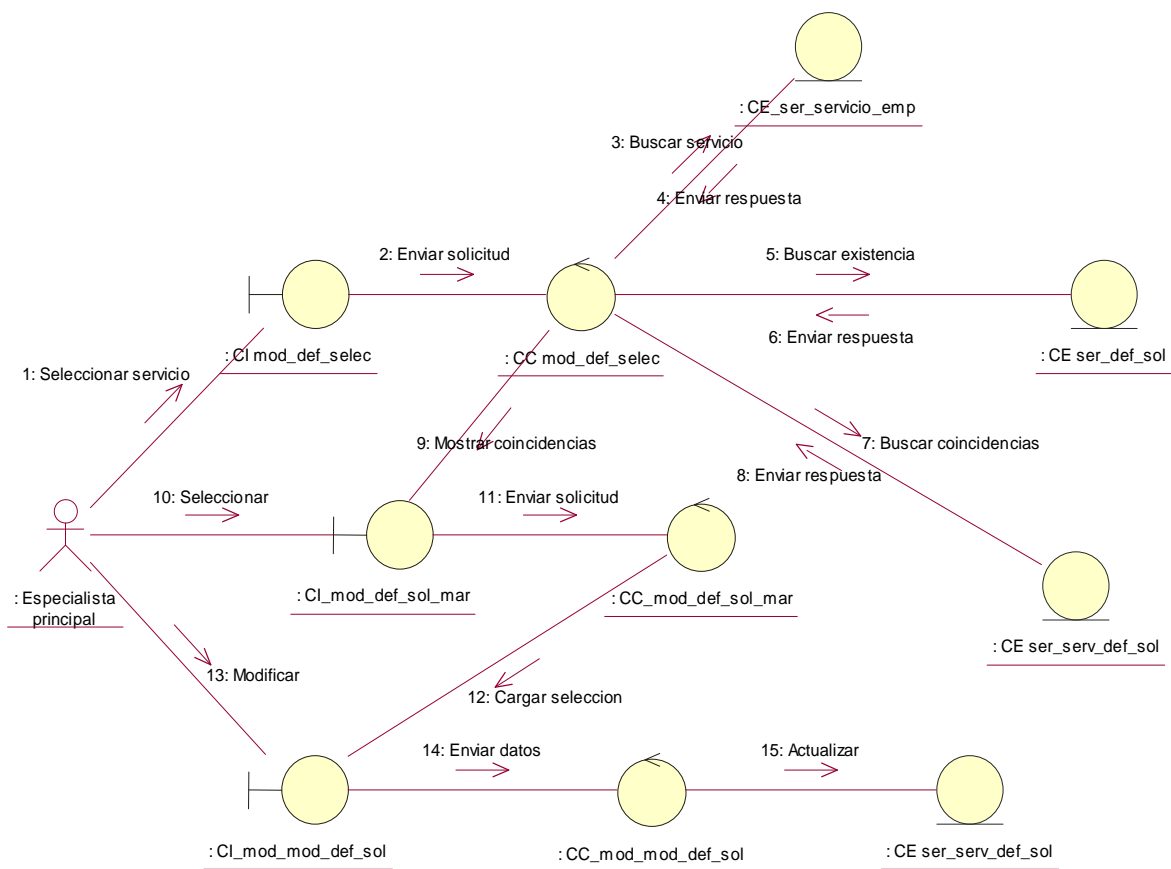
CU Registrar Cronograma de Trabajo.



