

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



FACULTAD 7

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

**MÓDULO GESTIÓN DE CLIENTES Y FACTURACIÓN DE SERVICIOS
TÉCNICOS EN EL SISTEMA DE GESTIÓN PARA LA INGENIERÍA
CLÍNICA Y ELECTROMEDICINA**

AUTORES

LAIMERYS MIRANDA ARANGO

DAVID PÉREZ-BORROTO LORENZO

TUTOR

ING. RANNIEL RIVERO SEVILA

CO-TUTORA

ING. YISLENYS SUÁREZ HERNÁNDEZ

La Habana, junio de 2011

“Año 53 de la Revolución”

“La responsabilidad nuestra es luchar porque la calidad del producto que aquí se haga sea de las mejores y la mejor posible...”

Ernesto “Che” Guevara



Datos de contacto

Tutor: Ing. Ranniel Rivero Sevilla.

Correo electrónico: rrivero@uci.cu

Ingeniero en Ciencias Informáticas graduado del curso 2007-2008. Líder de desarrollo del proyecto SACCEM durante 1 año. Ha impartido la asignatura de Práctica Profesional (PP) e Inteligencia Artificial, (HI) y Segundo Perfil (SP). En el curso 2008-2009 formó parte de un tribunal en 6 tesis de grado y fue tutor de 3 tesis y en el curso 2009-2010 de una. Actualmente desarrollador del proyecto Electromedicina perteneciente al Departamento Sistema de Apoyo a la Salud del Centro de Informática Médica (CESIM) y profesor adjunto a la Facultad 7.

Co-tutora: Ing. Yislenys Suárez Hernández

Correo electrónico: ysuarezh@uci.cu .

Ingeniera en Ciencias Informáticas, graduada en el 2009. Estuvo al frente del equipo de Auditorías y Revisiones en el grupo de calidad del Centro de Informática Médica (CESIM). Se desempeñó como analista en el proyecto Sistema de Gestión para la Ingeniería Clínica y Electromedicina (SIGICEM) del Departamento Sistemas de Apoyo a la Salud (SAS). Actualmente se desempeña como analista principal de este departamento.

Resumen

El Centro Nacional de Electromedicina (CNE), ha promovido el desarrollo de un sistema informático para llevar el control y mantenimiento de equipos médicos a nivel nacional. Por el alcance y desarrollo del centro se vio la necesidad de crear un sistema que contenga la mayor cantidad de funcionalidades, el cual se nombra, Sistema de Gestión para la Ingeniería Clínica y Electromedicina (SIGICEM), el mismo se lleva a cabo en el Centro de Ingeniería Clínica y Electromedicina (CICEM). Actualmente el sistema permite llevar el control de los equipos médicos, pero no cuenta con una forma de realizar la gestión de clientes y la facturación de servicios técnicos, por lo que se desarrolló el Módulo Gestión de Clientes y Facturación de Servicios Técnicos en el SIGICEM.

Para el desarrollo del módulo se usó Visual Paradigm como herramienta CASE, UML como lenguaje de modelado, el lenguaje de programación PHP y Apache como servidor web, se utilizaron los framework ExtJS y Symfony y el sistema gestor de base de datos MySQL.

El desarrollo del Módulo Gestión de Clientes y Facturación de Servicios Técnicos proporciona muchos beneficios para el CICEM así como en otras instituciones que utilicen dicho módulo, permitirá la optimización del trabajo en los procesos de facturación, se espera un aumento en la velocidad y precisión de la gestión de clientes, además, se podrá llevar a cabo todo el proceso de facturación de los servicios técnicos brindados a clientes.

Palabras claves: Facturación, Gestión de cliente, Equipos Médicos.



Índice

Introducción.....	1
CAPÍTULO 1: Fundamentación teórica	6
1.1 Conceptos fundamentales	6
1.2 Sistema de Gestión de Clientes y Facturación existentes a nivel internacional	7
1.2.1 Sistema de Gestión de Clientes y Facturación existentes en Cuba.....	8
1.3 Técnicas, tecnologías, metodologías y software usados para la solución del problema	10
1.3.1 Proceso Unificado de Desarrollo (RUP)	11
1.3.2 UML 2.1 (Unified Modeling Language)	12
1.3.3 Visual Paradigm for UML 3.6	12
1.3.4 Sistema Gestor de Bases de Datos.....	12
1.3.5 MySQL 5.3.....	13
1.3.6 DB Designer Fork 4.1	13
1.3.7 Sistema Operativo	14
1.3.8 Servidor Web Apache 2.2	14
1.3.9 ExtJS 2.2	14
1.3.10 NetBeans 6.9.....	15
1.3.11 Framework en PHP	15
1.3.12 Symfony 1.4.8.....	16
1.3.14 PHP 5.3.1	16
1.3.15 Otros componentes utilizados.....	17
CAPÍTULO 2: Características del sistema.....	18
2.1 Breve descripción del problema.....	18
2.2 Descripción de los procesos del negocio.....	18
2.3 Modelo de negocio	19
2.3.1 Reglas del Negocio. Módulo Gestión de Clientes y Facturación de Servicios Técnicos.....	19
2.3.2 Justificación de los actores y trabajadores del negocio. Módulo Gestión de Clientes y Facturación de Servicios Técnico	20
2.3.3 Diagrama de caso de uso del negocio.....	20



2.3.4 Descripción de Casos de Uso del Negocio. Módulo de Gestión de Clientes y Facturación de Servicios Técnico.....	21
2.3.5 Diagrama de Actividades	23
2.3.6 Modelo de objetos	24
2.4 Especificación de los requisitos de software	25
2.4.1 Requisitos Funcionales	25
2.4.2 Requisitos No Funcionales	25
3.1 Actor del sistema	27
3.2 Diagrama de Casos de uso del sistema.....	28
3.3 Descripción del Caso de uso Crear factura.....	31
3.4 Descripción del Caso de Uso Crear ficha cliente	31
CAPÍTULO 3: Análisis y Diseño del Sistema	32
3.1 Modelo de Análisis	32
3.2 Diseño.....	35
3.3 Patrones de Diseño	37
3.3.1 Diagrama de clases del diseño	38
3.4 Modelo de Despliegue	40
CAPÍTULO 4: Implementación	42
4.1 Modelo de Implementación	42
4.2 Diseño de la Base de Datos	42
4.3 Tratamiento de errores	44
4.4 Seguridad.....	44
Conclusiones	46
Recomendaciones	47
Referencias Bibliográficas	48
Bibliografía	50
Glosario de términos	53
Anexo1	56
Anexo 2	71



Introducción

En la actualidad el aporte de las Tecnologías de la Informática y las Comunicaciones (TIC) resultan fundamentales en todas las áreas, pero imprescindible en lo que respecta a la medicina. Es considerable que debe existir una interrelación entre medicina y tecnología; pues el manejo de los equipos médicos de alta complejidad es parte de los avances científico-técnicos que se han adquirido a través del tiempo. El desarrollo tecnológico ha propiciado un cambio asombroso en la medicina; su evolución ha permitido conocer infinidad de procesos que explican el porqué de muchas enfermedades que afectan el cuerpo humano y las consecuencias de relacionarse con su entorno. (1)

Cuba, a pesar de no ser un país desarrollado ha dedicado una importante parte de sus recursos al desarrollo de la informática en diferentes esferas de la sociedad, específicamente en el sector de la salud. Este proceso es controlado por el Ministerio de Salud Pública (MINSAP), encargado del desarrollo de las ciencias médicas y la industria farmacéutica. A raíz de esto, comienza la creación de nuevas instituciones de salud con su correspondiente equipamiento, lo cual hace necesaria la creación de pequeños talleres para ofrecer mantenimiento y atención especializada al mismo. Con el desarrollo vertiginoso de la tecnología médica se crea el Centro Nacional de Electromedicina (CNE). Posteriormente, comienza la construcción de los centros provinciales de esta rama y grupos técnicos en hospitales; creados para normalizar, controlar, dirigir y capacitar al personal de salud que laboran en dichos centros.

Por la necesidad de introducir la informática en las unidades de salud, se formó por parte del Sistema Nacional de Salud (SNS) el Grupo de Automatización y Desarrollo de Electromedicina (GADE), el cual tuvo la misión de apoyar la gestión de información, investigación y administración del CNE. Este grupo comenzó el desarrollo de un sistema automatizado que permitiría realizar los procesos llevados a cabo en dicho centro, además de integrar las informaciones de carácter general del SNS; el cual se le denominó: Sistema Integral de Gestión para Electromedicina (SIGEM). Este contó con funcionalidades que llevarían el control del aseguramiento técnico, el inventario y la gestión tecnológica para el procesamiento de la información en el control de equipos médicos y la gestión de los recursos humanos.

Como no existía una correlación entre la ingeniería clínica y la electromedicina, surgió posteriormente el Centro de Ingeniería Clínica y Electromedicina (CICEM), institución que se encarga actualmente de brindar servicios técnicos y gestionar la tecnología médica existente en el SNS. Debido al desarrollo del centro, se percibió la necesidad de crear un nuevo sistema, nombrado Sistema de Gestión para la Ingeniería Clínica y Electromedicina (SIGICEM), quien en la actualidad se encuentra en desarrollo por parte de un equipo de trabajo del Departamento de Sistemas de Apoyo a la Salud (SAS), el cual



conjuntamente a otros departamentos integran el Centro de Informática Médica (CESIM), subordinándose este, a la facultad 7 de la Universidad de las Ciencias Informáticas (UCI).

Como anteriormente el CICEM era una unidad presupuestada, la versión actual del SIGICEM no comprende la gestión de clientes y la facturación de los servicios técnicos que se brindan porque para ese entonces no se requerían. Al pasar a ser una empresa, surge la necesidad indispensable de llevar el control de las prestaciones de servicios a través de la facturación, pues la factura representa un documento que el vendedor entrega al comprador y que acredita que ha realizado una compra o una prestación de servicios por el importe reflejado en el mismo.

Actualmente estos procesos no se llevaban a cabo en la empresa, y una integración con otros sistemas que pudiesen soportar dicha funcionalidad produciría, incumplimiento de los cronogramas pactados por los clientes y elevados costos en cuanto a la prestación de los servicios técnicos brindados, por la no existencia de la gestión de información de clientes y la facturación de los servicios brindados por el CICEM, lo cual ocasiona el no comprometimiento de la entrega de un bien o servicio por parte de este centro.

De acuerdo a la situación descrita anteriormente, se identifica como **problema a resolver**: ¿Cómo contribuir a la gestión de información de clientes y la facturación de los servicios técnicos en el Sistema de Gestión para la Ingeniería Clínica y Electromedicina?

A partir del problema planteado se define como **objeto de estudio**: Proceso de gestión de información en el Centro de Ingeniería Clínica y Electromedicina.

El objeto de estudio delimita el **campo de acción**: Gestión de información de clientes y facturación de servicios técnicos en el Centro de Ingeniería Clínica y Electromedicina.

Para dar solución al problema se define como **objetivo general**: Desarrollar un módulo que permita la gestión de información de clientes y la facturación de los servicios técnicos para el Sistema de Gestión para la Ingeniería Clínica y Electromedicina.

Para dar solución al objetivo general, se plantea desarrollar las siguientes **tareas de la investigación**:

- 1- Caracterizar los sistemas existentes de gestión de información de clientes y facturación de servicios técnicos, obteniendo las bases de la investigación.
- 2- Aplicar las tecnologías y arquitectura definida por el Departamento de Sistema de Apoyo a la Salud.
- 3- Diseñar los artefactos correspondientes a las disciplinas del Proceso Unificado de Desarrollo (RUP), para la obtención de la documentación necesaria.
- 4- Diseñar una base de datos para la gestión de la información.



5- Implementar el Módulo Gestión de Clientes y Facturación de Servicios Técnicos.

Para darle cumplimiento a las tareas propuestas con anterioridad, en el presente trabajo se utilizaron los métodos científicos de la investigación. Estos no son más que una forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones. Además es un medio eficiente para responder y solucionar el problema propuesto. Por la importancia que estos atribuyen a la investigación, se clasifican en teóricos y empíricos, los cuales están dialécticamente relacionados. (2)

Los métodos teóricos permiten estudiar principalmente las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e hipótesis de investigación y crean las condiciones para ir más allá de las características superficiales de la realidad, contribuyendo al desarrollo de las teorías científicas y para su ejecución se apoyan en el proceso de análisis y síntesis. (3)

- **Análisis histórico lógico:** En esta investigación dicho método se usa con el objetivo de estudiar todo lo referente a los sistemas existentes de gestión de información de clientes y facturación de servicios técnicos, en el ámbito nacional e internacional, así como las características fundamentales de las herramientas a utilizar en la investigación. Este método ayuda a dar cumplimiento al estudio del estado del arte tratado en el capítulo 1 Fundamentación teórica, para obtener mayor información del tema.
- **Modelación:** A partir de todo lo investigado y aprendido, se realizan los modelos correspondientes al ciclo de vida del software, que ayudan a dar cumplimiento a las tareas de diseño de los procesos involucrados en la solución. Mediante su utilización se elaboran diferentes tipos de diagramas, a través de la metodología de desarrollo RUP y como lenguaje de modelado UML, para la obtención de un entendimiento detallado acerca del objeto de estudio en cuestión.
- **Analítico-Sintético:** Este método se usa para organizar y sintetizar toda la información obtenida del estudio del estado del arte, las tecnologías, metodologías, lenguajes y herramientas propuestas para la solución del problema, además, enunciar y describir los requerimientos funcionales de la solución planteada. La utilización de este método permite comprender mejor toda la información anterior, la cual será de gran utilidad para lograr un adecuado análisis y diseño del sistema.

Los métodos empíricos describen y explican las características esenciales del objeto de estudio, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional.



- **Grupo de discusiones:** Consiste en una reunión de personas que discuten sobre un tema de interés común, se usa en reuniones realizadas con el personal de proyecto, donde se analizan temas referentes a la investigación propuesta, dichos participantes no tienen como propósito un objetivo específico sino adquirir información sobre un tema o tomar decisiones conjuntas.

Una vez implementado el Módulo Gestión de Clientes y Facturación de Servicios Técnicos facilitará los siguientes beneficios:

- Facilitará el manejo de toda la información referente a la gestión de clientes y la facturación de servicios técnicos ofrecidos a diferentes instituciones, lo que posibilita el desarrollo de la empresa (CICEM).
- Proveerá a los especialistas del Centro de Ingeniería Clínica y Electromedicina (CECIM), la obtención de información que posibilite la planeación de los recursos asociados a la prestación de servicios técnicos, facilitando la administración de los mismos de manera organizada.
- Ofrecerá a los agentes decisores del Centro de Ingeniería Clínica y Electromedicina trazar estrategias para la consolidación de la actividad de atención técnica postmercado, en aras de alcanzar las metas y objetivos de la institución con éxito.

Para lograr un trabajo satisfactorio, se ha estructurado la información expuesta en este documento en cuatro capítulos.

CAPÍTULO 1: Fundamentación teórica

Se aborda el estado del arte del tema a tratar a nivel nacional, internacional y en la universidad. Se explican y justifican los principales conceptos que se abordan, se hace un análisis profundo de la bibliografía relacionada con el objeto de estudio, e incluye una explicación de las técnicas, tecnologías, metodologías y herramientas empleadas en la investigación para darle solución al problema.

CAPÍTULO 2: Características del sistema

Se realiza una breve descripción del problema a resolver y un estudio de los principales procesos que se llevan a cabo en el CICEM, relacionados con los de Gestión de Clientes y Facturación de Servicios Técnicos. Además, contiene información referente a todos los artefactos obtenidos tras haber llevado a cabo el desarrollo de los flujos de trabajo modelamiento del negocio y requerimientos.

CAPÍTULO 3: Análisis y Diseño del Sistema

Se centra en la modelación detallada y la construcción de la estructura del subsistema, mediante la arquitectura definida para dar solución a la investigación propuesta. Se definen los elementos del



diseño, así como la estructura del mismo, se muestran los diagramas de clases del análisis y del diseño de los casos de uso, conformándose finalmente, el modelo de diseño, el cual constituye una base para la futura implementación, además, se obtiene la distribución física del subsistema mediante el diagrama de despliegue.

CAPÍTULO 4: Implementación

Este capítulo está enfocado al flujo de trabajo Implementación para dar solución a los requerimientos funcionales identificados. Se describe la implementación del sistema propuesto a través del diagrama de componentes, así como la descripción de los componentes del sistema.



CAPÍTULO 1: Fundamentación teórica

Este capítulo está dedicado a realizar un análisis detallado sobre el estado del arte de los sistemas de Gestión de Clientes y Facturación de Servicios Técnicos, a nivel nacional e internacional. Se abarcan una serie de conceptos que permiten una mejor comprensión de las terminologías utilizadas. Se describen los lenguajes de programación, la metodología y las herramientas de trabajo a utilizar.

1.1 Conceptos fundamentales

Actualmente el mundo está muy ligado a las tecnologías, en estos momentos casi todo es automatizable debido a la informatización que existe. Cuba se encuentra en vías de desarrollo en cuanto al perfeccionamiento de las tecnologías, por tal motivo en esta época existen muchos sistemas de gestión, por ejemplo; gestión de información, de contenidos, calidad, calificaciones, seguridad, de clientes, entre otros. También existen muchos sistemas de facturación que permiten llevar de forma digital el proceso de facturación de los servicios prestados a clientes. La mayoría de estos sistemas son utilizados para proporcionar mejor eficiencia al desarrollo de las actividades, efectuadas en las diferentes unidades de trabajo.

Sistema de gestión

Un sistema de gestión apoya a alcanzar los objetivos de la organización mediante una serie de estrategias, que incluyen la optimización de procesos. (4)

La implementación de un sistema de gestión eficaz puede ayudar a:

- Mejorar la efectividad operativa.
- Reducir costos.
- Aumentar la satisfacción de clientes y partes interesadas.
- Proteger la marca y la reputación.
- Lograr mejoras continuas.
- Potenciar la innovación. (5)

Sistemas de Gestión de Clientes

Un Sistema de Gestión de Clientes se utiliza para identificar los productos y servicios que estos requieren, para desarrollar estrategias y obtener conocimientos acerca de las necesidades actuales de los mismos, entender a los grupos dominantes de clientes para establecer relaciones a largo plazo con ellos, aumentar las ventas de las empresas y los servicios brindados junto con la satisfacción del cliente.



Factura

La factura es un documento que el vendedor entrega al comprador, que acredita que ha realizado una compra o una prestación de un servicio por el importe reflejado en la misma. (6) Es confeccionado por el proveedor, quien lo ofrece como constancia de la transacción realizada y representa el derecho de cobro de este. La factura incluye datos de los participantes en el acuerdo y las especificaciones de las operaciones.