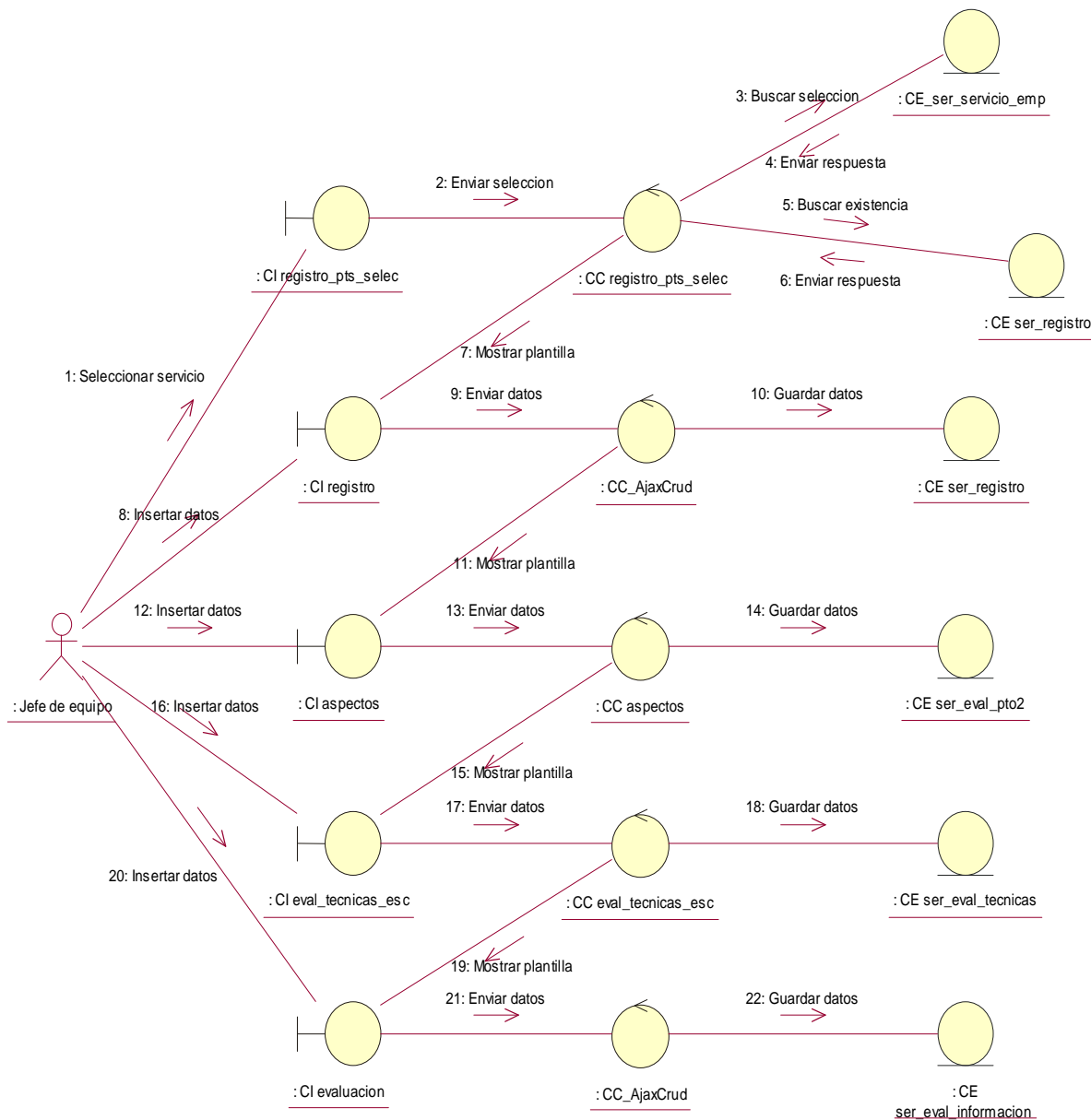
CU Modificar Cronograma de Trabajo.



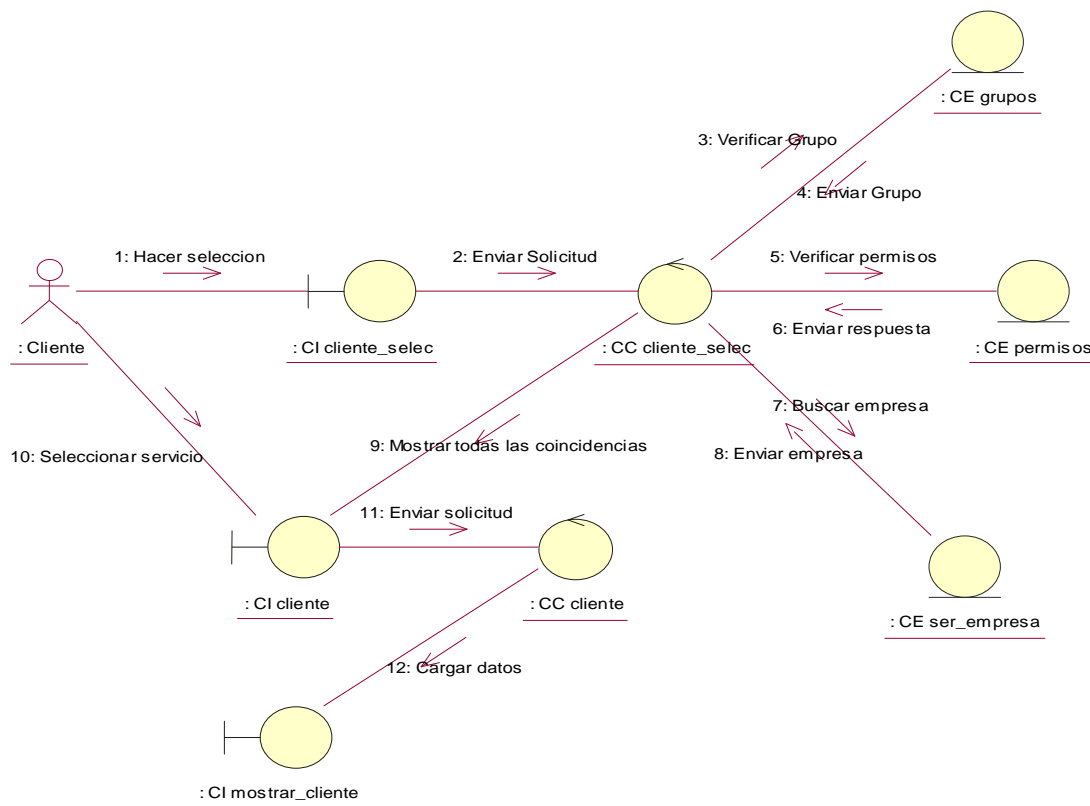
Registrar Modelo de Deficiencias y Posibles Soluciones.



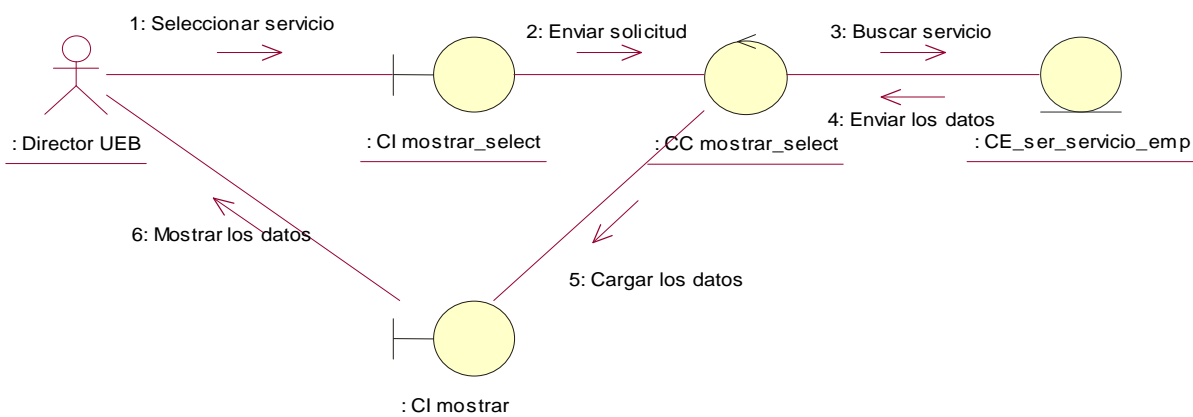
CU Modificar Modelo de Deficiencias y Posibles Soluciones.



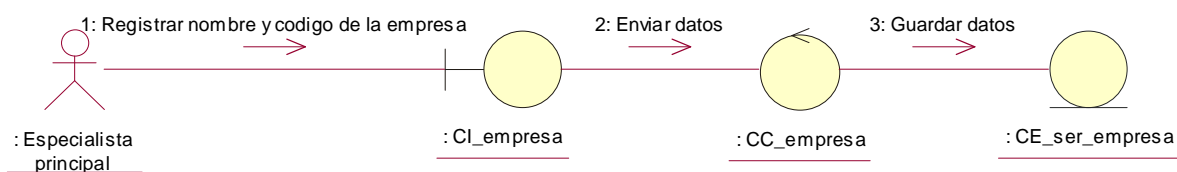
CU Registrar Registro de Puntos de Inspección.