1.2 Sistema de Gestión de Clientes y Facturación existentes a nivel internacional

A continuación se exponen algunos ejemplos de sistemas de Gestión de Clientes y Facturación de Servicios Técnicos con sus características fundamentales:

Molecule CRM: Solución integrada de atención al cliente: Es una aplicación de gestión de relaciones con el cliente. Su edición Free es totalmente gratuita e incluye muchas de las funcionalidades de Molecule CRM Profesional, a saber, como son: centro de contactos, CRM, incidencias, gestión comercial, marketing, RRHH, proyectos e informes. La generación de gráficas e informes de aspecto profesional ultimán la oferta de Molecule CRM, un excelente paquete de gestión de relaciones con el cliente cuyo objetivo es la instalación, aunque esta es bastante lenta en algunos equipos. Utiliza como gestor de base de datos Microsoft SQL Server Express 2005 o superior incluido en el paquete instalador de la aplicación. El uso y redistribución de SQL Server Express es gratuito, funciona en los sistemas operativos Windows XP, Vista, 7, 2008. (7)

Express Invoice: Es un completo sistema de facturación para empresas de cualquier rama ya que puede facturar mercancías como servicios. Las facturas que emite son totalmente personalizables desde el logotipo hasta los símbolos de moneda y separación de decimales. Todas las facturas emitidas se guardan en un histórico que se puede consultar y pueden estar organizadas por distintas empresas. Además dentro de cada empresa definida se pueden tener sus clientes, géneros o servicios correspondientes. También permite gestionar los cobros de las facturas y emite diversos listados que ayudan a saber el estado de los diferentes cobros, funciona en los sistemas operativos Windows 2000, XP y en Vista 7. (8)

Pagarés y cobros Pro: Es un programa que incorpora una gran cantidad de opciones útiles para llevar un sistema de pagos y cobros, además permite añadir clientes, y para cada uno podrá añadir información sobre los conceptos por los que se le ha de cobrar, por ejemplo fecha de vencimiento, estado de la venta, empresa a la que pertenece entre otras. Puede añadir comentarios personalizados a cada uno, por ejemplo, los motivos por los que no ha pagado. Tiene un calendario, en el que se puede ver todas las



acciones a realizar en cada fecha, rellena automáticamente pagarés bancarios o de empresa (en papel blanco) y al mismo tiempo gestionar el cobro, donde se registran las gestiones realizadas, votos pendientes de cobro, en el cual lo que prima es controlar y gestionar el cobro adecuadamente. (9)

SeniorFactu: Es una completa aplicación de gestión comercial y facturación para PYMES (pequeñas y medianas empresas), de cualquier tipo de actividad. Sus principales características son: Multiusuario, Multiempresa, Multiejercicio, Multialmacén e incluye un extenso número de listados e informes. Permite múltiples niveles en el acceso a la información restringiendo el acceso a los datos. Potente sistema de búsquedas, a través del sistema de filtros. Dispone de una gestión de envíos y control de rutas. Estadísticas multidimensionales definibles por el usuario. Permite conexión con hojas de cálculo Excel, Calc y Correo Electrónico. Admite una amplia configuración del enlace contable. Dispone de un sistema de gestión documental. (10)

FactuSol: Es un completo sistema de facturación orientado a PYMES ofreciendo probar el producto de manera totalmente operativa y sin ningún tipo de limitación. La interfaz ayuda mucho a la hora de realizar cualquier acción, estando todas ellas accesibles desde el menú o, las principales, desde la barra de control. Todo el trabajo se organiza en tres módulos principales: Compras, Ventas y Administración. Pero también incluye numerosas herramientas para informes, catálogos, mantenimiento y de utilidad interna, por ejemplo, calculadora, calendario, tareas y agendas. No incluye ninguna empresa de prueba para ver un resultado avanzado, pero es bastante simple de aprender en poco tiempo. Es multiusuario, multialmacén, multiempresa y multitarifas, con permisos especiales para tipos de usuario. (11)

Después de realizado un estudio acerca de los sistemas existentes a nivel internacional relacionados con la Gestión de Clientes y Facturación, se aprecia que a pesar de los múltiples beneficios, los mismos no se ajustan a las necesidades del SNS, por lo que no son capaces de satisfacer las necesidades del CICEM, porque todos son sistemas propietarios y otros solo están programados según las características propias del sistema, por ejemplo Molecule CRM, Express Invoice funcionan en los sistemas operativos Windows 2000, XP y en Vista 7. Además que en algunos casos no se cumple con las políticas de desarrollo para las aplicaciones del MINSAP que corresponden al uso de tecnologías libres.

1.2.1 Sistema de Gestión de Clientes y Facturación existentes en Cuba

A pesar que Cuba es un país que se encuentra en vías de desarrollo y posee una gran desventaja en cuanto avance tecnológico en general, en las empresas cubanas se ha hecho extensivo el sistema de perfeccionamiento empresarial, cuyas bases generales constituyen una guía y un instrumento de dirección



para lograr la máxima eficacia, eficiencia y competitividad. Actualmente en Cuba existen varias instituciones que emplean los sistemas de gestión de la información de cliente y facturación de servicios técnicos por ejemplo:

Gestión Económica (VINDSOR): Sistema financiero que proyecta, maneja y rastrea con rapidez una amplia gama de facturas, transacciones bancarias, registra las recepciones relacionadas con cobros y pagos y administra el efectivo, así como la ejecución del presupuesto aprobado. (12)

Brinda las prestaciones siguientes:

- Distribución financiera: Registra recepciones relacionadas con los cobros y pagos.
- Administración de efectivos: Captura transacciones bancarias relacionadas con los registros de efectos por cobrar, por pagar, ingresos bancarios, entre otros.
- Presupuesto: Detalla el presupuesto aprobado y obtiene informes a diferentes niveles.

Este sistema fue desarrollado por la Empresa Nacional de Software (DESOFT S.A.), y está actualmente en uso en la Empresa de Suministros Médicos de Cuba (EMSUME). El mismo es utilizado para apoyar el proceso de gestión de la facturación por compraventa de productos médicos dentro de esta organización, aunque no se dedican a la facturación por servicios; ha servido como punto de apertura en el proceso de apoyo para la gestión económica-financiera en la salud.

Sistema Integral Económico Administrativo RODAS XXI: Sistema usado para gestionar e integrar los elementos, procesos y actividades de una organización o negocio que permite: la gestión contable-financiera, producción, desarrollo o investigación, ventas y comercialización, el control de compras, recursos humanos, y gestión de documentos.

Es un sistema que posibilita automatizar el funcionamiento de cualquier empresa o unidad presupuestada, constituye una solución integral al cliente, donde se combina el producto de software con los servicios de atención al cliente, las aplicaciones, que son configurables, tienen un adecuado nivel de integración y están preparadas para suministrar información e intercambiarla entre varios niveles. Está diseñado para operar en una red de computadoras e intranet, donde se intercambia información entre módulos y especialistas que lo operan según las reglas de acceso y seguridad propias de cada cliente, maneja doble moneda, nacional u otras aprobadas para negociar con Cuba.

Actualmente cuenta con siete módulos: Finanzas, Contabilidad, Activos Fijos, Nomina, Inventario, y Facturación. Este último permite realizar las operaciones de ventas tanto de productos como de servicios, incluir descuentos y recargos, generar comprobante, llevar un historial de cada cliente por varios años.



Presenta un grupo amplio de listados y reportes dentro de los que se incluyen: listado de la factura, de clientes por varios criterios, de las ventas, reportes de ventas por clientes o tipos de clientes, ventas por gestores de ventas o vendedores, ventas por productos y/o servicios y de la ficha del cliente.

Plataforma y Arquitectura del Sistema Rodas XXI

- Rodas XXI en estos momentos está soportado sobre la plataforma Microsoft.
- La arquitectura de la aplicación se basa en el modelo cliente - servidor de n-capas (datos, negocios, presentación).
- En el desarrollo del producto se emplea el Proceso Unificado de Rational dentro del proceso de la Ingeniería del Software.
- Para la implementación de las bases de datos del sistema se emplea Microsoft SQL Server.
- Utiliza XML en la transferencia de datos. (13)

Configuración mínima requerida para ejecutar RODAS XXI.

- Computadora IBM PC o compatible.
- Pentium III o superior.
- 256 MB de memoria RAM o más.
- Unidad de disco flexible, un disco duro, lector de Cd y sistema operativo Windows 2000/XP.
- 20GB de espacio en disco duro.

CEDRUX: Sistema que es el primer ERP de Cuba: Es un proyecto que presenta un módulo de facturación de productos y servicios. Además su objetivo es formalizar las ventas mayoristas de productos y prestaciones de servicios que se efectúen, incluyendo las ventas internas que se generen siempre y cuando medien operaciones de cobros y pagos, así como las entregas de productos en consignación o en depósito. Se utiliza además, para formalizar las ventas de activos fijos tangibles y para efectuar devoluciones de productos. Este módulo está integrado a otro de inventario y logística es decir no es independiente y la parte de facturación de servicios tiene muchas funcionalidades que actualmente se encuentran en desarrollo. (14)

1.3Técnicas, tecnologías, metodologías y software usados para la solución del problema

Las tecnologías y herramientas a utilizar para el desarrollo del Módulo Gestión de Clientes y Facturación de Servicios Técnicos, son aquellas que han sido definidas por la dirección del CICEM y el Departamento de SAS.



1.3.1 Proceso Unificado de Desarrollo (RUP)

El Proceso Unificado de Desarrollo es un proceso de software genérico que puede ser utilizado para una gran cantidad de tipos de sistemas de software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de competencia y tamaños de proyectos. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible. Está basado en componentes, lo que significa que el sistema en construcción está hecho de componentes de software interconectados por medio de interfaces bien definidas, usa el Lenguaje de Modelado Unificado (UML). Los aspectos distintivos del Proceso Unificado están capturados en tres conceptos clave: dirigido por casos de uso (use-case driven), centrado en la arquitectura (architecture-centric), iterativo e incremental. Esto es lo que hace único al Proceso Unificado. (15)

Características del Proceso Unificado de Desarrollo RUP

El ciclo de vida de RUP se caracteriza por:

Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta una vez que se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo porque los modelos que se obtienen como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, pues describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.

Iterativo e Incremental: Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada uno de ellos es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada, es por eso que se dice que son mini proyectos.

Para el desarrollo del módulo propuesto se empleará el Proceso Unificado de Desarrollo RUP, este proceso está basado en componentes, además es dirigido por casos de usos, centrado en la arquitectura y su desarrollo es iterativo e incremental. La arquitectura provee la estructura sobre la cual guiar el trabajo en iteraciones, mientras que los casos de uso definen las metas y dirigen el trabajo en cada iteración. De esta manera los desarrolladores obtendrán una visión detallada y organización del trabajo.



1.3.2 UML 2.1 (Unified Modeling Language)

Es un lenguaje de modelado visual que se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema de software. Intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los desarrollos se crea una documentación común, que cualquier desarrollador con conocimientos de UML será capaz de entender independientemente del lenguaje utilizado para el desarrollo. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, pues UML ha sido diseñado para modelar cualquier tipo de proyecto, tanto informático como de arquitectura.

Para el modelado del módulo propuesto se empleará el UML 2.1, pues este lenguaje proporciona a los desarrolladores un mejor entendimiento y asimilación del proyecto, minimiza el tiempo invertido en el desarrollo de la arquitectura, y además agiliza la detección y resolución de errores.

1.3.3 Visual Paradigm for UML 3.6

Visual Paradigm es una herramienta CASE para el modelado UML muy potente, gratuita, fácil de instalar, utilizar y actualizar. Permite dibujar todo tipo de diagramas UML, revertir código fuente a modelos UML, generar código fuente desde los diagramas UML. Incluye los objetos más recientes de UML además de diagramas de casos de uso, diagramas de clase, diagramas de componentes, reversa instantánea para Java, C++, DotNet Exe/DLL, XML, XML Schema, y Corba IDL, ofrece soporte para Rational Rose, integración con Microsoft Visio, además de generar reportes y documentación en HTML/PDF . (16)

Se empleará Visual Paradigm por ser una herramienta que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y documentación.

1.3.4 Sistema Gestor de Bases de Datos

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Existen unos programas denominados Sistemas Gestores de Bases de Datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Los SGBD poseen grandes ventajas entre las que se destacan:

- Facilidad de manejo de grandes volúmenes de información.



- Gran velocidad de ejecución de las consultas.
- Independencia del tratamiento de información.
- Seguridad de la información almacenada. (17)

Se utilizará el SGBD que proporciona a los usuarios la interacción con los datos en términos abstractos, en lugar de tenerlo que hacer directamente con la forma en que esos datos están físicamente almacenados. Al iniciar el diseño de la BD, toda la atención se debe centrar en el aspecto lógico de la información, pues los detalles relacionados con el almacenamiento físico son parte de todo SGBD que se utilice.

1.3.5 MySQL 5.3

MySQL es un gestor de base de datos sencillo de usar e increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales. (18)

Las características principales de MySQL son:

- Es un gestor de base de datos capaz de manejar este conjunto de datos de manera eficiente y cómoda.
- Es una base de datos relacional, la cual se establece una relación para manejar los datos de una forma eficiente y segura.
- Es una base de datos muy rápida, segura y fácil de usar.
- Existe una gran cantidad de software que la usa.

Para la implantación de la BD se usará el gestor de base de datos MySQL 5.3. Es una BD que facilita el trabajo a los desarrolladores por ser eficiente y cómoda, fácil de usar y segura. Además permite definir las tablas que se utilizarán en la futura implantación.

1.3.6 DB Designer Fork 4.1

DB Designer 4.1 es un sistema de diseño visual que integra diseño de base de datos, modelado, creación y mantenimiento en un entorno único, sin figuras. Combina características profesionales y una interfaz de usuario clara y sencilla para ofrecer la manera más eficiente para gestionar sus bases de datos. DB Designer 4.1, es desarrollado y optimizado para el código abierto de MySQL-BD para apoyar a los usuarios de MySQL con una herramienta de gran alcance y libre de diseños disponibles. Todas las



características específicas de MySQL se han construido para ofrecer la forma más conveniente para diseñar y mantener el control de sus bases de datos de MySQL. (19)

Se empleará DB Designer Fork 4.1 que puede ser ejecutado en distintos sistemas operativos, es capaz de generar código para SQL Server, MySQL, Oracle y PostgreSQL (PGSQL). Emplea Licencia Pública General (GPL) pues facilita al usuario la posibilidad de compartir el software, así como realizar cambios en él.

1.3.7 Sistema Operativo

La plataforma de desarrollo que se utilizará para la realización del Módulo Gestión de Clientes y Facturación de Servicios Técnicos será Ubuntu 10.4. (20)

1.3.8 Servidor Web Apache 2.2

El servidor HTTP Apache: es un software libre de código abierto para plataformas Unix (BSD, GNU/Linux.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. El servidor Apache se desarrolla dentro del proyecto HTTP Server (HTTPd) de la Apache Software Foundation. (21)

El servidor web Apache 2.2 proporciona contenidos al cliente web o navegador como:

- **Páginas estáticas:** es el uso más generalizado que se hace de un servidor web. De esta forma se transfieren archivos HTML, imágenes, entre otros.
- **Páginas dinámicas:** la información que muestran las páginas que sirve Apache cambia porque se obtiene a partir de consultas a bases de datos u otras fuentes de datos. Son, por tanto, páginas con contenido dinámico, cambiante.

Se utilizará como servidor Web el Apache 2.2 que corre sobre una multitud de plataformas y sistemas operativos y en específico es el servidor web por el cual correrá el subsistema a desarrollar.

1.3.9 ExtJS 2.2

Framework de presentación JavaScript del lado del cliente para el desarrollo de aplicaciones web. Ofrece múltiples opciones para el trabajo con las validaciones y manejo de errores en el cliente. La personalización de temas de estilos es posible en su utilización, además que provee el trabajo con una



amplia configuración e intenso trabajo con las hojas de estilo CSS. Cuenta con dos licencias, una comercial y otra Open Source. Es una herramienta multiplataforma, integrada a la tecnología AJAX, pues permite evita las recargas de páginas completas al solicitarse nuevos contenidos. Actualmente cuenta con una extensa comunidad de usuarios.

Ventajas del ExtJS 2.2

- Permite la reutilización de código.
- Independiente o adaptable a Framework diferentes (prototype, jquery, YUI).
- Orientada a la programación de interfaces tipo escritorio en la web. (22)

Como framework de presentación del lado del cliente se utilizará ExtJS 2.2. Ofrece múltiples opciones para trabajar con las validaciones así como el manejo de errores en el cliente. Es multiplataforma y adaptable a framework diferentes como prototype, YUI y jquery, además, cuenta con dos licencias; Comercial y Open Source.

1.3.10 NetBeans 6.9

NetBeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. El mismo es un producto libre y gratuito sin restricciones de uso. (23)

La integración con Symfony permite desarrollar aplicaciones de forma más sencilla y productiva. En primer lugar, es posible crear nuevos proyectos y aplicaciones directamente desde el IDE. También se pueden ejecutar todas las tareas de Symfony, incluso pasándole argumentos y opciones, visualizando el resultado sin necesidad de utilizar una consola de comandos externa. (24)

Como entorno de desarrollo se utilizará NetBeans 6.9. Permite ejecutar todas las tareas de Symfony pasándole argumentos y opciones determinados. Además, es una herramienta que permite a los programadores escribir, compilar, depurar y ejecutar programas.

1.3.11 Framework en PHP

Los framework ayudan en el desarrollo de software, proporcionan una estructura definida la cual ayuda a crear aplicaciones con mayor rapidez a la hora de realizar el mantenimiento del sitio, gracias a la organización durante el desarrollo de la aplicación. Son desarrollados con el objetivo de brindarles a los



programadores y diseñadores una mejor organización y estructura a sus proyectos. Se utiliza la Programación Orientada a Objetos (POO), que permite la reutilización de código.

1.3.12 Symfony 1.4.8

Symfony es un completo framework diseñado para optimizar la creación de aplicaciones Web, gracias a sus características. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Está desarrollado en PHP 5, es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft.

Características de Symfony 1.4.8.

- Fácil de instalar y configurar en la mayoría de las plataformas.
- Independiente del sistema gestor de bases de datos.
- Basado en la premisa de *"convenir en vez de configurar"*; en la misma el desarrollador solo debe configurar aquello que no es convencional.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros. (25)

1.3.13 Lenguaje de programación del lado del servidor

Los lenguajes del lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato fácil de entender.

1.3.14 PHP 5.3.1

PHP, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Su primer uso fue el de mantener un control sobre quien visitaba su currículum en su web. Es el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web. La versión más reciente de PHP es la 5.3.1 (para Windows).

Ventajas de PHP 5.3.1



- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Tiene manejo de excepciones (desde PHP5).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. (26)

1.3.15 Otros componentes utilizados

Componentes SIGICEM

Durante el desarrollo del sistema han sido concebidas varias clases auxiliares que agilizan las tareas más comunes realizadas, con el objetivo principal de agilizar el proceso de desarrollo en el Sistema de Gestión para la Ingeniería Clínica y Electromedicina (SIGICEM) y tratar de disminuir el tiempo utilizado para el desarrollo de los implementadores de dicho sistema.