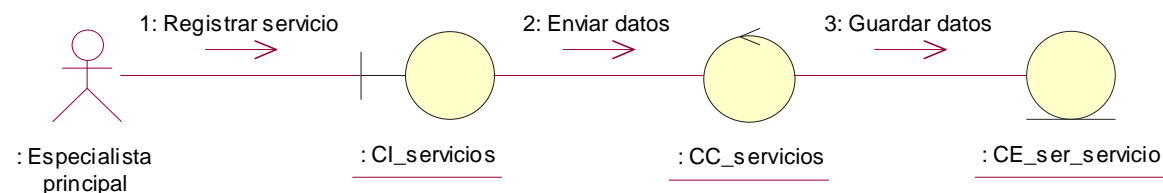
CU Revisar Servicios Prestados.



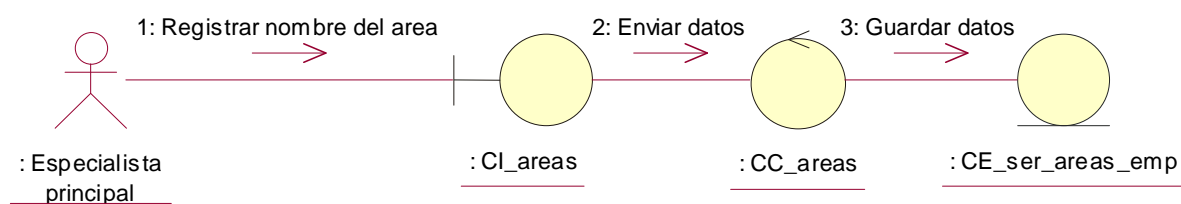
CU Mostrar Servicio.



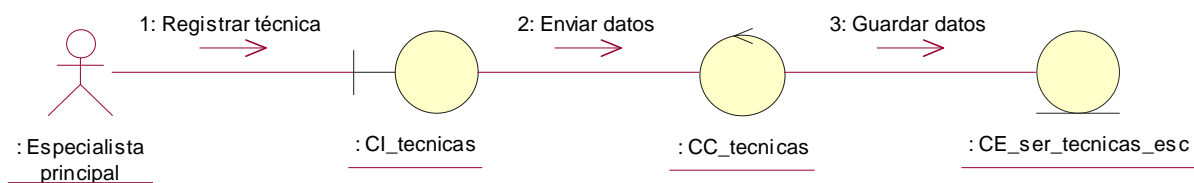
CU Revisar Empresa.



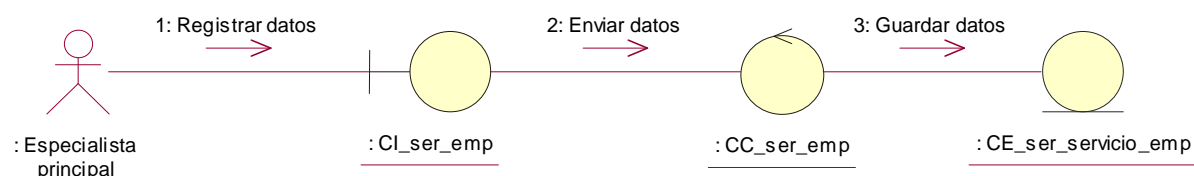
CU Revisar Servicios.



CU Revisar Área.

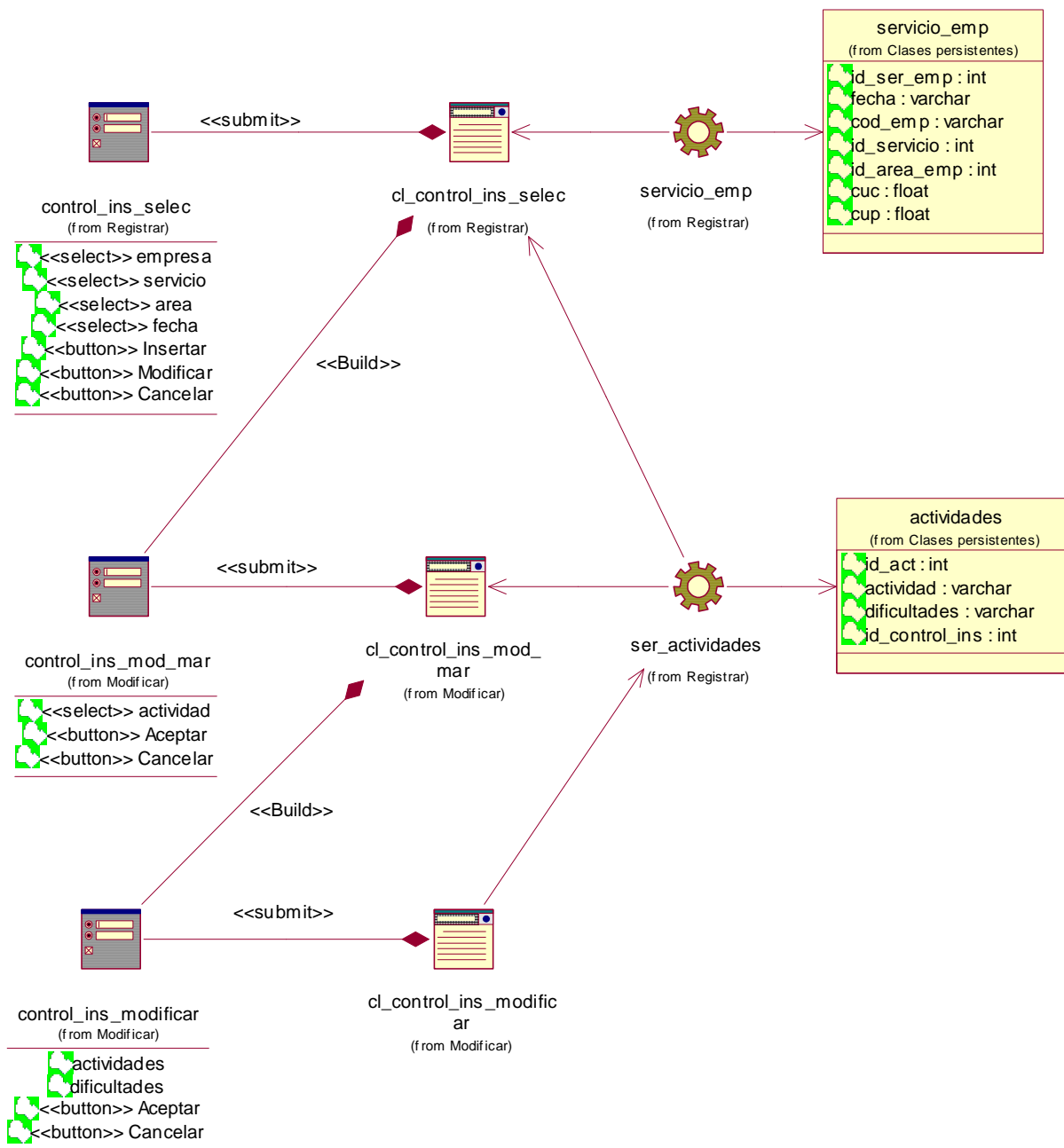


CU Revisar Técnicas y Métodos.