Se cuenta con funcionalidades que se encargan del trabajo con las interfaces visuales, haciendo que la creación de componentes incluidos en el framework de presentación ExtJS sean gestionados de una forma más fácil desde el framework de desarrollo Symfony. También posee clases externas que han sido ajustadas a las pautas de uso del sistema, por ejemplo, aquellas que permiten la importación y exportación de documentos en formatos XLS y PDF. (27)

En este capítulo se realizó el análisis de los sistemas existentes de gestión de información de clientes y facturación de servicios técnicos a nivel nacional e internacional, arribándose a la conclusión que los sistemas existentes no establecen una solución factible para aplicarlos en el CICEM, porque abarcan parcialmente al objeto de estudio en cuestión. Además, la mayoría de los sistemas internacionales investigados son software propietarios, porque se dificulta su adquisición en el mercado. Por las razones expuestas durante este capítulo se decide desarrollar una solución nacional que facilite la gestión de información de clientes y la facturación de los servicios técnicos. Para ello se utilizará la metodología RUP, el lenguaje de programación PHP, el framework Symfony y EXT JS, las herramientas Visual Paradigm y DB Designer 4.1 para el modelado del sistema; así como para el diseño de la base de datos.



CAPÍTULO 2: Características del sistema

En este capítulo se realiza una breve descripción del problema planteado. Se describen los procesos que serán objeto de automatización en el CICEM, los cuales se representan a través del modelo de negocio y sus diagramas correspondientes, describiéndose los actores y trabajadores que intervienen en el mismo. Además, se presentan las reglas del negocio, detallándose los requisitos funcionales, los mismos no son más que las capacidades o condiciones que el sistema debe cumplir; así como se describen los requisitos no funcionales, los cuales constituyen propiedades o cualidades que se deben tener presentes para una buena utilización del módulo de facturación. Finalmente se muestra el diagrama de casos de usos del sistema.

2.1 Breve descripción del problema

El Centro de Ingeniería Clínica y Electromedicina (CICEM) es la institución encargada de brindar servicios técnicos y llevar a cabo la gestión tecnológica de los equipos médicos del SNS. Este centro sufrirá grandes cambios como institución, debido a que se establecerá como empresa. Anteriormente, el centro era una unidad presupuestada, por lo que la versión actual del Sistema de Gestión de Ingeniería Clínica y Electromedicina no comprende la gestión de clientes y la facturación de los servicios técnicos que se le brindarán a diferentes instituciones.

Para la versión actual de SIGICEM se hace imprescindible el desarrollo de la solución propuesta (Módulo Gestión de Clientes y Facturación de Servicios Técnicos). Se debe tener en cuenta que una integración con otros sistemas que soporten las funcionalidades requeridas, traería como consecuencia elevados costos en cuanto al desarrollo de esta actividad y significativos retrasos en el cumplimiento de los cronogramas pactados con el cliente. Por lo que se decide que el módulo a implementar se desarrolle por el grupo de trabajo del proyecto SIGICEM.

2.2 Descripción de los procesos del negocio

Un vez que el cliente solicita servicios técnicos de equipos en la empresa, el especialista en facturación verifica si el mismo ha sido registrado anteriormente, de no ser así, el especialista registra sus datos en la ficha de cliente, una vez que el especialista llena la ficha cliente realiza el contrato a dicha solicitud, registra la fecha de inicio y la fecha de fin de la prestación de servicio en la tarjeta de contrato. Luego el cliente al recibir la prestación de servicio por parte del CICEM, solicita realizar el pago de los servicios adquiridos. El especialista en facturación verifica los datos del cliente para realizar la factura. Finalmente el cliente realiza su pago correspondiente al importe reflejado en la factura.



Mediante este flujo de eventos existe la posibilidad de, no brindar los servicios técnicos con la calidad y rapidez requerida por el cliente, puede existir mala interpretación de la información, puede ocasionar pérdidas de información, además, ocurrir una alteración de importes en el pago de la factura.

2.3 Modelo de negocio

El primer flujo de trabajo que propone RUP para el desarrollo de un sistema informático es el modelado del negocio. El cual tiene como objetivos fundamentales:

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- Percibir 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.

Para lograr los objetivos presentes el proceso de modelado posibilita obtener una visión de la organización, que permite definir los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos.

2.3.1 Reglas del Negocio. Módulo Gestión de Clientes y Facturación de Servicios Técnicos

Describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. El Módulo Gestión de Clientes y Facturación de Servicios Técnicos posee reglas específicas a tener en cuenta a la hora de modelar los casos de usos que de él se derivan, previendo que puedan ser controlados para que el negocio no colapse. A continuación se muestran las reglas fundamentales a cumplir:

1. Para poder solicitar un servicio técnico, el cliente tiene que estar registrado teniendo en cuenta los datos pertenecientes a la ficha cliente.
2. Los especialistas tendrán acceso a la información de los equipos médicos correspondientes a la unidad de salud a la cual pertenecen.
3. El especialista verifica que el contrato presente el formato correcto para no incurrir en inconformidades a la hora de realizar la factura.
4. A la hora de realizar una factura en la empresa el principal responsable de realizar la operación es el gerente de la empresa, para una mejor seguridad, aunque pueden existir otros funcionarios responsables de realizar la facturación en la empresa, como el especialista en facturación.



- Una vez que el especialista en facturación elimina todas las órdenes de servicio de la factura, por cualquier motivo dicha factura desaparece.

2.3.2 Justificación de los actores y trabajadores del negocio. Módulo Gestión de Clientes y Facturación de Servicios Técnico

Actores del negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega una vez que se interactúa con el negocio para beneficiarse de sus resultados (Ver tabla 1).

Actores del negocio	Justificación
Cliente	Representa a todas las unidades de salud o entidades que se benefician con los servicios que brinda el Centro de Ingeniería Clínica y Electromedicina (CICEM), son los que pagan los servicios técnicos de equipos médicos solicitados, también solicitan el contrato de los servicios a obtener.

Tabla #1: Actor del negocio.

Trabajador del negocio

Un trabajador del negocio representa un rol desempeñado en las realizaciones de casos de uso, éste por lo general colabora con otros trabajadores, es notificado de los eventos del negocio y manipula las entidades del negocio para realizar sus responsabilidades (Ver tabla 2).

Trabajador del negocio	Justificación
Especialista en facturación	Representa al especialista en facturación, que se encarga de realizar el contrato a la unidad de salud y confeccionar la factura de los pagos de servicios técnicos.

Tabla #2: Trabajador del negocio.

2.3.3 Diagrama de caso de uso del negocio

Un diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio. A continuación se presenta el diagrama de casos de uso del negocio correspondiente a dicho proceso.

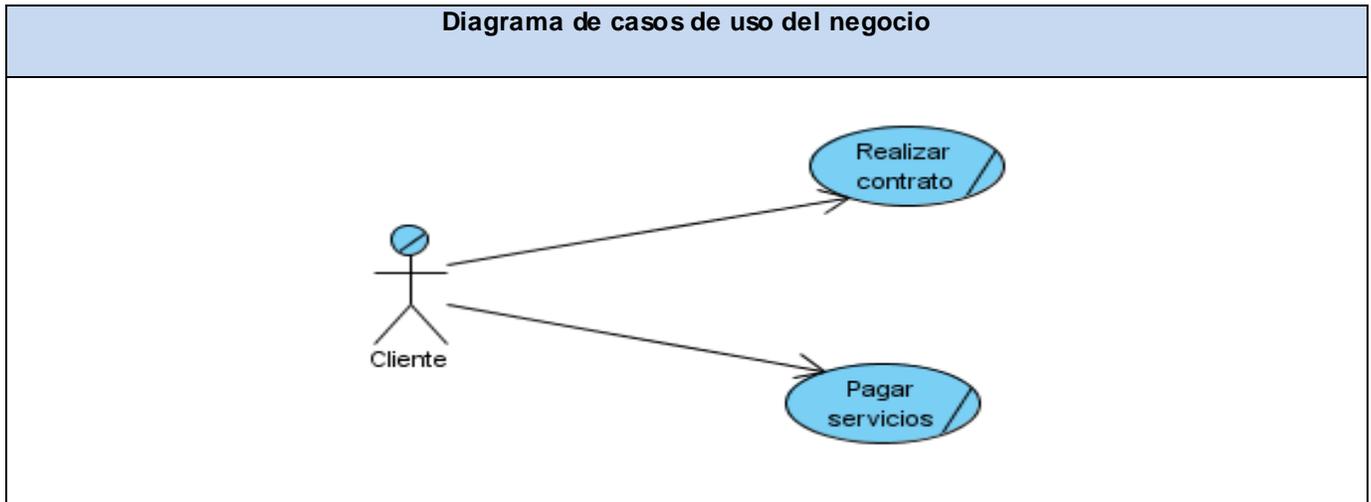


Figura # 1: Diagrama de casos de uso. Módulo de Gestión de Clientes y Facturación de Servicios Técnicos.

2.3.4 Descripción de Casos de Uso del Negocio. Módulo de Gestión de Clientes y Facturación de Servicios Técnico

A continuación se describen los casos de usos del negocio identificados.

➤ **Descripción del Caso de Uso: Realizar contrato**

Caso de Uso:	Realizar Contrato
Actores:	Cliente
Trabajadores:	Especialista en facturación
Resumen:	El caso de uso inicia una vez que el cliente solicita el servicio técnico de equipos médicos, luego el especialista en facturación verifica que el cliente se encuentre registrado y recoge los datos del contrato como son la fecha de inicio y fecha de fin, concluye así el caso de uso.
Precondiciones:	El cliente debe estar registrado.
Flujo Normal de Eventos	
Sección	
Acción del Actor	Respuesta del Negocio
1-El cliente solicita la creación del contrato de los servicios técnicos.	2. Verifica que el cliente esté registrado.



	<p>3. El especialista recoge los datos que requiere el contrato.</p> <p>4. Crea la tarjeta del contrato.</p> <p>5. Verifica si ya ese contrato se realizó y crea el informe del contrato.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	<p>2.1. Si el cliente no está registrado, el especialista en facturación registra lo registra según los datos del mismo en la ficha cliente.</p> <p>5.1 Si el contrato no se ha realizado se realiza el mismo.</p>
Pos condiciones	Creación de la tarjeta del contrato con todos los datos que requiere.

Tabla #3: Caso de uso "Realizar contrato"

Descripción del Caso de Uso: Pagar Servicios

Caso de Uso:	Pagar servicios
Actores:	Cliente
Trabajadores:	Especialista en facturación
Resumen:	El caso de uso inicia una vez que el cliente solicita realizar el pago de los servicios que ha adquirido. El especialista solicita los datos del cliente para realizar la factura. El cliente realiza su correspondiente pago, concluye así el caso de uso.
Precondiciones:	Que el cliente esté registrado.
Flujo Normal de Eventos	
Sección	
Acción del Actor	Respuesta del Negocio
1-El cliente solicita realizar el pago de los servicios obtenidos.	<p>2. El especialista recoge los datos que requiere la factura.</p> <p>3. Llena la factura de servicios prestados.</p>



4-El cliente realiza el pago correspondiente.	5. El especialista deja registrado el pago del servicio brindado.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
Pos condiciones	Confección de la factura de pago.

Tabla #4 Caso de uso "Pagar servicio"

2.3.5 Diagrama de Actividades

Un diagrama de actividades representa el comportamiento interno de un caso de uso, bajo la forma de un desarrollo por etapas, agrupadas secuencialmente. Por lo que los casos de uso del negocio, consisten en la descripción de la secuencia de actividades que en conjunto producen algo observable para el actor del negocio. La estructura del mismo se describe gráficamente con la ayuda de un diagrama de actividad.

Diagramas de actividad del Negocio. Módulo de Gestión de Clientes y Facturación de Servicios Técnicos

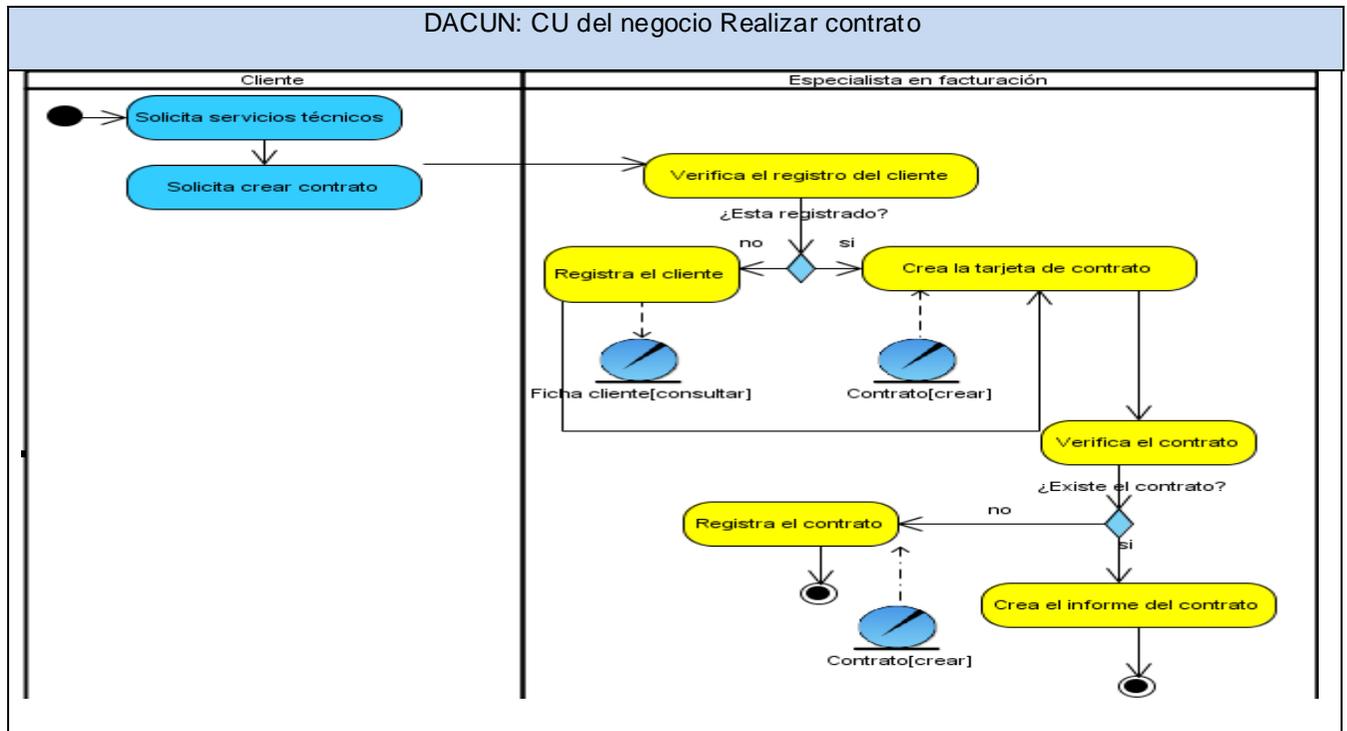


Figura #2: DACUN: Módulo de Gestión de Clientes y Facturación de Servicios Técnicos .Realizar contrato.

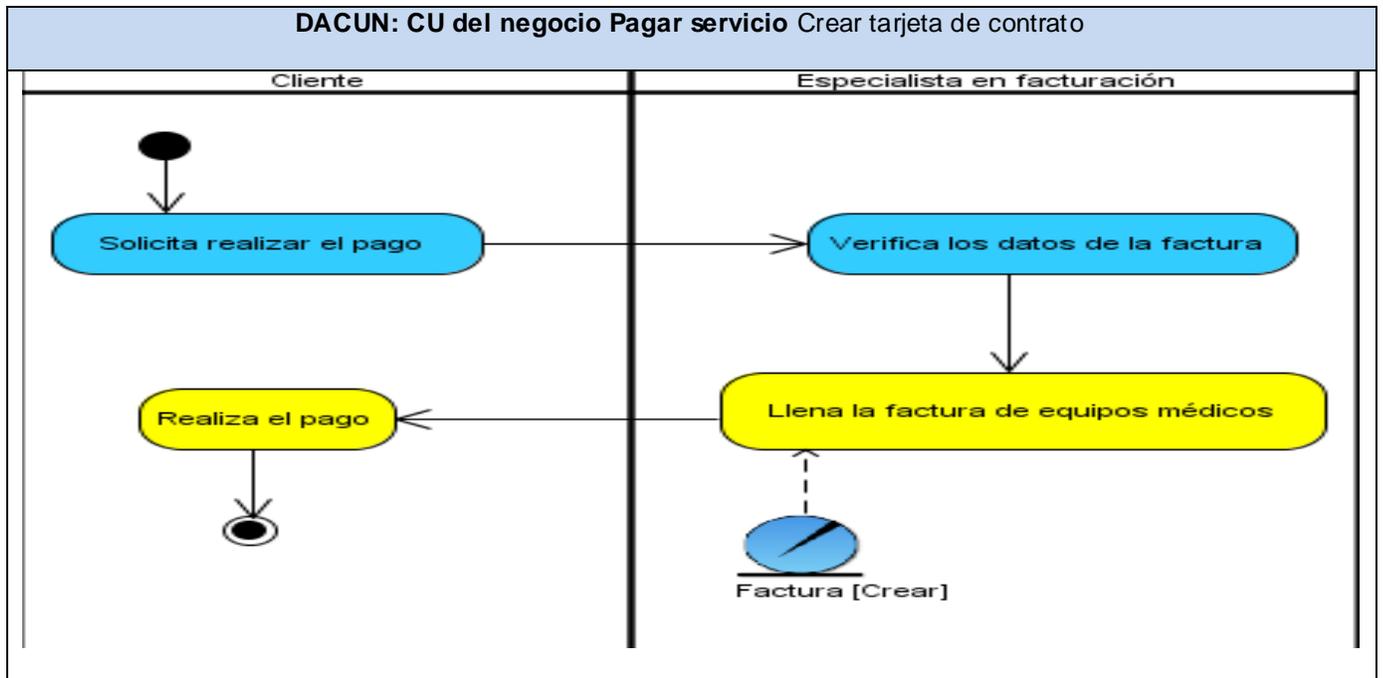


Figura # 3 DACUN: Módulo de Gestión de Clientes y Facturación de Servicios Técnicos. Pagar servicio.

2.3.6 Modelo de objetos

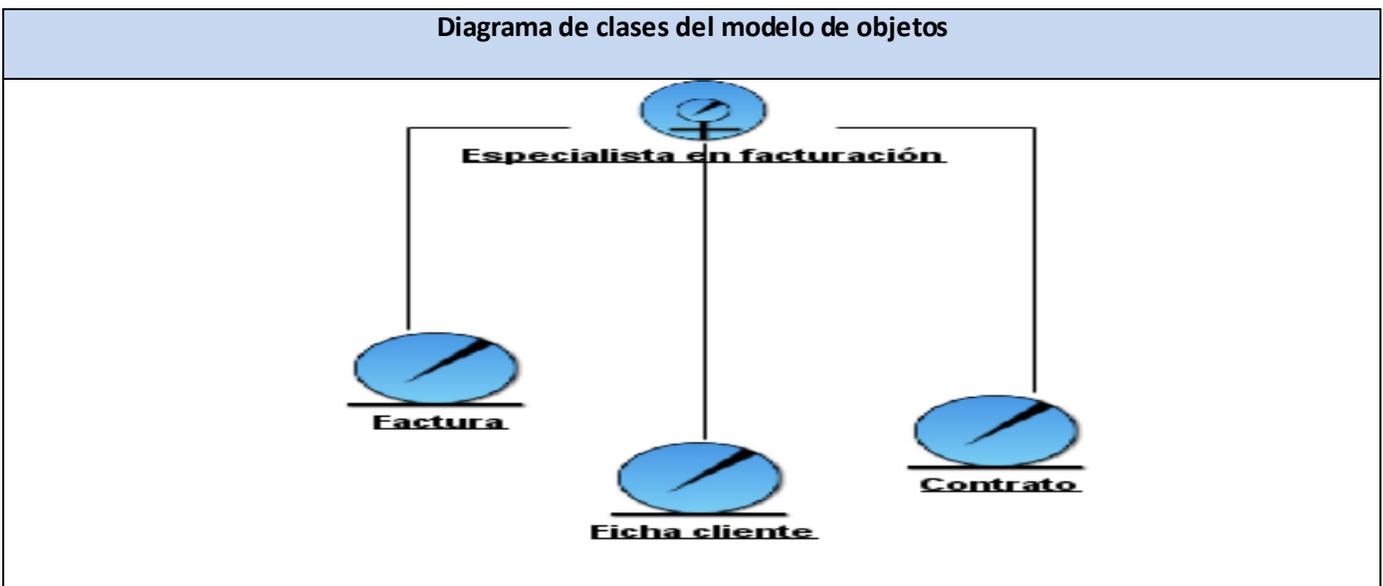


Figura # 4: Modelo de objetos. Módulo Gestión de Clientes y Facturación de Servicios Técnicos.