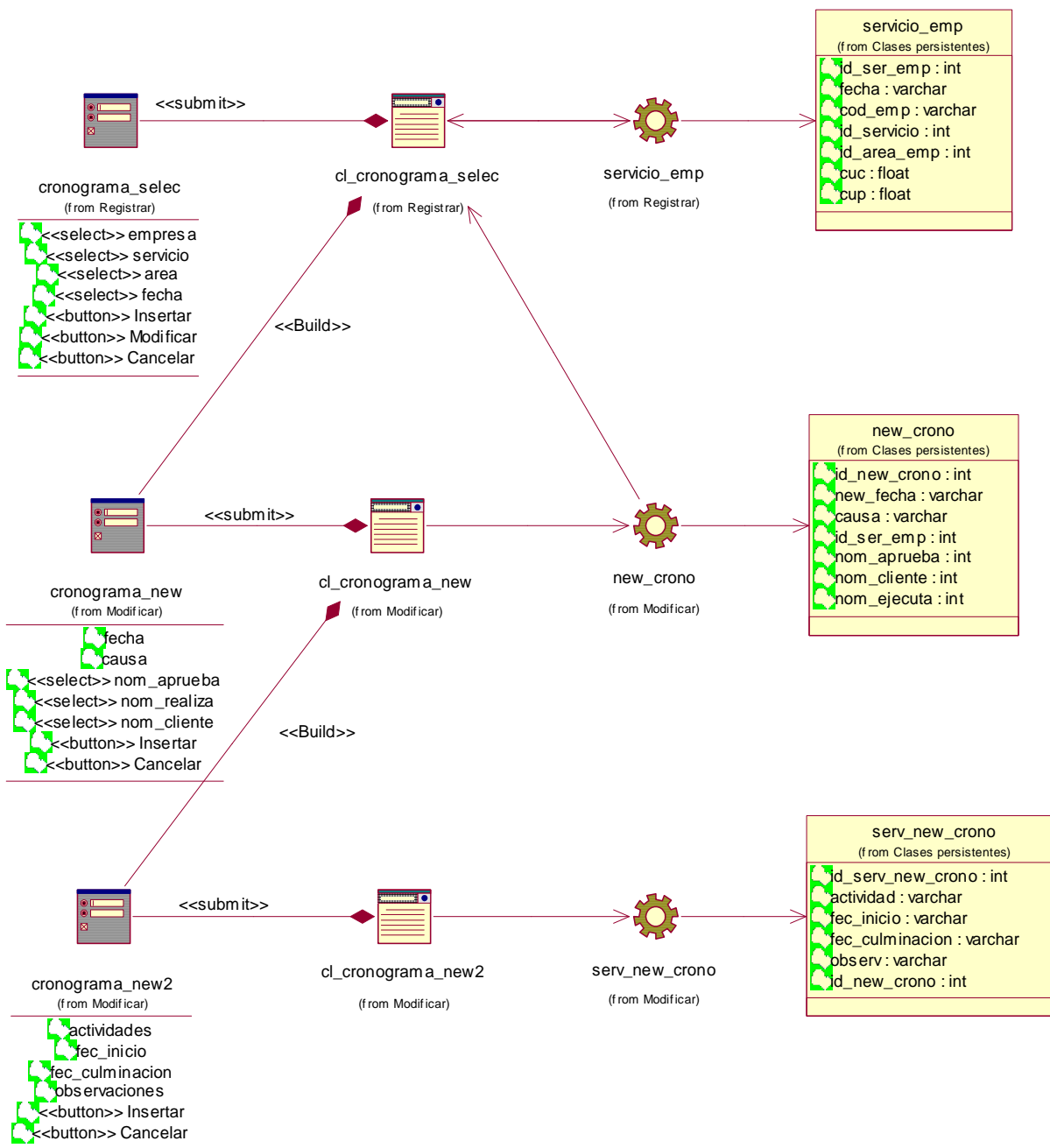


CU Conformar Servicio a Empresa.

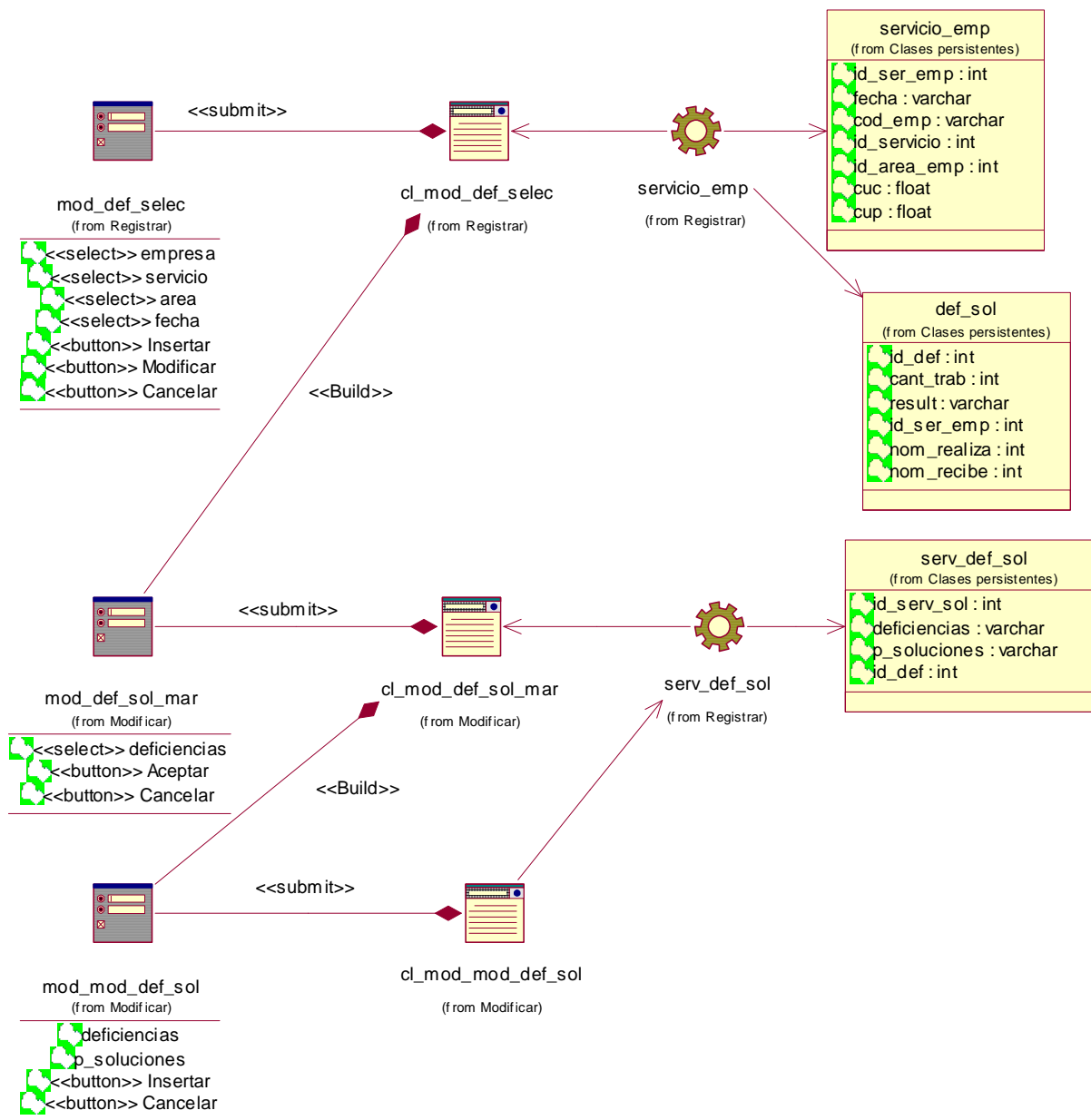
Anexo # 4 Clases del Diseño.



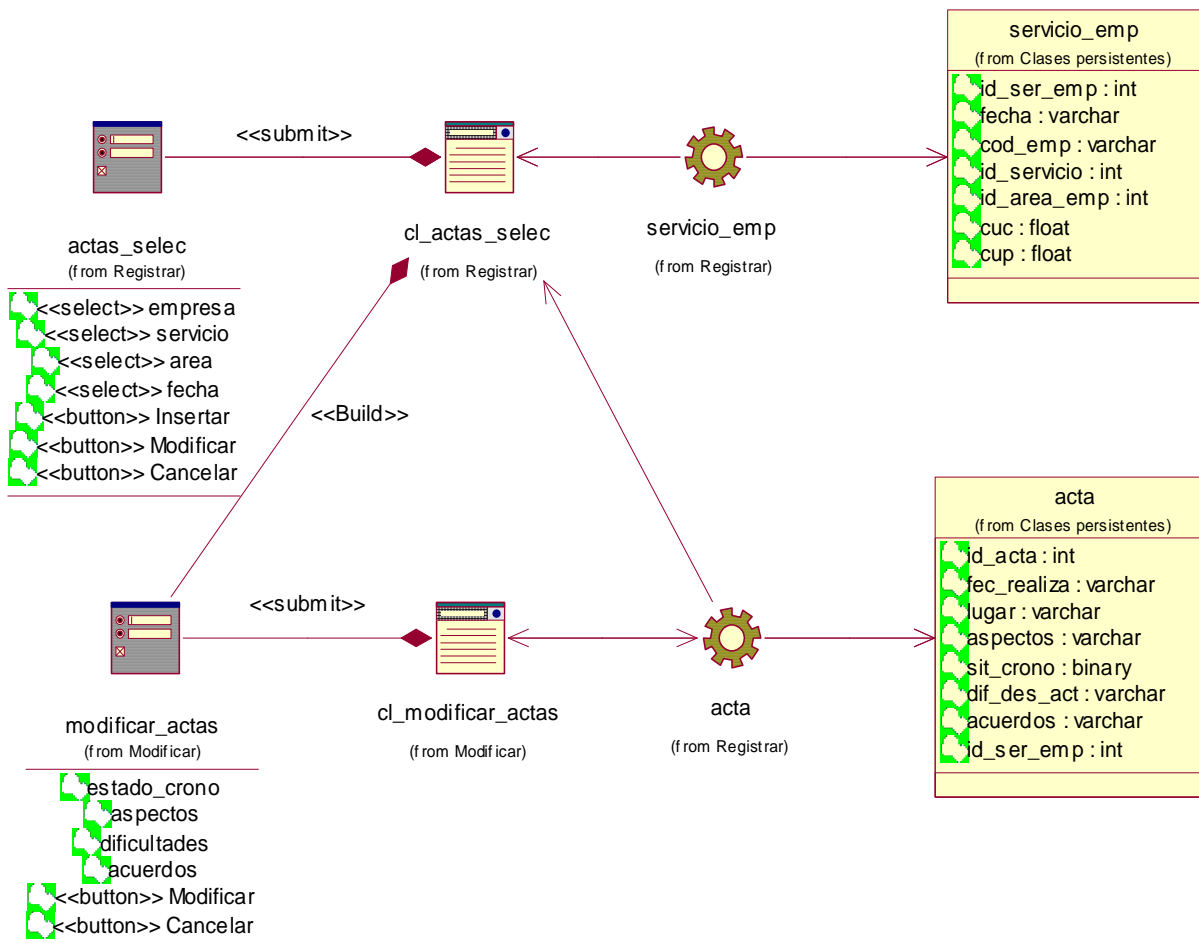
CU Modificar Control de Incidencias.