2.4 Especificación de los requisitos de software

2.4.1 Requisitos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Se mantienen invariables sin importar con qué propiedades o cualidades se relacionan. Para satisfacer a los clientes, el sistema debe cumplir con los siguientes requisitos funcionales definidos:

Requisitos Funcionales	
<p>RF 1 Crear una factura.</p> <p>RF 2 Listar facturas.</p> <p>RF 3 Mostrar factura ampliada.</p> <p>RF 4 Actualizar factura.</p> <p>RF 5 Crear ficha cliente.</p> <p>RF6 Listar ficha cliente.</p> <p>RF 7 Actualizar la ficha cliente.</p> <p>RF 8 Eliminar cliente.</p> <p>RF 9 Listar costos.</p> <p>RF 10 Adicionar costo.</p> <p>RF 11 Eliminar costo.</p> <p>RF 12 Modificar costo</p> <p>RF 13 Listar tipos de trabajo.</p> <p>RF 14 Adicionar el tipo de trabajo.</p> <p>RF 15 Modificar tipo de trabajo.</p>	<p>RF 16 Eliminar tipos de trabajo.</p> <p>RF 17 Listar el tipo de pago.</p> <p>RF 18 Adicionar el tipo de pago.</p> <p>RF 19 Modificar el tipo de pago.</p> <p>RF 20 Eliminar el tipo de pago.</p> <p>RF21 Exportar resultados en formato PDF.</p> <p>RF 22 Listar tipos de moneda.</p> <p>RF 23 Adicionar el tipo de moneda.</p> <p>RF 24 Modificar tipo de moneda.</p> <p>RF 25 Eliminar tipos de moneda.</p> <p>RF 26 Listar tipos de cliente.</p> <p>RF 27 Adicionar el tipo de cliente.</p> <p>RF 28 Modificar tipo de cliente.</p> <p>RF 29 Eliminar tipos de cliente.</p>

2.4.2 Requisitos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades se ven como las características que hacen al producto atractivo, rápido o confiable. A continuación se especifican los requisitos no funcionales que el sistema debe cumplir por categorías.

Categoría	Requisitos no funcionales del sistema
Usabilidad	El subsistema tiene que ser capaz de ofrecer facilidades de uso para un buen entendimiento y aceptación del producto por los usuarios finales. Debe ser sencillo a la vista de los usuarios, con el objetivo de implementar una herramienta que gane aceptación entre sus usuarios, se hace



	necesario el uso de un menú que le facilite, a los mismos el trabajo con el sistema.
Apariencia o interfaz externa	El subsistema será diseñado con una interfaz amigable, de fácil navegación por parte del usuario. No contiene numerosas imágenes para evitar demoras en la respuesta a cualquier acción del usuario. Similar a la interfaz de Windows por su estructura y facilidad de uso, pues la mayoría de los usuarios están adaptados al sistema operativo Windows, para garantizar que el personal que trabaje con el subsistema esté lo más familiarizado posible de manera que agilice y facilite el trabajo con el software.
Rendimiento	El subsistema debe ser de rápida respuesta frente a una petición del usuario. Para lograr un buen rendimiento, debe tener un rápido procesamiento de los datos pues los usuarios estarán en constante intercambio de información. Está concebido para un ambiente cliente/servidor, el cual debe garantizar la rapidez de respuesta ante las solicitudes de los usuarios, así como la velocidad de procesamiento de la información; todo esto se logrará luego de una validación de los datos, manipulación de eventos en el cliente, y en el servidor, aquellas que por cuestiones de seguridad o acceso a los datos lo requieran.
Soporte	Una vez puesto en marcha el subsistema y usado por los usuarios finales, se recogerá toda la información referente a los defectos, para incorporar las mejoras sugeridas al mismo.
Portabilidad	El subsistema será multiplataforma lo que permitirá ejecutarse sobre diferentes sistemas operativos sin importar sus versiones, y sin necesidad de modificar su código fuente.
Seguridad	El subsistema contará con una política de seguridad diseñada en base a la restricción de usuarios en dependencia del nivel jerárquico que cumpla su rol dentro del mismo.
Contabilidad	El subsistema debe ser confiable y preciso en la información que le suministra al usuario para evitar cualquier tipo de error. Estará disponible todo el tiempo, permitiendo el trabajo a los usuarios y las acciones de mantenimiento. Este debe ser estable, fiable y la velocidad de respuesta debe ser rápida durante la utilización del mismo. La información almacenada debe ser confiable en cuanto a su veracidad e integridad desde su recopilación y durante toda su manipulación. Además la interfaz se realizará fundamentalmente, con campos de selección para garantizar la seguridad de la información contenida en la base de datos.
Político cultural	Se deberá hacer un uso correcto del idioma español en la Interfaz del subsistema.



Legales	El subsistema estará desarrollado en base a las políticas del software libre, que fueron ajustadas al Sistema Nacional de Salud como: uso de servidores GNU/Linux, gestor de base de datos MySQL, lenguaje del lado del servidor PHP, que son herramientas libres además del uso de Visual Paradigm, que no es libre pero la Universidad de las Ciencias Informáticas posee licencia para su uso.
Software	Se requiere del navegador Mozilla 3.6 para la visualización del subsistema. Se usará como gestor de bases de datos MySQL Server, ya que da la posibilidad que el sistema sea multiplataforma y que se ajuste a las políticas que plantea el Sistema Nacional de Salud en cuanto al uso del software libre para sus aplicaciones. Para su implementación se usará Symfony 1.4.
Hardware	<p>Requerimientos mínimos para el servidor:</p> <ul style="list-style-type: none"> ➤ Computadora Pentium a 2.8 GHz o superior. ➤ 512 MB RAM o superior. ➤ 40 GB de espacio libre en Disco Duro como mínimo. <p>Requerimientos mínimos para la conexión del cliente:</p> <ul style="list-style-type: none"> ➤ Computadora Pentium a 333 MHz o superior. ➤ 128 MB RAM o superior. ➤ MODEM o red con TCP-IP para conexión al servidor.

3.1 Actor del sistema

Los actores del sistema son personas que interactúan directamente con las funcionalidades implementadas en el sistema, ven las funcionalidades del mismo y describen cómo serán usadas, además puede participar en varios casos de usos. El actor que se describe a continuación es el relacionado con el sistema en el Módulo Gestión de Clientes y Facturación de Servicios Técnicos.

Actor	Descripción
Especialista en facturación.	Es la persona encargada de llevar a cabo todo el proceso de facturación de los pagos de servicios técnicos. También es el encargado de la realización de contratos al cliente que solicite la prestación de servicio.

Tabla # 5. Descripción de los actores del sistema.



3.2 Diagrama de Casos de uso del sistema

A continuación se muestra el diagrama de casos de uso del sistema.

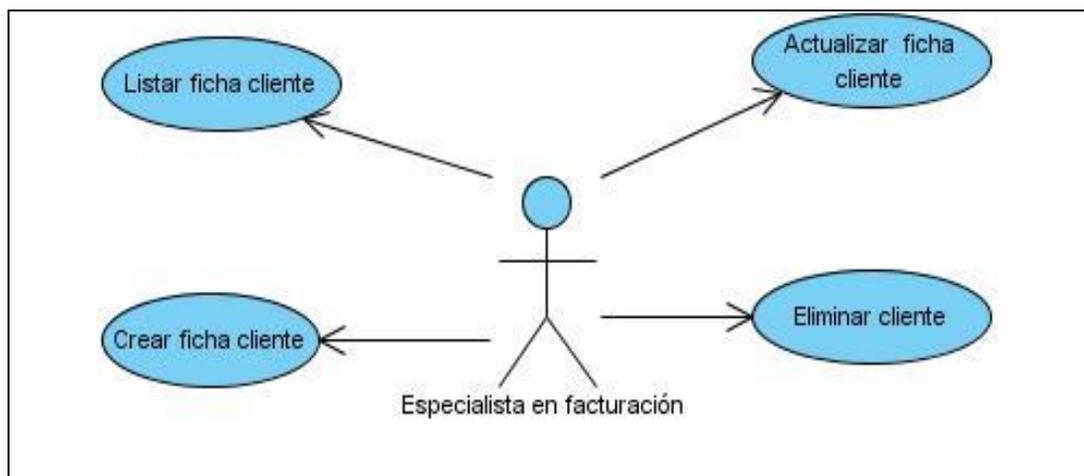
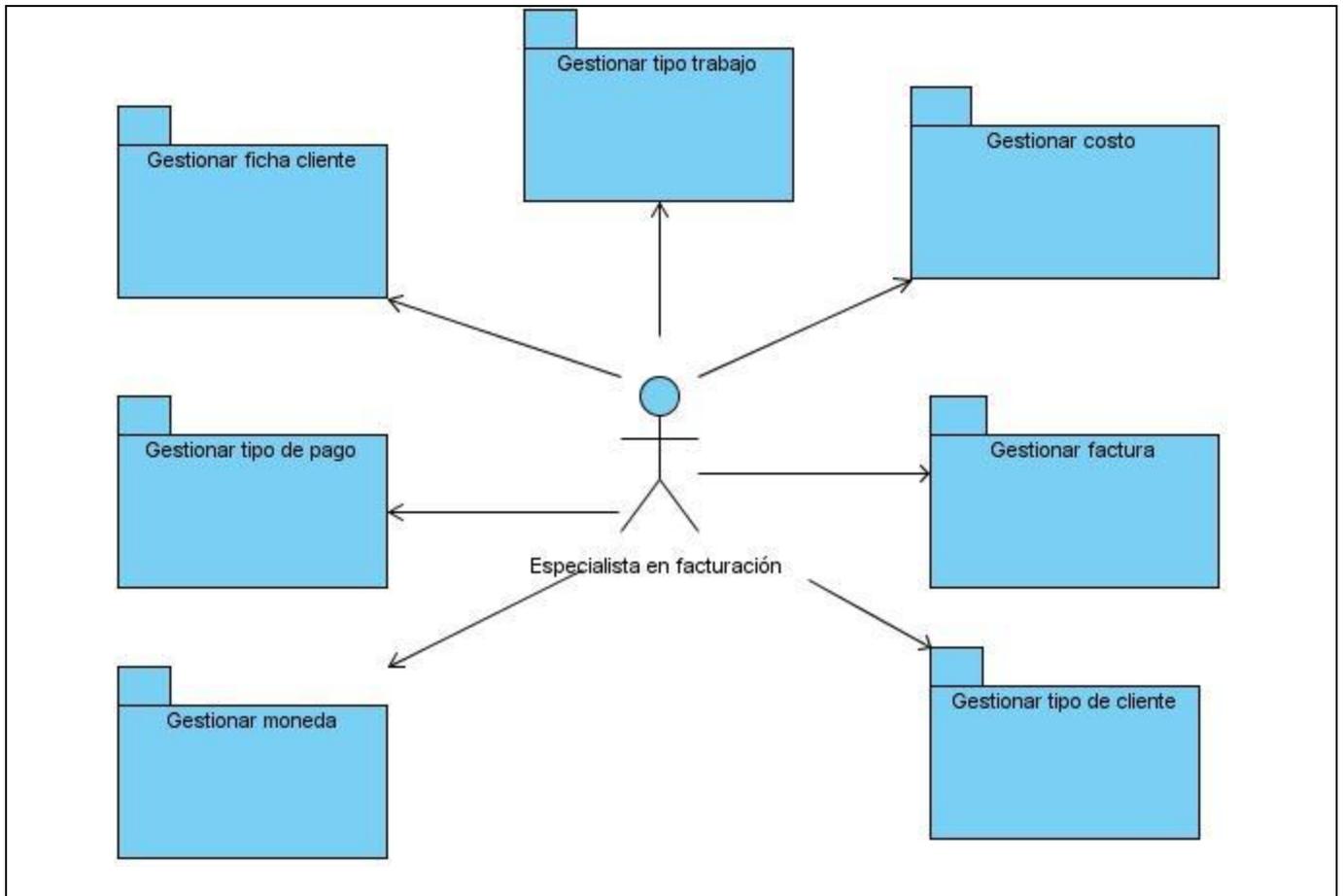


Figura # 5: Diagrama de caso de uso del sistema. Gestionar Ficha cliente.

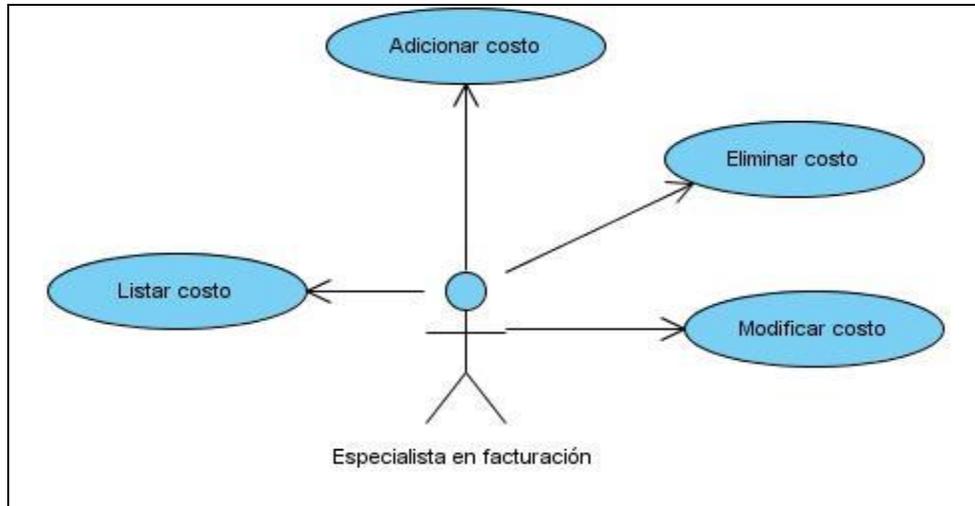


Figura # 6: Diagrama de caso de uso del sistema. Gestionar costo.

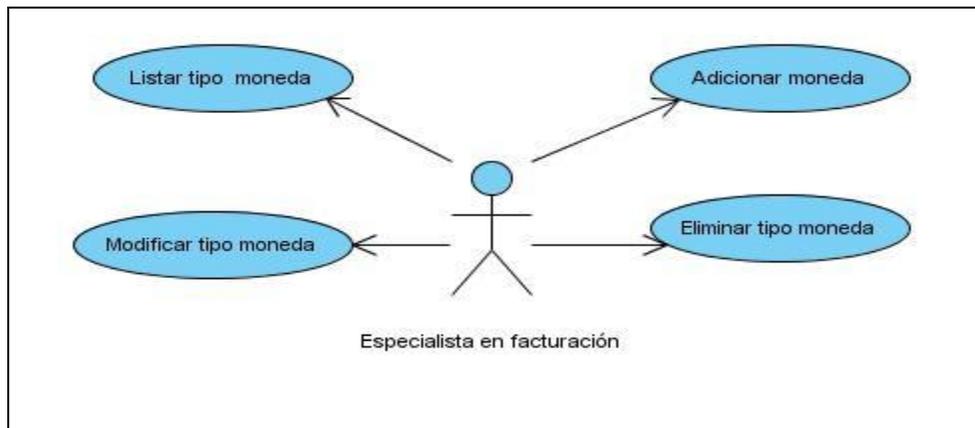


Figura # 7: Diagrama de caso de uso del sistema. Gestionar moneda.

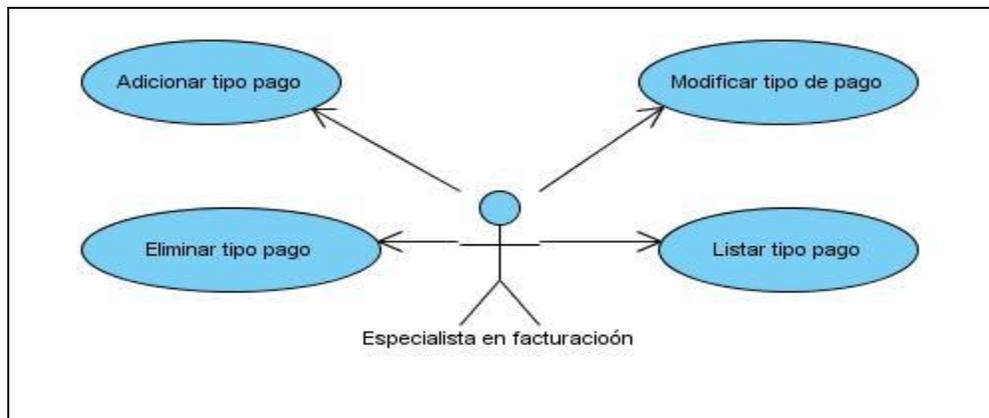


Figura # 8: Diagrama de caso de uso del sistema. Gestionar tipo de pago.

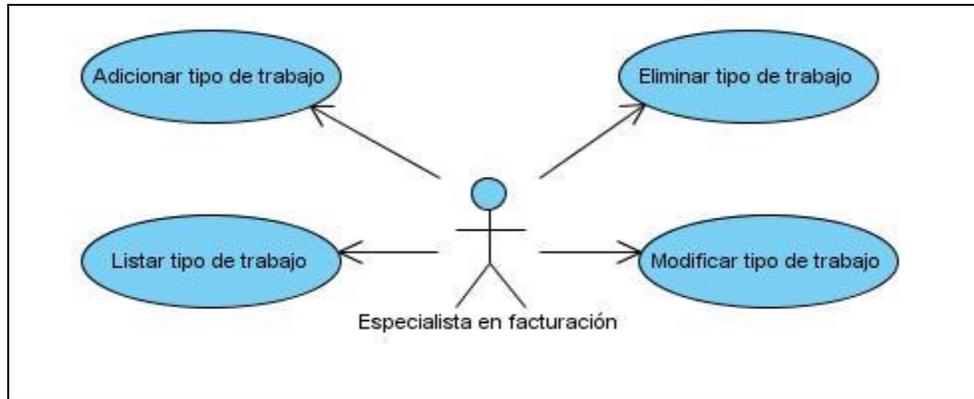


Figura # 9: Diagrama de caso de uso del sistema. Gestionar tipo de trabajo.

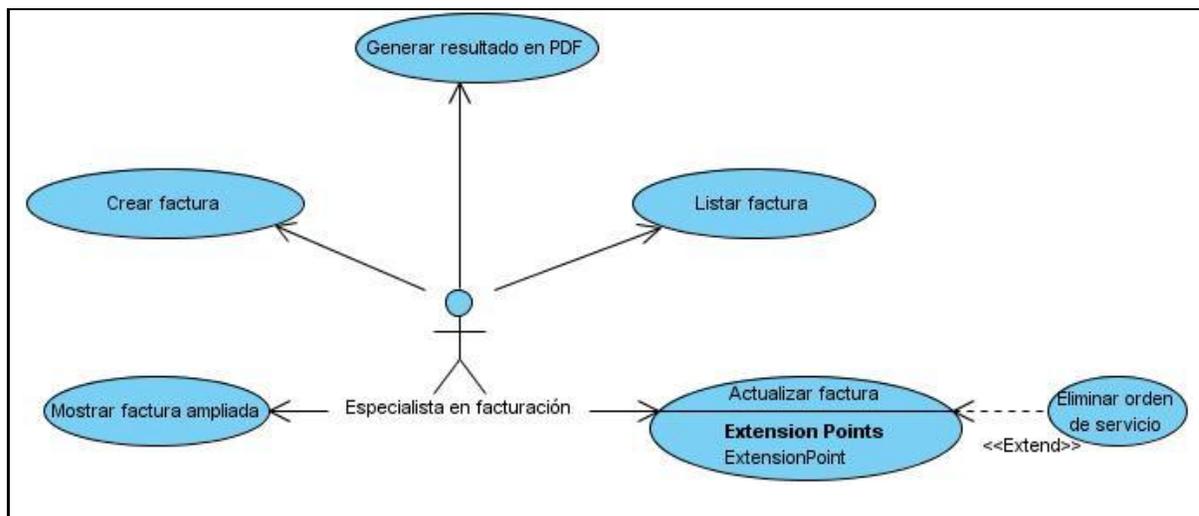


Figura # 10: Diagrama de caso de uso del sistema. Gestionar factura.

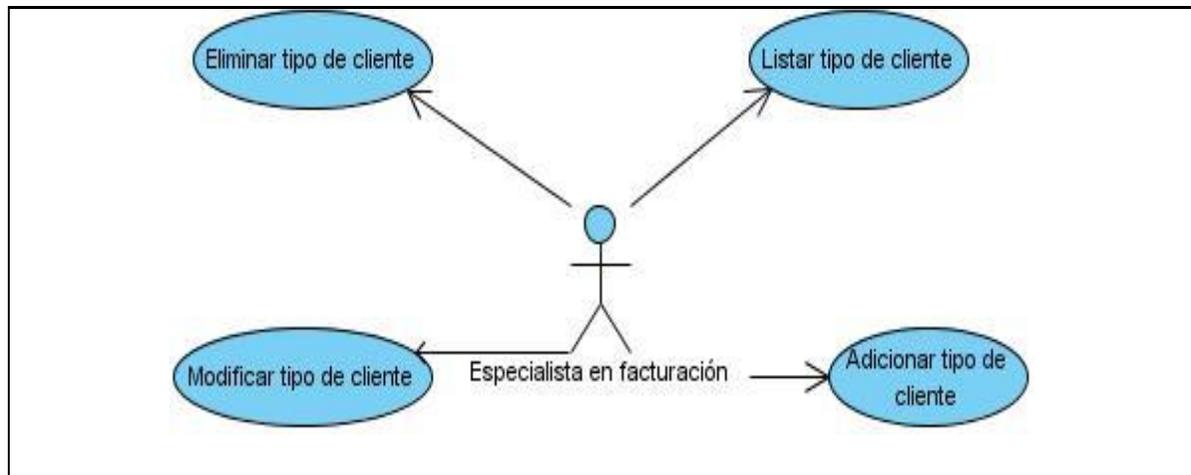


Figura # 11: Diagrama de caso de uso del sistema. Gestionar tipo de cliente.



3.3 Descripción del Caso de uso Crear factura

Caso de Uso:	Crear factura
Actores:	Especialista en facturación
Resumen:	El caso de uso inicia una vez que el Especialista en facturación accede a la opción Crear factura, el sistema brinda la posibilidad de introducir o seleccionar los datos para crear la factura, el Especialista en facturación introduce los datos de la factura, el sistema permite seleccionar la orden de servicio, y seleccionar la unidad, concluye así el caso de use.

Tabla # 6. Descripción del Caso de Uso Crear factura.

3.4 Descripción del Caso de Uso Crear ficha cliente

Caso de Uso:	Crear ficha cliente
Actores:	Especialista en facturación
Resumen:	El caso de uso inicia una vez que el Especialista en Facturación accede a la opción Crear Ficha Cliente el sistema brinda la posibilidad de introducir y seleccionar los datos para crear la ficha cliente, el Especialista en Facturación introduce los datos de la ficha cliente, el sistema crea la ficha cliente, concluye así el caso de uso.

Tabla # 7. Descripción del Caso de Uso Crear Ficha Cliente.

Con el desarrollo de este capítulo se obtuvo una mejor comprensión del problema en cuestión, mediante el flujo de trabajo modelamiento del negocio y requerimientos. En el modelamiento se identifican los actores, trabajadores y casos de uso del negocio, los cuales fueron descritos detalladamente. Además, se definen las reglas del negocio, se construye el diagrama de casos de uso del negocio así como, los diagramas de actividades y modelo de objeto correspondientes a los casos de uso. De forma detallada en el flujo de trabajo requerimiento se identifican y se describen los requisitos funcionales y no funcionales correspondientes, actores y casos de usos. Se realiza el diagrama de casos de uso del sistema, mostrando la relación entre casos de uso y actores, así como la descripción de los mismos, de esta manera se obtendrá una visión entendible de las funcionalidades propuestas en los requerimientos definidos.



CAPÍTULO 3: Análisis y Diseño del Sistema

El presente capítulo se centra fundamentalmente en el flujo de trabajo análisis y diseño del sistema. Se detallan los casos de usos identificados, y por cada realización de los mismos se muestran los diagramas de clases del diseño así como, los diagramas de análisis, comunicación y diagrama de despliegue correspondientes a utilizar en la implementación del Módulo Gestión de Clientes y Facturación de Servicios Técnicos.

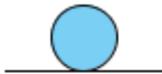
3.1 Modelo de Análisis

El análisis consiste en obtener una visión detallada del sistema, de modo que sólo se interesa por los requisitos funcionales, para refinarlos y estructurarlos, por lo que es considerado el principal eslabón para el comienzo de las actividades de diseño e implementación. Para describir un modelo de análisis se utiliza el lenguaje de los desarrolladores al razonar sobre los funcionamientos internos del sistema.

El objetivo del análisis, se centra en comprender perfectamente los requisitos del software y no precisar cómo se implementa el producto. En el modelo del análisis se refinan los requisitos, no se toma en cuenta el lenguaje de programación a usar en la construcción, la plataforma en la que se ejecuta la aplicación, entre otras características que afectan al sistema.

Diagrama de clase del análisis

El diagrama de clase del análisis se realiza para cada caso de uso del sistema, en el que se muestran las clases participantes en dichos CUS y las relaciones entre ellas. En los diagramas de clases se identifican tres tipos de clases: Interfaz, Controladora, Entidad. (Ver tabla# 8)

Representación	Nombre	Característica
	Entidad	Modelan información que posee larga vida y que es a menudo persistente, generalmente posee información específica de dicha entidad.
	Control	Se encarga de controlar la realización de uno o unos pocos casos de uso, ya que controla las actividades de los objetos que implementan la funcionalidad del caso de uso.



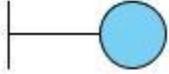
 <p>CI_WebDesktop</p>	<p>Interfaz</p>	<p>Se encarga de representar el acceso del usuario al sistema. Atiende todas las selecciones y peticiones que pueda realizar un determinado usuario.</p>
---	-----------------	--

Tabla # 8: Representación de los estereotipos a utilizar en el diagrama de clases del análisis.

Diagramas de clases del análisis.

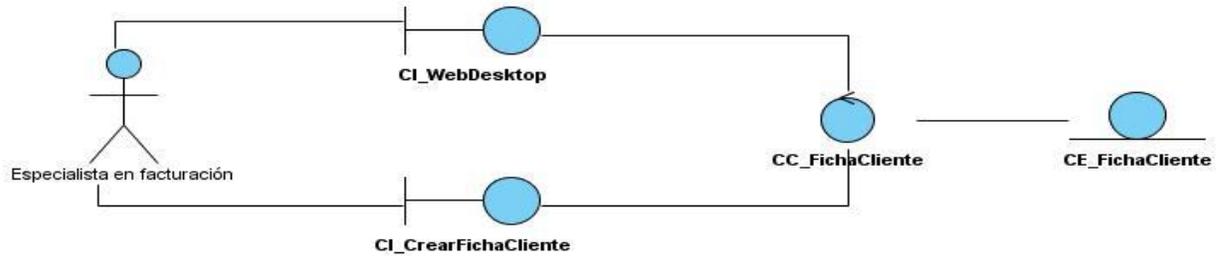


Figura # 12 Diagrama de clases del análisis. Crear Ficha Cliente.

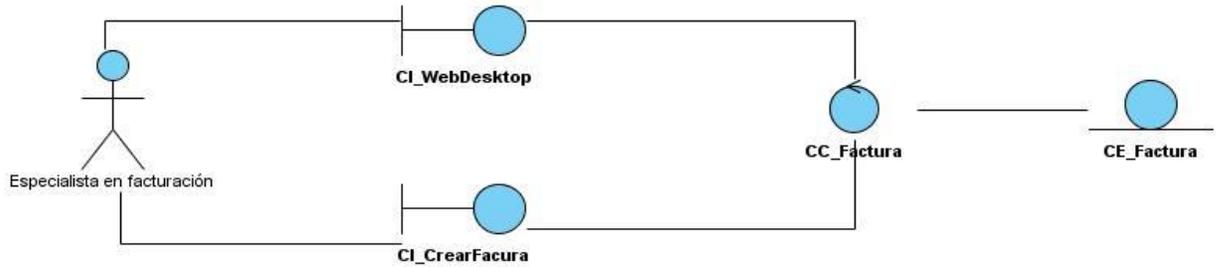


Figura # 13 Diagrama de clases del análisis. Crear factura.

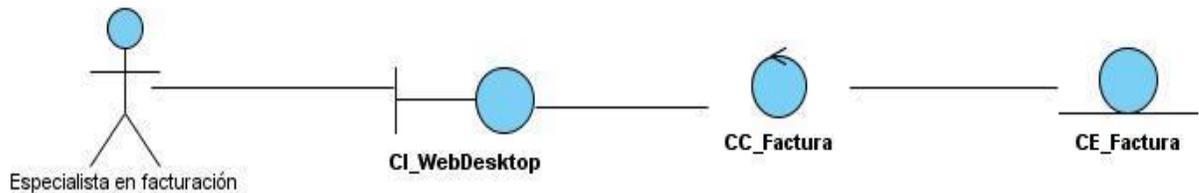


Figura # 14 Diagrama de clases del análisis. Listar factura.

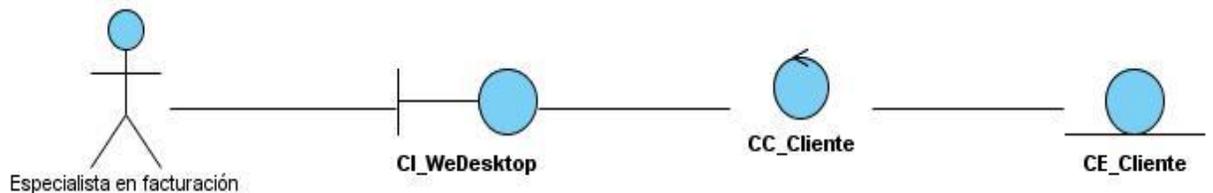




Figura # 15 Diagrama de clases del análisis. Listar cliente.

Diagramas de interacción. Comunicación

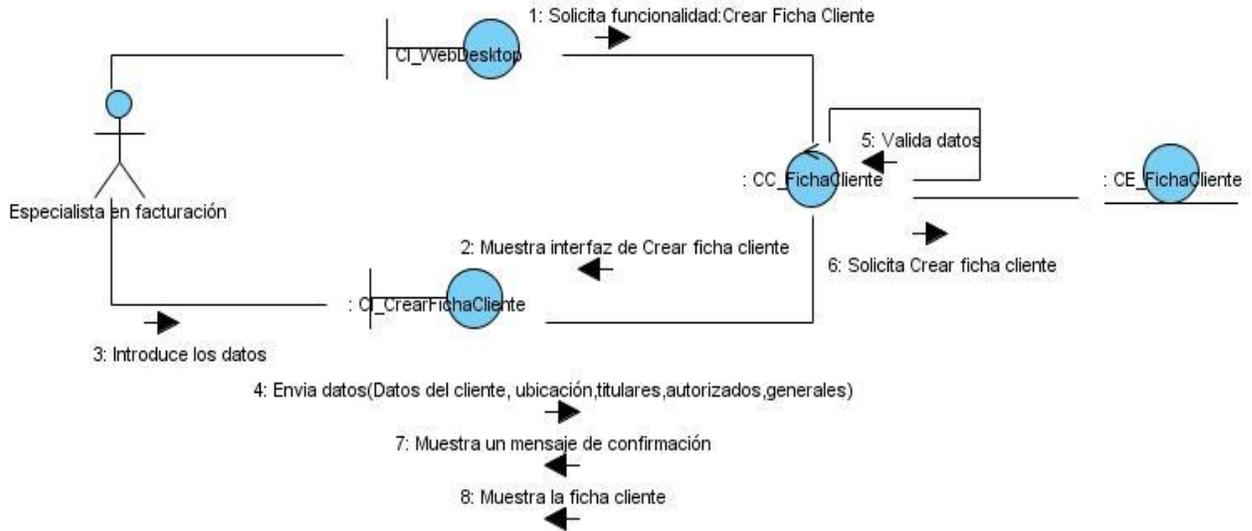


Figura # 16: Diagrama de comunicación. Crear ficha cliente.

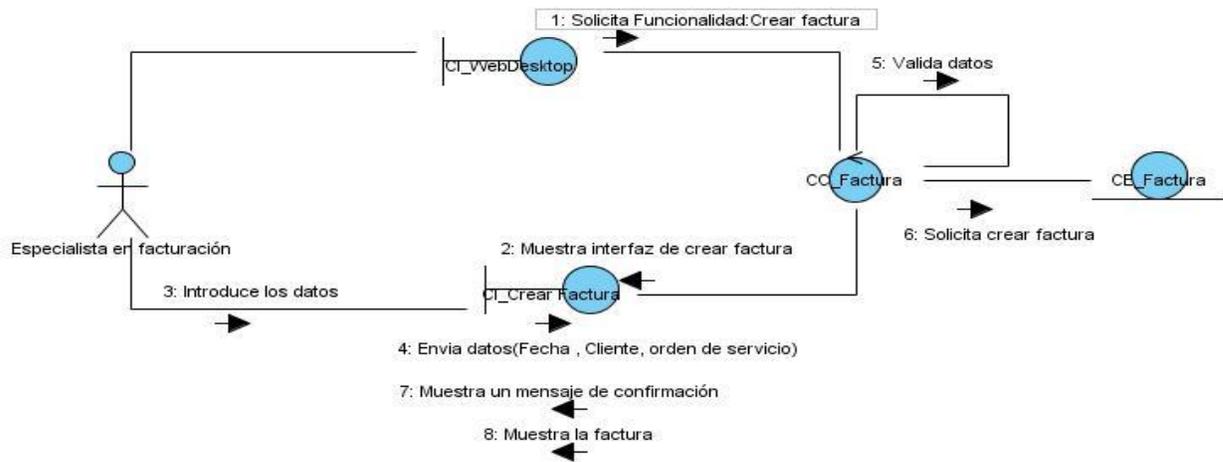


Figura # 17: Diagrama de comunicación. Crear factura.



Figura # 18: Diagrama de comunicación. Listar factura.



Figura # 19: Diagrama de comunicación. Listar cliente.

3.2 Diseño

El propósito del diseño es modelar el sistema y encontrar la forma para que soporte todos los requisitos definidos. Se plantean todos los aspectos relacionados con las restricciones y características del sistema como son; el lenguaje de programación a utilizar, el sistema operativo y las tecnologías de interfaz de usuario. El diseño tiene en cuenta los requisitos funcionales y no funcionales para obtener una visión detallada de la implementación futura para que el sistema sea realizado sin errores.

Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los CUS, centrándose específicamente en los requisitos funcionales y no funcionales. Además, es utilizado para concebir y documentar el diseño del sistema de software pues abarca todas las clases del diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos.

Para el desarrollo del modelo de diseño es necesario establecer una descomposición del mismo en subsistemas, con sus interfaces y las dependencias. Esta representación es de gran importancia para la arquitectura, los subsistemas y sus interfaces constituyen la estructura fundamental del producto de software. Se utilizó el patrón de arquitectura Modelo Vista Controlador (MVC), que separa los datos de la interfaz de usuario, y la lógica del negocio, conformándose de la siguiente manera (Ver figura #20).

El modelo: Conjunto de clases que representan la información del mundo real que el sistema debe procesar, sin tomar en cuenta ni la forma en la que esa información va a ser mostrada ni los mecanismos que hacen que esos datos estén dentro del modelo, es decir, sin tener relación con ninguna otra entidad dentro de la aplicación. (28)

La vista: Conjunto de clases que se encargan de mostrar al usuario la información contenida en el modelo. Una vista está asociada a un modelo, pudiendo existir varias vistas asociadas al mismo modelo.