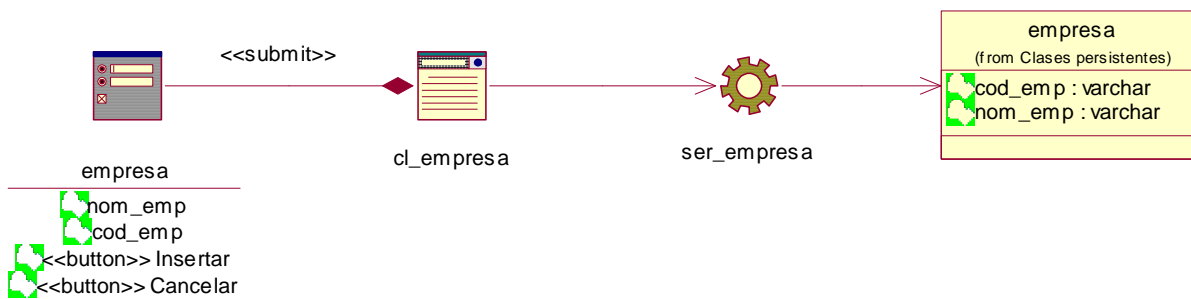
CU Modificar Cronograma de Trabajo.



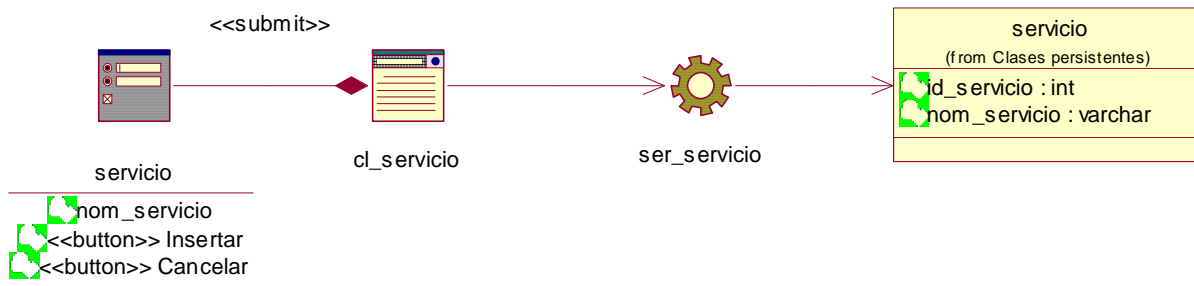
CU Modificar Modelo de Deficiencias y Posibles Soluciones.



CU Modificar Acta del Despacho.



CU Registrar Empresa.



CU Registrar Servicio.

Anexo # 5 Modelo Físico.