Una vista obtiene del modelo solamente la información que necesita para desplegar y se actualiza cada vez que el modelo del dominio cambia por medio de notificaciones generadas por el modelo de la aplicación. Las vistas son responsables de recibir los datos del modelo y mostrarlos al usuario y tener un registro de su controlador asociado. (29)

El controlador: El controlador es un objeto que se encarga de dirigir el flujo del control de la aplicación debido a mensajes externos, como datos introducidos por el usuario u opciones del menú seleccionadas por él. A partir de estos mensajes, el controlador se encarga de modificar el modelo o de abrir y cerrar vistas. El controlador tiene acceso al modelo y a las vistas, pero las vistas y el modelo no conocen de la existencia del controlador. Es el responsable de recibir los eventos de entrada y contener las reglas de gestión de eventos. (30)

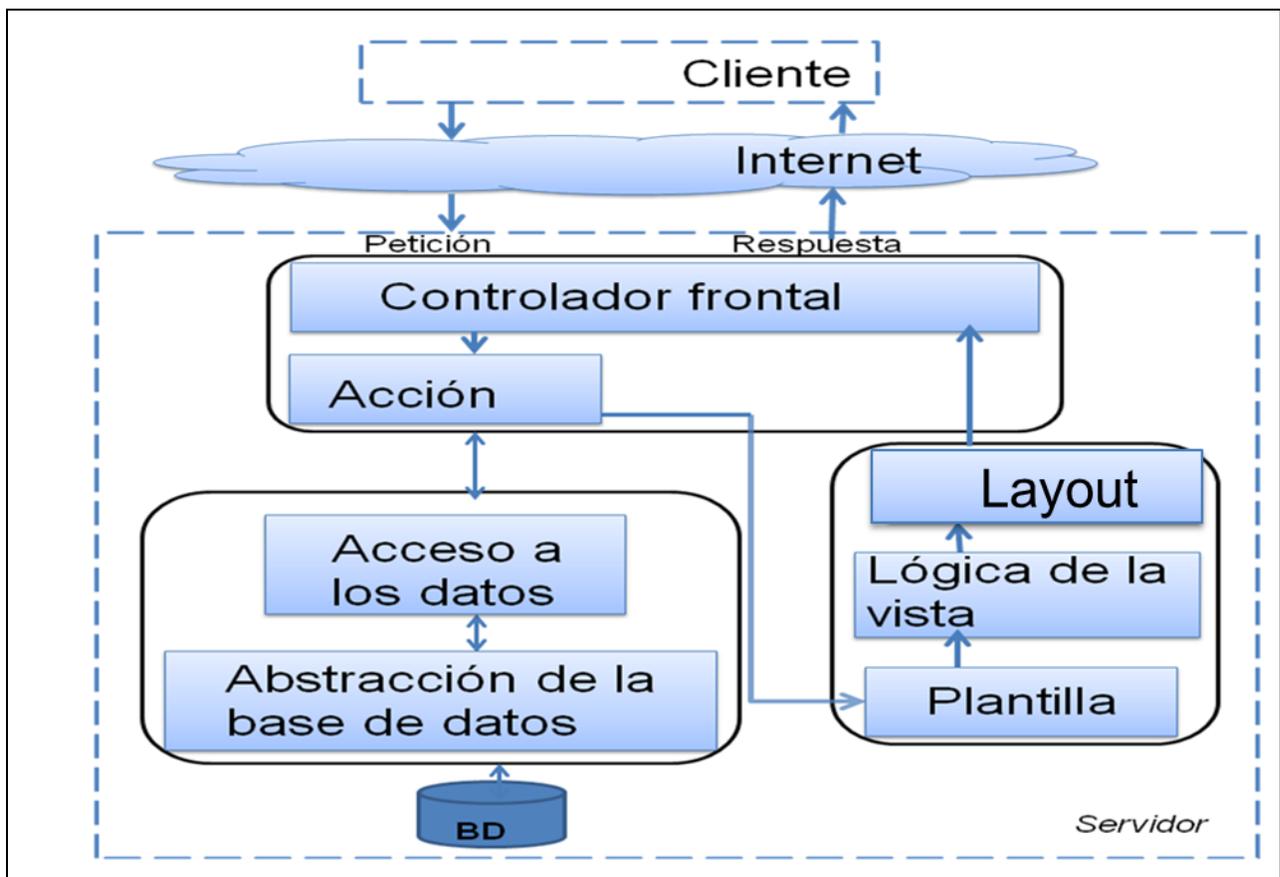


Figura #20. Modelo Vista Controlador.

La arquitectura descrita está conformada por tres capas fundamentales, las cuales están compuestas por script. La vista se separa en un layout y en una plantilla. El layout es global en toda la aplicación o al menos en un grupo de páginas, mientras que la plantilla sólo se encarga de visualizar las variables definidas en el controlador. En el modelo se encuentra el acceso a los datos y la abstracción de la base de



datos. El controlador se encarga de manejar las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación y otras tareas similares. Por tal motivo este se divide en un controlador frontal, que es único para cada aplicación, y las acciones, que incluyen el código específico del controlador de cada página, ofreciendo un punto de entrada único para toda la aplicación.

3.3 Patrones de Diseño

Son una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Patrones de asignación de responsabilidades (GRASP)

- **Experto:** propone como solución asignar la responsabilidad a la clase que cuenta con la información necesaria para cumplir la responsabilidad. Las clases que brinda el framework ExtJS se encargarán de visualizar las interfaces pues cuentan con la información para crear los diferentes componentes visuales.
- **Creador:** este patrón se tiene en cuenta para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto solo pueda ser creada por el objeto que contiene la información necesaria. El uso del mismo permite crear las dependencias mínimas necesarias entre las clases, favoreciendo al mantenimiento del sistema.
- **Bajo Acoplamiento:** brinda como solución asignar responsabilidades de manera que las clases no dependan fuertemente de otras. Ofreciendo como beneficio que son fáciles de entender por separadas, fáciles de reutilizar y no se afectan por cambios de otros componentes. Dicho patrón se tiene en cuenta debido a la importancia de realizar un diseño de clases independientes que soporten los cambios.
- **Alta cohesión:** propone asignar la responsabilidad de manera que la complejidad se mantenga dentro de límites manejables asumiendo solamente las responsabilidades que deben manejar, evadiendo un trabajo excesivo. Su utilización mejora la claridad y facilidad para comprender el diseño, simplifica el mantenimiento y las mejoras de funcionalidad, genera un bajo acoplamiento, soporta mayor capacidad de reutilización.
- **Controlador:** es un patrón que se basa en usar un controlador como punto inicial para la gestión de las peticiones. La presencia del controlador frontal, es un ejemplo que evidencia de forma clara



su utilización en Symfony. Es el punto de entrada único de toda la aplicación ya que todas las peticiones realizadas por el usuario son gestionada por él, así como la seguridad, entre otras tareas.

- **Decorador:** es utilizado cuando deseamos modificar el comportamiento básico de una instancia específica sin la necesidad de crear una nueva subclase. Específicamente Symfony presenta el archivo `layout.php` o también conocido como plantilla global, éste contiene el código HTML común a todas las páginas de la aplicación para no tener que repetirlo en cada página. Este fichero se complementa con las plantillas, decorándolas y obteniéndose la interfaz final que será mostrada al usuario.

3.3.1 Diagrama de clases del diseño

Es un tipo de diagrama estático que describe la estructura de un sistema mediante la visualización de sus clases, atributos y las relaciones entre ellos. Además, representa la interacción de las clases de diseño y sus objetos en la realización de los casos de uso, de forma que de cada uno de ellos depende un diagrama de clases.

Modelado mediante Estereotipos Web

Para la realización de los diagramas de clases del diseño se presentan como elementos significativos tres clases UML fundamentales: *Server Page*, *Client Page* y *Form*, empleadas para el código servidor, código cliente y formularios respectivamente, permitiendo representar ficheros contenedores de sentencias script.

Descripción de los elementos del diseño

En la elaboración de los diagramas de clases del diseño se utilizarán los elementos siguientes:

- **Link:** Representa un apuntador desde una *client page* hacia una *client page* o *server page*.
- **Submit:** Esta relación siempre se da entre una *form* y una “*server page*”, por lo que, la *server page* procesa los datos que la “*form*” le envía mediante un submit.
- **Build:** Permite identificar cuales *server page* son responsables de la creación de una *client page*. Una “*server page*” puede crear varias *client page*, pero una *client page* sólo puede ser creada por una sola *server page*.
- **Redirect:** Es una relación unidireccional que indica que una página Web redirige hacia otra.
- **Server Page:** Representa una página web que tiene scripts ejecutados por el servidor. Estos scripts interactúan con los recursos que se encuentran al alcance del servidor. Sólo puede mantener relaciones con objetos que se encuentren en el servidor.



- **Client Page:** Representa páginas que son dibujadas por el navegador web y pueden ser una combinación de algún o algunos lenguajes.
- **Form:** Representa una colección de campos de entrada que forman parte con una página del lado cliente (Client Page). (31)

Para observar con detalle los diagramas de clases del diseño correspondiente al Módulo Gestión de Clientes y Facturación de Servicios Técnicos ([Ver Anexo # 1](#)).

Descripción de las Clases del Diseño

Nombre de la clase	Descripción
ClienteActions.	Contiene la lógica de la aplicación del módulo gestión de clientes, esta clase se encuentra compuesta por métodos especiales y acciones que a su vez utilizan el modelo y define las variables para la vista.
FacturaActions.	Contiene la lógica de las funcionalidades del proceso de facturación, la misma está compuesta por métodos y acciones que utilizan el modelo y definen las variables para la vista.
EquipoActions.	Contiene la lógica de los equipos, está compuesta por métodos y acciones que utilizan el modelo y definen las variables para la vista.
indexSuccess.php.	Es utilizada para definir la interfaz principal de la aplicación. Contiene el código de presentación, HTML, CSS y JavaScript común a todas las páginas de la aplicación para no tener que repetirlo en cada una. Se encuentra ubicada en la capa Vista de la arquitectura definida, MVC.
layout.php.	Contiene el código HTML común a todas las páginas de la aplicación para no tener que repetirlo en cada página. Esta es conocida como plantilla global. Se encuentra ubicada en la capa Vista de la arquitectura definida, MVC.
editSuccess.php.	Contiene el código de presentación, siendo el resultado de una acción a la que previamente está asociada por el nombre seguido del resultado. Se encuentra ubicada en la capa Vista de la arquitectura definida, MVC.
sigicem.php.	Es el punto de entrada único de toda la aplicación, denominada comúnmente como controlador frontal, todas las peticiones realizadas son gestionadas por él. Hace uso del sistema de enrutamiento para realizar la asociación del nombre de una acción y el nombre de un módulo mediante la URL definida por



	el usuario.
CP_WebDesktop.	Representa el acceso del usuario al sistema, esta se encarga de atender todas las selecciones y peticiones que pueda realizar un determinado usuario.
CP_Crear_Factura.	Representa el acceso del usuario a la interfaz de crear factura, esta se encarga de atender todas las peticiones y selecciones realizadas por un determinado usuario.
fr_Crear_Factura.	Representa la entrada y salida de datos en el sistema, a través de esta el usuario podrá introducir los datos necesarios para la creación de la Factura de servicios técnicos.
CP_Crear_matriz.	Representa el acceso del usuario a la interfaz de crear matriz, esta se encarga de atender todas las selecciones y peticiones que pueda realizar un determinado usuario.
fr_Crear_Matriz.	Representa la entrada y salida de datos en el sistema, a través de esta el usuario podrá introducir los datos necesarios para la creación de la matriz de costo.
t_Cliente.	Es la representación Orientada a Objetos (OO) de los registros de la base de datos, a través de la cual es posible el tratamiento en forma de objeto de las entidades. Se debe tener en cuenta que t_Cliente es una representación generalizada de cualquiera de las tablas de la base de datos del SIGICEM.
t_ClientePeer	Contiene los métodos estáticos para obtener registros de la base de datos. Se debe tener en cuenta que t_ClientePeer es una representación generalizada de cualquiera de las tablas de la base de datos del SIGICEM.

Tabla # 9: Descripción de las clases utilizadas en la realización de los diagramas de clases del diseño.

3.4 Modelo de Despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema. Para la estructura de la distribución física del subsistema se utilizó un nodo PC_Cliente que representa la computadora del usuario. Esta se conecta mediante el protocolo de Transferencia de Hipertexto (http, HyperText Transfer Protocol) al nodo Servidor Web. Desde el servidor se puede acceder al nodo Servidor de BD mediante el protocolo diseñado para facilitar la reutilización de código de bases de datos (TCP/IP). Además desde el nodo PC_Cliente se accede a la impresora mediante el puerto USB (Ver figura # 21).

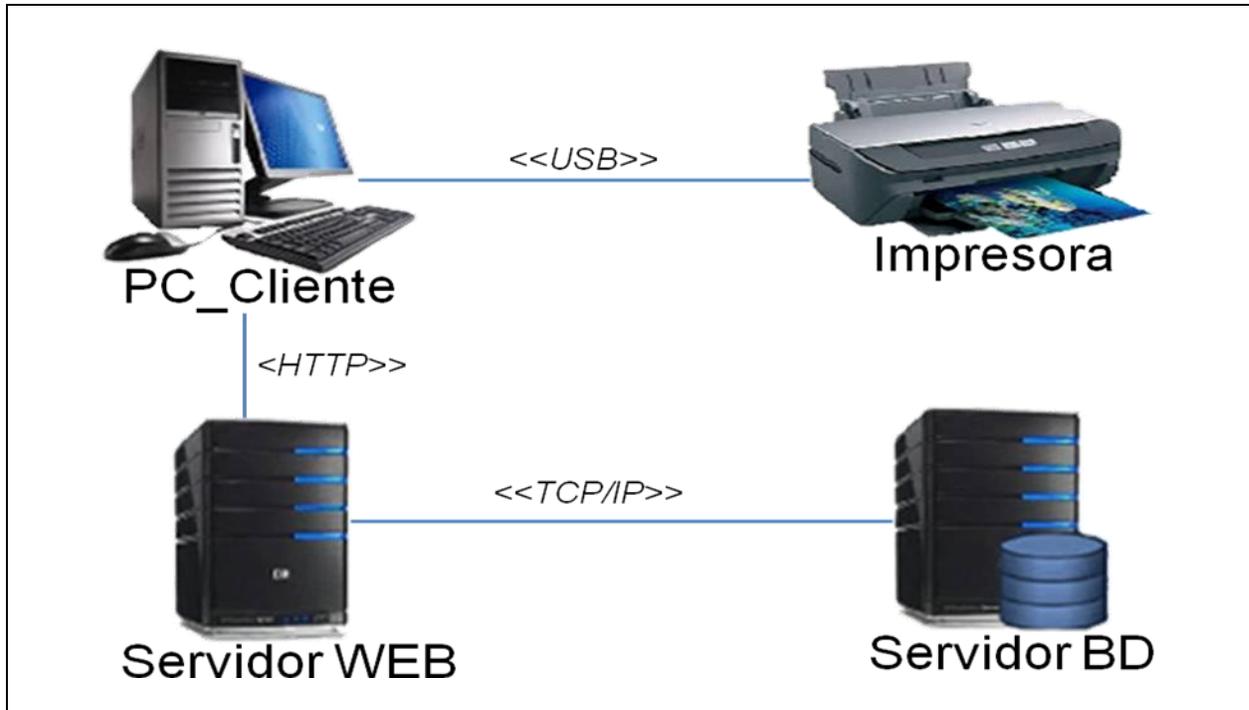


Figura # 21 Modelo de Despliegue.

Con el desarrollo de este capítulo se obtuvo el modelo de análisis correspondiente al módulo propuesto. Se realizaron diferentes artefactos mediante los cuales es posible adquirir una visión detallada de las responsabilidades de cada clase del análisis, para el correcto funcionamiento del módulo a desarrollar. Estos artefactos realizados fueron, diagrama de despliegue, diagramas de clases del análisis y la descripción de las clases. Además, se obtuvo el modelo de diseño, donde se realizaron los artefactos que definen la entrada a las actividades que se realizan en el flujo de trabajo de implementación, estos son; los diagramas de clases del diseño así como la descripción de las clases.



CAPÍTULO 4: Implementación

El presente capítulo constituye la secuencia lógica del diseño del sistema, a través del mismo se implementan las clases y subsistemas encontrados en el flujo de trabajo anterior en términos de componentes, se obtiene el diagrama de componentes correspondiente al Módulo Gestión de Clientes y Facturación de Servicios Técnicos.

4.1 Modelo de Implementación

El modelo de implementación describe los elementos del modelo del diseño, el mismo es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componente se encuentran datos, archivos, ejecutables, código fuente y los directorios. Principalmente permite describir la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos.

Diagrama de Componentes

Muestra las organizaciones y dependencias lógicas entre componentes. Además representa la separación de un sistema de software en componentes físicos (cabecera, paquetes, archivos, módulos). Estos componentes son utilizados para modelar la vista estática de un sistema.

Los diagramas de componentes muestran los componentes de software que constituyen una parte reusable, sus interfaces, y sus interrelaciones, en muchos aspectos se puede considerar que un diagrama de componentes es un diagrama de clases a gran escala. Cada componente en el diagrama debe ser documentado con un diagrama de componentes más detallado, un diagrama de clases o un diagrama de casos de uso. (32)

Estos son utilizados para modelar la vista estática de un sistema. Cada diagrama describe un apartado del sistema. En ellos se sitúan librerías, tablas, archivos, ejecutables y documentos que forman parte del sistema. Uno de los usos principales es que puede servir para ver que componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema. (33) **Diagrama de componentes del Módulo Gestión de Clientes y Facturación de Servicios Técnicos.** ([Ver Anexo 2](#))

4.2 Diseño de la Base de Datos

En el desarrollo de un sistema informático, el diseño de la Base de Datos (BD) es de suma importancia, debido que permite almacenar todos los datos que son necesarios en la modelación del problema que se desea resolver, además es la fuente de obtención de toda la información que se quiera recuperar del sistema. Las BD necesitan de una definición de su estructura que le permitan almacenar datos, así como recuperar la información deseada.



Modelo de datos

Representa todas las tablas de la BD utilizadas en la implementación del subsistema, permite describir los datos contenidos en las mismas y las relaciones que existen entre cada tabla a utilizar. Para obtener mayor información acerca de las tablas del modelo de datos utilizadas en la realización del Módulo Gestión de Clientes y Facturación de Servicios Técnicos se recomienda consultar el Expediente del Proyecto SIGICEM del Departamento Sistema de Apoyo a la Salud (SAS).

http://10.36.7.202:5800/sas_pm/Electromedicina/SIGICEM/Expediente%20de%20proyecto/

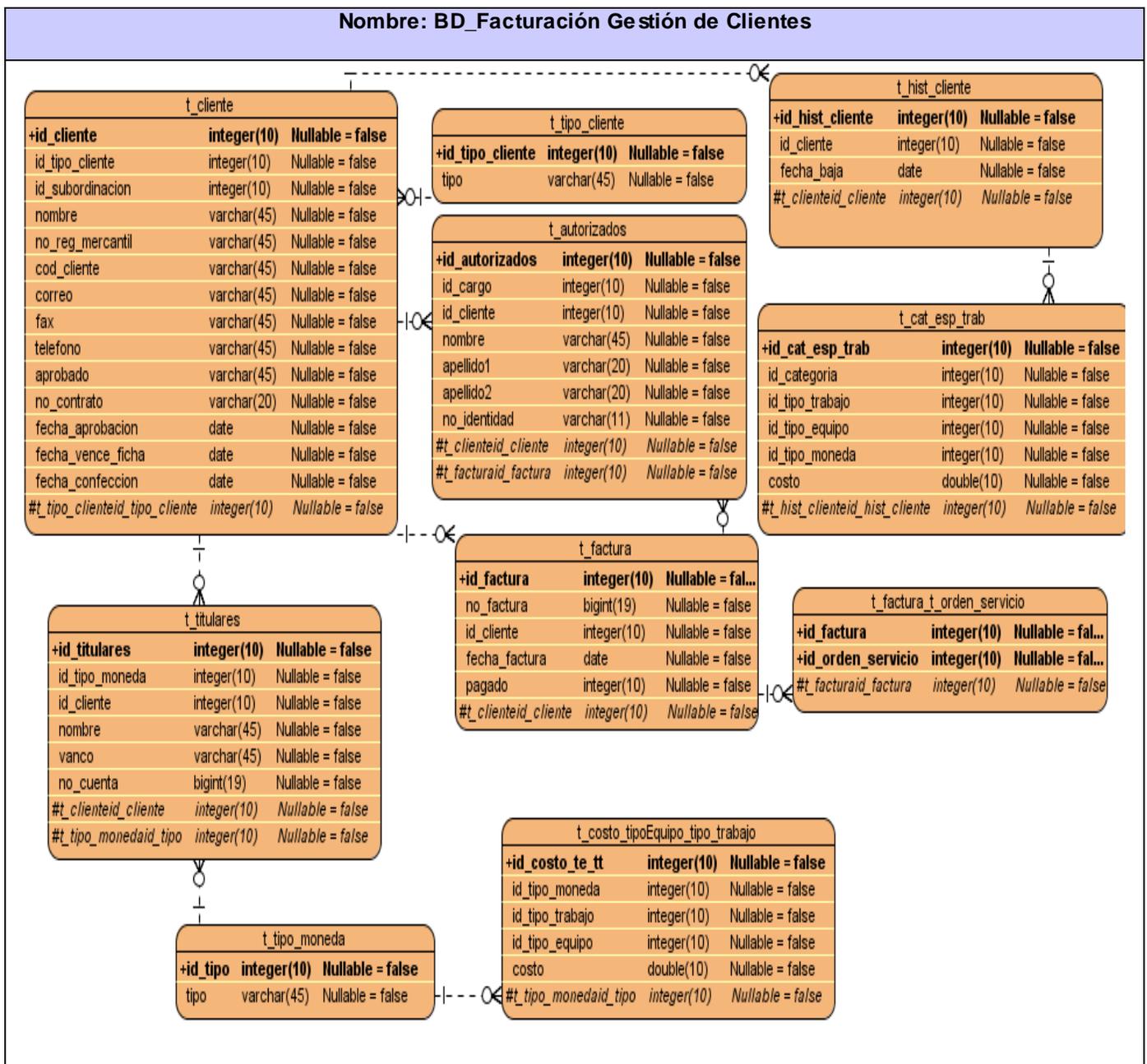


Figura # 22 Modelo de Datos.



4.3 Tratamiento de errores

El tratamiento de errores es un paso indispensable para el buen funcionamiento de cualquier sistema. Los errores más comunes que se pueden presentar en el funcionamiento del subsistema implementado, están relacionados fundamentalmente con el trabajo de la BD, esencialmente los procesos de eliminar, actualizar, e insertar datos. Se deben establecer mecanismos que visualicen la información, para evitar la introducción de los datos de forma manual y minimizar los errores que puede presentar el funcionamiento del subsistema. Para minimizar estos errores se cuenta con cuadros de opción y menú de selección que facilita la entrada de datos por parte del usuario al subsistema.

A continuación se visualizan a través de imágenes ejemplos de tratamientos de errores del Módulo Gestión de Clientes y Facturación de Servicios Técnicos.

The screenshot shows a window titled "Insertar cliente" with several tabs: "Datos del Cliente", "Unidades", "Titulares", and "Autorizados". The "Datos del Cliente" tab is active. The form contains the following fields:

- Tipo de cliente: Seleccione... (dropdown menu, highlighted with a red box)
- No.Reg Mercantil: [text input]
- Nombre cliente: [text input]
- Aprobado:
- Aprobación: [text input]
- Vence : [text input]
- Correo Oficial: [text input]
- Cod.Cliente: [text input]
- Sub.Cliente: Seleccione... (dropdown menu, highlighted with a red box)
- Fax: [text input]
- Confeción: [text input]
- No.Contrato: [text input]

Below the main form, there are two sections:

- Ubicación:** This section is expanded and contains five dropdown menus, all highlighted with red boxes:
 - País: Seleccione...
 - Región: Seleccione...
 - Provincia: Seleccione...
 - Municipio: Seleccione...
 - Localidad: Seleccione...
- Lista de teléfonos:** This section contains a table with a header "Teléfonos" and a sub-header "Teléfonos". It has a "+" icon and a trash icon. Below the table, there are buttons for "Teléfonos" and "Eliminar".

At the bottom of the window, there is a "Guardar" button.

Figura #23 Ejemplo de cuadro de selección.

4.4 Seguridad

Actualmente en el mundo, la información constituye un eslabón de vital importancia, por lo que es necesario que existan mecanismos que garanticen la protección de los datos que se manipulan en la



realización del subsistema, así como los permisos a los usuarios en dependencia del nivel jerárquico que cumple su rol dentro del mismo.

La seguridad del subsistema será responsabilidad del sistema en general (SIGICEM) el cual implementará una jerarquía de accesos para los diferentes usuarios del sistema porque todos no tendrán los mismos privilegios sobre las posibles acciones a realizar.

Con la realización de este capítulo se completó el ciclo de vida del software con la etapa de Implementación. Se realizó el diseño del diagrama de componentes, así como, las relaciones que existen entre los mismos, las que indican el grado de interrelación que existe entre ellos, a través de los diagramas de componentes se logra dividir el sistema en partes más entendibles por los programadores. Con esto se obtiene el módulo con todas las funcionalidades definidas en la especificación de requisitos y que satisface las necesidades de los clientes. Por lo que es significativo, mantener el modelo de implementación a lo largo de todo el ciclo de vida del software.



Conclusiones

Al culminar el presente trabajo de diploma, se ha cumplido con el objetivo general y las tareas propuestas, por lo que se han obtenido los siguientes resultados:

- Luego de realizar un estudio de los sistemas de gestión de clientes y facturación a nivel internacional y nacional, se evidenció que los mismos no son capaces de satisfacer las necesidades del Centro de Ingeniería Clínica y Electromedicina (CICEM).
- Mediante la confección de los artefactos pertenecientes a las Disciplinas propuestas por el Proceso Unificado de Desarrollo (RUP), se logró establecer la documentación necesaria para la realización de la investigación.
- Se desarrolló el Módulo Gestión de Clientes y Facturación, basado en las tecnologías y arquitectura definidas por el Centro de Informática Médica y el Departamento Sistema de Apoyo a la Salud (SAS).



Recomendaciones

A partir del estudio realizado en la presente investigación y teniendo en cuenta las ideas que surgieron en la etapa de análisis, diseño e implementación del módulo, se recomienda:

- Integrar el Sistema de Gestión para la Ingeniería Clínica y Electromedicina con un sistema contable que posea como funcionalidades: cobro y pago de los servicios ofrecidos a los clientes .



Referencias Bibliográficas

1. Tecnología. [En línea] [Citado el: 20 de enero de 2011.] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1729-519X2009000300022.
2. **Rolando Alfredo Hernández León.** *El paradigma cuantitativo de la investigación científica.* Cuba : publicación, 2002.
3. *Idem2.*
4. Gestión de clientes. [En línea] [Citado el: 25 de Octubre de 2010.] <http://apuntes-negocios.blogspot.com/2010/05/24-gestion-de-relaciones-con-el-cliente.html>.
5. Gestión de cliente. [En línea] [Citado el: 10 de Noviembre de 2010.] <http://apuntes-negocios.blogspot.com/2010/05/24-gestion-de-relaciones-con-el-cliente.html>.
6. Factura. [En línea] [Citado el: 16 de Noviembre de 2010.] <http://www.aigclassic.com/articulos/que-es-una-factura>
7. Molecule-CRM. [En línea] [Citado el: 18 de Noviembre de 2010.] <http://gratis.portalprogramas.com/Molecule-CRM.html>.
8. Express-Invoice. [En línea] [Citado el: 24 de Noviembre de 2010.] <http://gratis.portalprogramas.com/Express-Invoice.html>.
9. Pagares-y-cobros-Pro. [En línea] [Citado el: 24 de Noviembre de 2010.] <http://gratis.portalprogramas.com/Pagares-y-cobros-Pro.html>.
10. SeniorFactu. [En línea] [Citado el: 24 de Noviembre de 2010.] <http://gratis.portalprogramas.com/SeniorFactu.html>.
11. FactuSol. [En línea] [Citado el: 24 de Noviembre de 2010.] <http://gratis.portalprogramas.com/FactuSol.html>.
12. Vindsor. [En línea] [Citado el: 24 de 10 de 2010.] <http://www.desoft.cu/Productos1/Vindsor/tabid/437/Default.aspx>
13. **Unidad Empresarial de Base División de Sistemas Gerenciales, CITMATEL-2005.** *Sistema Integral de Gestión Económica Administrativa.* Cuba : Publicación, 2011.
14. *Plantilla Módulo Facturación 2008.* Cuba : Publicación, 2010.
15. RUP. [En línea] [Citado el: 24 de Noviembre de 2010.] <http://yaqui.mxl.uabc.mx/~molquin/as/RUP.htm>.



16. Visual parading. [En línea] [Citado el: 24 de Noviembre de 2010.] <http://www.visual-paradigm.com/product/vpum/>.
17. **Nelismi Barros (Tesi)**. *SIGICEM: Módulo de Gestión Tecnológica*. Cuba : Publicación, 2009.
18. MYSQL. [En línea] [Citado el: 25 de Noviembre de 2010.] <http://www.webestilo.com/mysql/intro.phtml>.
19. dbdesigner4. [En línea] [Citado el: 24 de Noviembre de 2010.] <http://www.fabforce.net/dbdesigner4/>.
20. ubuntu.. [En línea] 9 de febrero de 2011. [Citado el: 17 de febrero de 2011.] http://doc.ubuntu-es.org/Ubuntu#Principios_de_Ubuntu
21. Maestro del web. [En línea] [Citado el: 30 de Noviembre de 2010.] <http://www.maestrosdelweb.com/editorial/phpintro/>.
22. cysfe. [En línea] [Citado el: 20 de noviembre de 2011.] <http://www.crysfel.com/archivo/>.
23. netbeans. [En línea] [Citado el: 28 de Noviembre de 2010.] <http://blogultura.com/java/netbeans-6-8-liberado/>.
24. netbeans. [En línea] [Citado el: 19 de Enero de 2011.] <http://netbeans.org/community/releases/69/>.
25. Symfony. [En línea] [Citado el: 5 de Diciembre de 2010.] <http://www.symfony.es/2009/10/05/netbeans-ya-incluye-soporte-para-symfony/>.
26. *Documento Visión SIGICEM*. Cuba : Publicación, 2011.
27. MVC. [En línea] [Citado el: 24 de Noviembre de 2010.] <http://www.ucbcb.edu.bo/Publicaciones/revistas/actanova/documentos/v2n4/v2.n4.bascon.pdf>
28. *Idem 27*.
29. *Idem 27*.
30. **Hernández, Kati**. *Aplicación Web para el Control de la Asistencia a la facultad 7*. Cuba : s.n., 2011.
31. *Idem 30*.
32. Libros Web.es. [En línea] [Citado el: 18 de Febrero de 2011.] http://www.librosweb.es/symfony/capitulo6/metodos_de_validacion_y_manejo_de_errores.html
33. *Idem 32*.



Bibliografía

- **Barros, Nelismi.(Tesi) SIGICEM: Módulo de Gestión Tecnológica.** Cuba : Publicación, 2009
- **CMMI.** [En línea] [Citado el: 19 de Enero de 2011.] <http://www.ingenierossoftware.com/calidad/cmm-cmmi.php>
- Comparación entre la representación continua y la representación por etapa. [En línea] [Citado el: 19 de Enero de 2011.]
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/ulibbarri_p_im/capitulo4.pdf.
- **Desoft.** [En línea] [Citado el: 26 de Noviembre de 2010.]
<http://www.desoft.cu/Productos1/Vindsor/tabid/437/Default.aspx>.
- **Designer4.** [En línea] [Citado el: 28 de Noviembre de 2010.] <http://www.fabforce.net/dbdesigner4/>.
- **Desoft.** [En línea] [Citado el: 24 de Noviembre de 2010]
<http://www.desoft.cu/Productos1/Vindsor/tabid/437/Default.aspx>
- *Desarrollo de los módulos Reporte de Eventos Adversos por las Unidades de Salud (REM) y Reporte de Eventos Adversos por los Fabricantes de Equipos Médicos (RFAB)* . Cuba : Publicaciones, 2011.
- **Diagrama de componentes. Cuba :** Aplicación Web para el Control de la Asistencia a la facultad 7, 2011.
- **Dragofac.** [En línea] [Citado el: 15 de Diciembre de 2010.]
<http://gratis.portalprogramas.com/Dragofac.html>.
- **dbdesigner4.** [En línea] [Citado el: 24 de Noviembre de 2010.]
<http://www.fabforce.net/dbdesigner4/>.
- **Documento Visión SIGICEM.** [En línea] [Citado el: 20 de Enero de 2011.]
- **Electromedicina.** [En línea] 14 de Diciembre de 2010. [Citado el: 14 de Diciembre de 2010.]
http://www.wikilearning.com/curso_gratis/la_electromedicina_o_ingeneria_hospitalaria/26817.
- **Etapas de diseño.** [En línea] [Citado el: 20 de Enero de 2011.]
<http://www.scribd.com/doc/395783/RUP-etapa-diseno>.
- **Express-Invoice.** [En línea] [Citado el: 24 de Noviembre de 2010.]
<http://gratis.portalprogramas.com/Express-Invoice.html>.
- **Factura.** [En línea] [Citado el: 16 de Noviembre de 2010.] <http://www.aigclassic.com/articulos/que-es-una-factura>.
- **FactuSol.** [En línea] [Citado el: 24 de Noviembre de 2010.]
<http://gratis.portalprogramas.com/FactuSol.html>.



- **Gestión de cliente.** [En línea] [Citado el: 25 de Octubre de 2010.] <http://apuntes-negocios.blogspot.com/2010/05/24-gestion-de-relaciones-con-el-cliente.html>.
- **Hernández León, Rolando Alfredo.** *El paradigma cuantitativo de la investigación científica* . s.l. : Publicación, 2002.
- **Hernández, Kati.** *Aplicación Web para el Control de la Asistencia a la facultad 7.* Cuba : s.n., 2011.
- **LibrosWeb.es.** [En línea] [Citado el: 18 de Febrero de 2011.] http://www.librosweb.es/symfony/capitulo6/metodos_de_validacion_y_manejo_de_errores.html
- **Lenguaje Unificado de modelado.** [En línea] [Citado el: 19 de Enero de 2011.] file:///F:/Herramientas%20para%20el%20primer%20capitulo/UML/UML_-_Lenguaje_Unificado_de_Modelado.html
- **Michael Marca Huallpara, Hugo.** *Trabo de investigación y expocición.* s.l. : publicacion, 2011.
- **Macías, Charlie y Orozco, Sergio.** *Uso de UML en aplicaciones Web: páginas y relaciones* . [En línea] [Citado el: 23 de febrero de 2011.] http://www.milestone.com.mx/articulos/uso_de_uml_en_aplicaciones_web.htm.
- **Molecule-CRM.** [En línea] [Citado el: 18 de Noviembre de 2010.] <http://gratis.portalprogramas.com/Molecule-CRM.html>.
- **MYSQL.** [En línea] [Citado el: 25 de Noviembre de 2010.] <http://www.webestilo.com/mysql/intro.phtml>.
- **Maestro del web.** [En línea] [Citado el: 30 de Noviembre de 2010.] <http://www.maestrosdelweb.com/editorial/phpintro/>.
- **MVC.** [En línea] [Citado el: 24 de noviembre de 2010.] <http://www.ucbcba.edu.bo/Publicaciones/revistas/actanova/documentos/v2n4/v2.n4.bascon.pdf>.
- **netbeans.** [En línea] [Citado el: 28 de Noviembre de 2010.] <http://blogultura.com/java/netbeans-6-8-liberado/>.
- **netbeans.** [En línea] [Citado el: 19 de Enero de 2011.] <http://netbeans.org/community/releases/69/>.
- **Pagares-y-cobros-Pro.** [En línea] [Citado el: 24 de Noviembre de 2010.] <http://gratis.portalprogramas.com/Pagares-y-cobros-Pro.html>.
- **RUP.** [En línea] [Citado el: 24 de Noviembre de 2010.] <http://yaqui.mxl.uabc.mx/~molquin/as/RUP.htm>.
- **Servidor apache.** [En línea] [Citado el: 24 de Noviembre de 2010.] <http://httpd.apache.org/docs/2.0/>.
- **SeniorFactu.** [En línea] [Citado el: 24 de Noviembre de 2010.] <http://gratis.portalprogramas.com/SeniorFactu.html>.
- **Servidor web apache.** [En línea] [Citado el: 28 de Noviembre de 2010.] <http://httpd.apache.org/docs/2.0/>.



- **Symfony.** [En línea] [Citado el: 5 de Diciembre de 2010.]
<http://www.symfony.es/2009/10/05/netbeans-ya-incluye-soporte-para-symfony/>.
- **Unidad Empresarial de Base División de Sistemas Gerenciales, CITMATEL-2005.** *Sistema Integral de Gestión Económica Administrativa.* Cuba : s.n., 2011.
- **ubuntu.** [En línea] 9 de febrero de 2011. [Citado el: 17 de febrero de 2011.] http://doc.ubuntu-es.org/Ubuntu#Principios_de_Ubuntu
- **Uso de UML en aplicaciones Web: páginas y relaciones .** Uso de UML en aplicaciones Web: páginas y relaciones . *Uso de UML en aplicaciones Web: páginas y relaciones .* [En línea] [Citado el: 23 de Febrero de 2011.]
http://www.milestone.com.mx/articulos/uso_de_uml_en_aplicaciones_web.htm.
- **UNEC .** [En línea] [Citado el: 14 de Diciembre de 2010.]
http://www.aloj.us.es/notas_tecnicas/Adaptacion_UNE_EN13269_a_Mto_de_Electromedicina.pdf
- **Unidad Empresarial de Base División de Sistemas Gerenciales, CITMATEL-2005.** *Sistema Integral de Gestión Económica Administrativa.* Cuba : s.n., 2011.
- **Uso de UML en aplicaciones web.** [En línea] [Citado el: 21 de Enero de 2011.]
http://www.milestone.com.mx/articulos/uso_de_uml_en_aplicaciones_web.htm.
- **Vindsor.** [En línea] [Citado el: 24 de 10 de 2010.]
<http://www.desoft.cu/Productos1/Vindsor/tabid/437/Default.aspx>
- **Visual parading.** [En línea] [Citado el: 24 de Noviembre de 2010.] <http://www.visual-paradigm.com/product/vpum/>.



Glosario de términos

BD: Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

CCEEM: Centro para el Control Estatal de Equipos Médicos.

CICEM: Centro de Ingeniería Clínica y Electromedicina.

CESIM: Centro de Informática Médica.

CUN: Caso de uso del negocio.

CUS: Caso de uso del sistema.

CU: Casos de Usos.

ERP: Enterprise Resource Planning es una herramienta de estrategia, que equipa a la empresa con las capacidades y recursos necesarios para integrar y sincronizar las funciones aisladas en un proceso continuo de negocios en miras de ganar una competitividad en el turbulento ambiente de los negocios.

Framework: Se conoce como marco de trabajo y constituye un conjunto de conceptos, metodologías y herramientas de administración y diseño para el desarrollo de forma estandarizada de una aplicación.

Facturación: Documento mercantil que refleja toda la información de una operación de compraventa. La información fundamental que aparece en una factura es que debe reflejar la entrega de un producto o la provisión de un servicio, junto a la fecha de devengo, además de indicar la cantidad a pagar como contraprestación.

Gestión de Clientes: gestionar todos los contactos entre profesionales y clientes necesarios para una correcta organización del negocio y de toda la información que éste genera. Almacena los datos de clientes, empresas asociadas, profesionales, ventas y otros.

GADE: Grupo de Automatización y Desarrollo de Electromedicina.



HTTP: (Hypertext Transfer Protocol). Se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que se conoce como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

Herramientas CASE: Conjunto de aplicaciones informáticas orientadas al incremento de la productividad en el desarrollo de software, las siglas CASE vienen dadas por su nombre en inglés Computer Aided Software Engineering que se conoce como Ingeniería de Software Asistida por Computadoras.

HTML: HyperText Markup Language (Lenguaje de Marcas de Hipertexto).

MINSAP: Ministerio de Salud Pública de la República de Cuba es el organismo rector del Sistema Nacional de Salud, encargado de dirigir, ejecutar y controlar la aplicación de la política del estado y del gobierno en cuanto a la Salud Pública, el Desarrollo de las Ciencias Médicas y de la industria farmacéutica.

MVC: Modelo Vista Controlador, es un patrón de arquitectura de software.

PHP: (Hypertext PreProcessor) Lenguaje de programación para el desarrollo de Web dinámicas. Originalmente se conocía como Personal Home Page tools, herramientas para páginas personales (en Internet).

PostgreSQL: Es un servidor de base de datos relacional libre, liberado bajo la licencia BSD.

Postgree SQL: Gestor de Bases de Datos (Square Query Language).

RUP: Rational Unified Process. El Proceso Unificado es un proceso de desarrollo de software (Conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software).

RNF: Requerimientos no funcionales.

RF: Requerimientos funcionales.

Symfony: Es un framework que está programado en PHP- 5 y está enfocado al desarrollo de aplicaciones web en el mismo lenguaje de programación.

SAS: Departamento de Sistemas de Apoyo a la Salud perteneciente al CESIM.

SIGICEM: Sistema de Gestión para la Ingeniería Clínica y Electromedicina.

SIGEM: Sistema de Gestión para la Electromedicina.



SNS: Sistema Nacional de Salud.

SNE: Sistema Nacional de Electromedicina.

TIC: Tecnologías de la Informática y las Comunicaciones.

UML: (Unified Modeling Language) Web Services: Servicio Web. Programa que implementa el protocolo.

UCI: Universidad de Ciencias Informáticas.

Anexo1

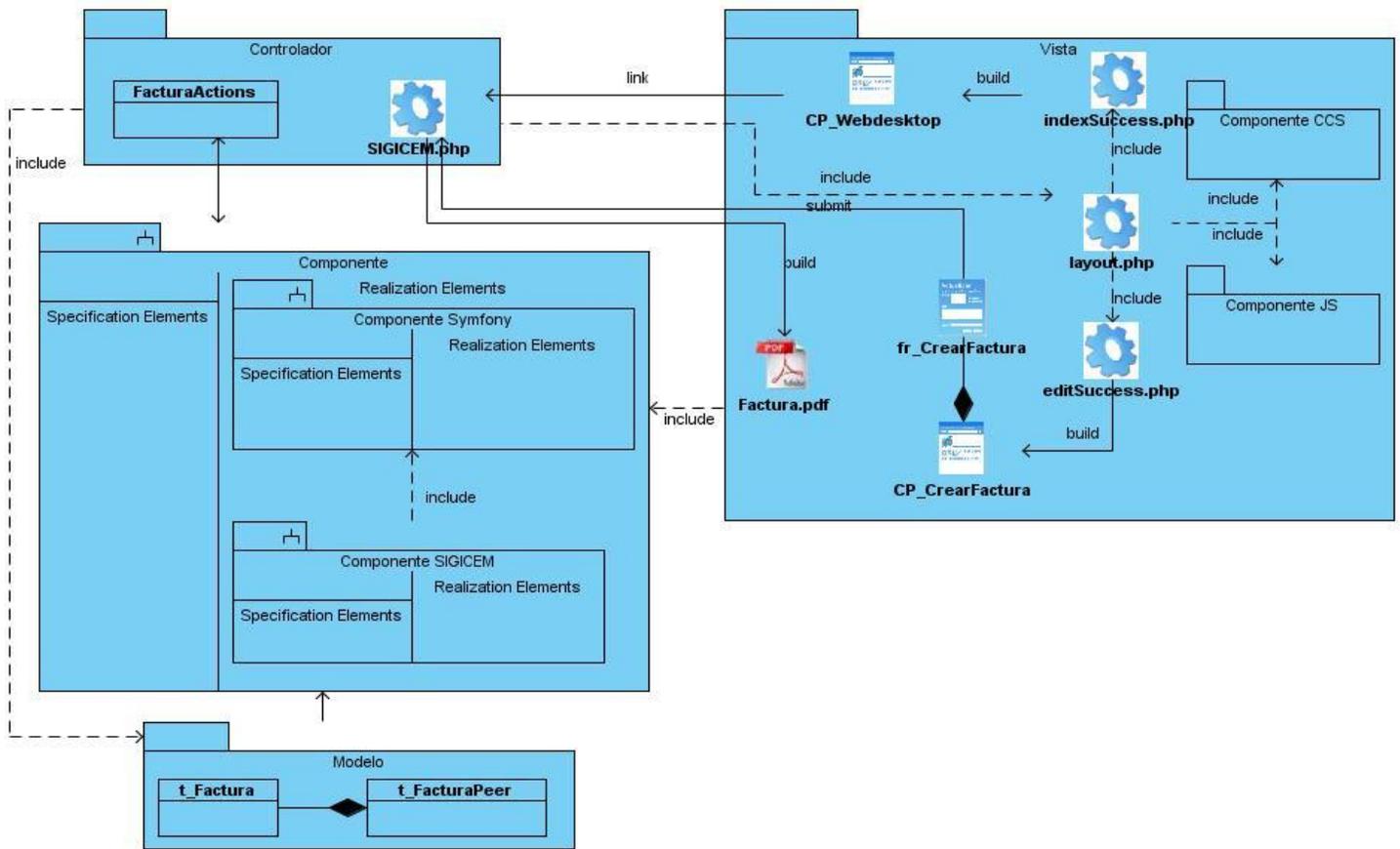


Diagrama. 1 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: Crear Factura.

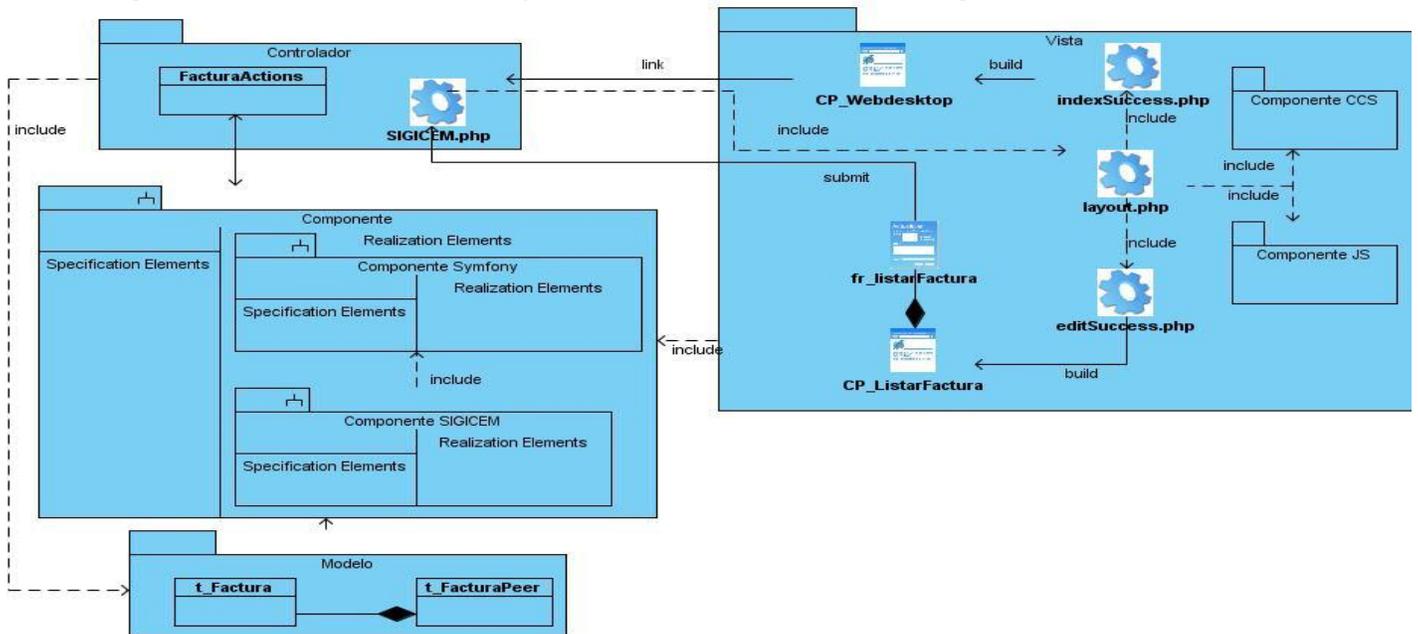


Diagrama. 2 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: Listar Factura.

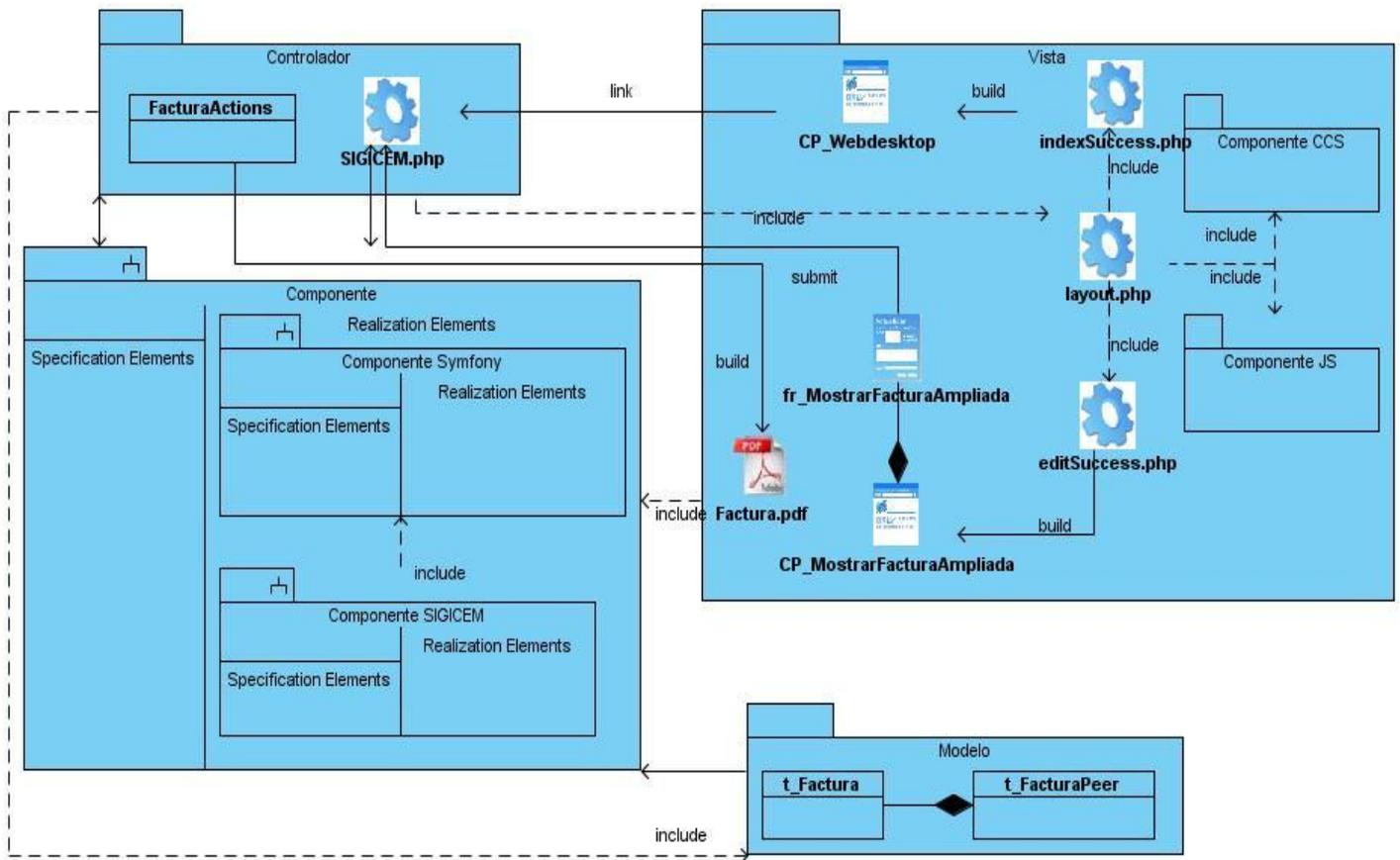


Diagrama. 3 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: *MostrarFacturaAmpliada*.

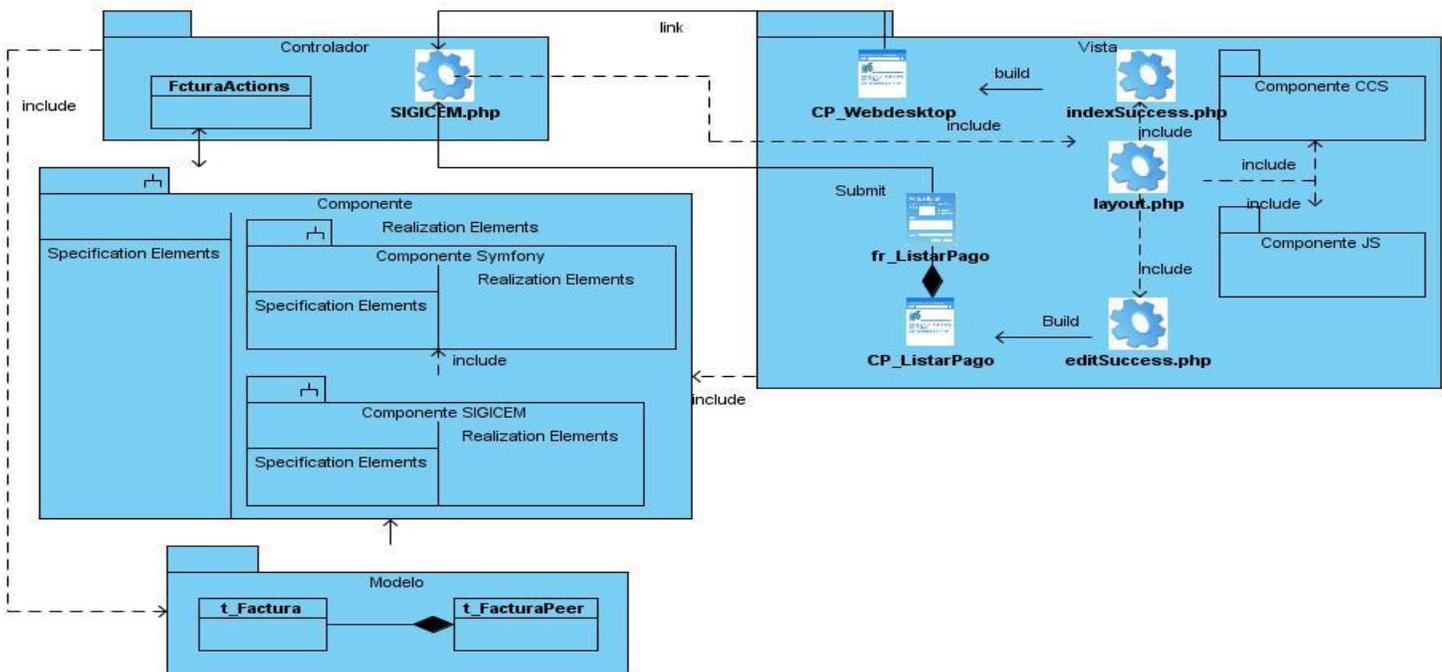


Diagrama. 4 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: *ListarPago*.

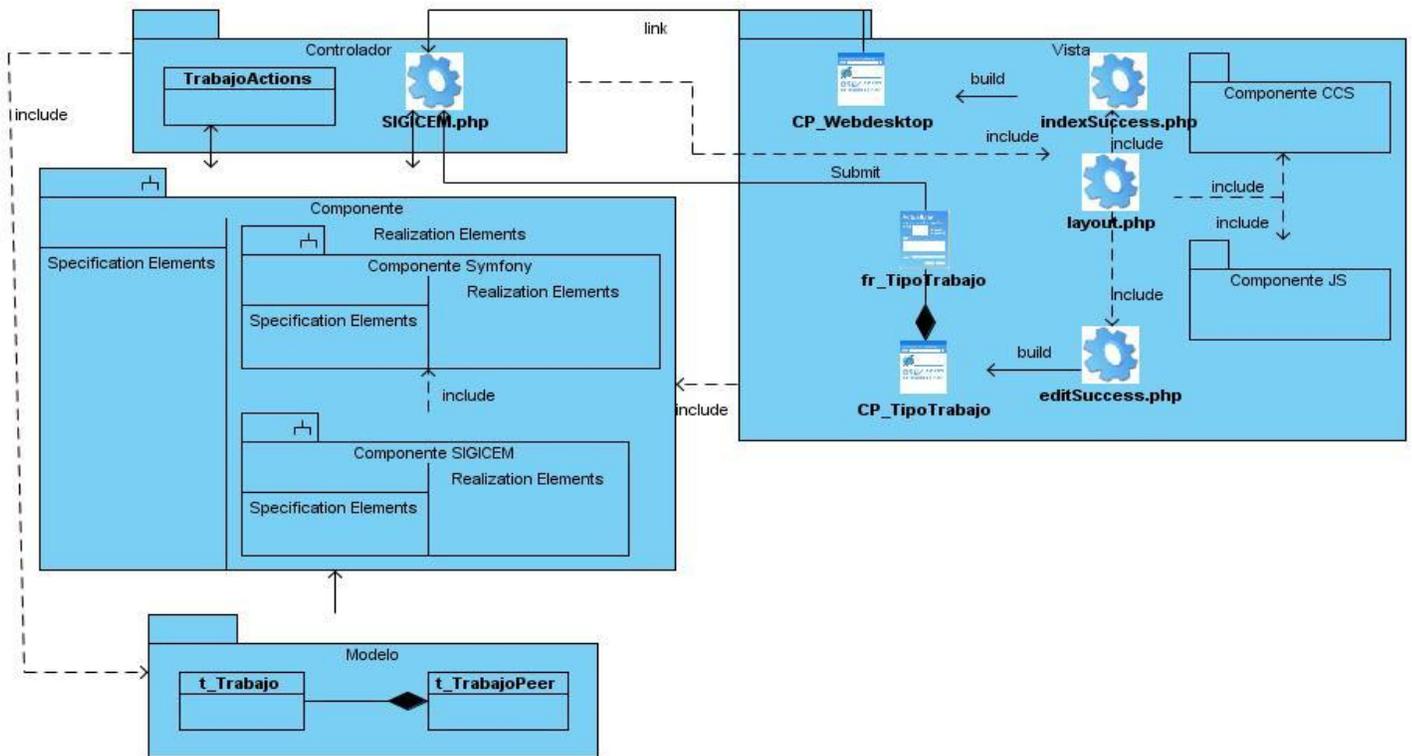


Diagrama. 5 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: ListarTrabajo.

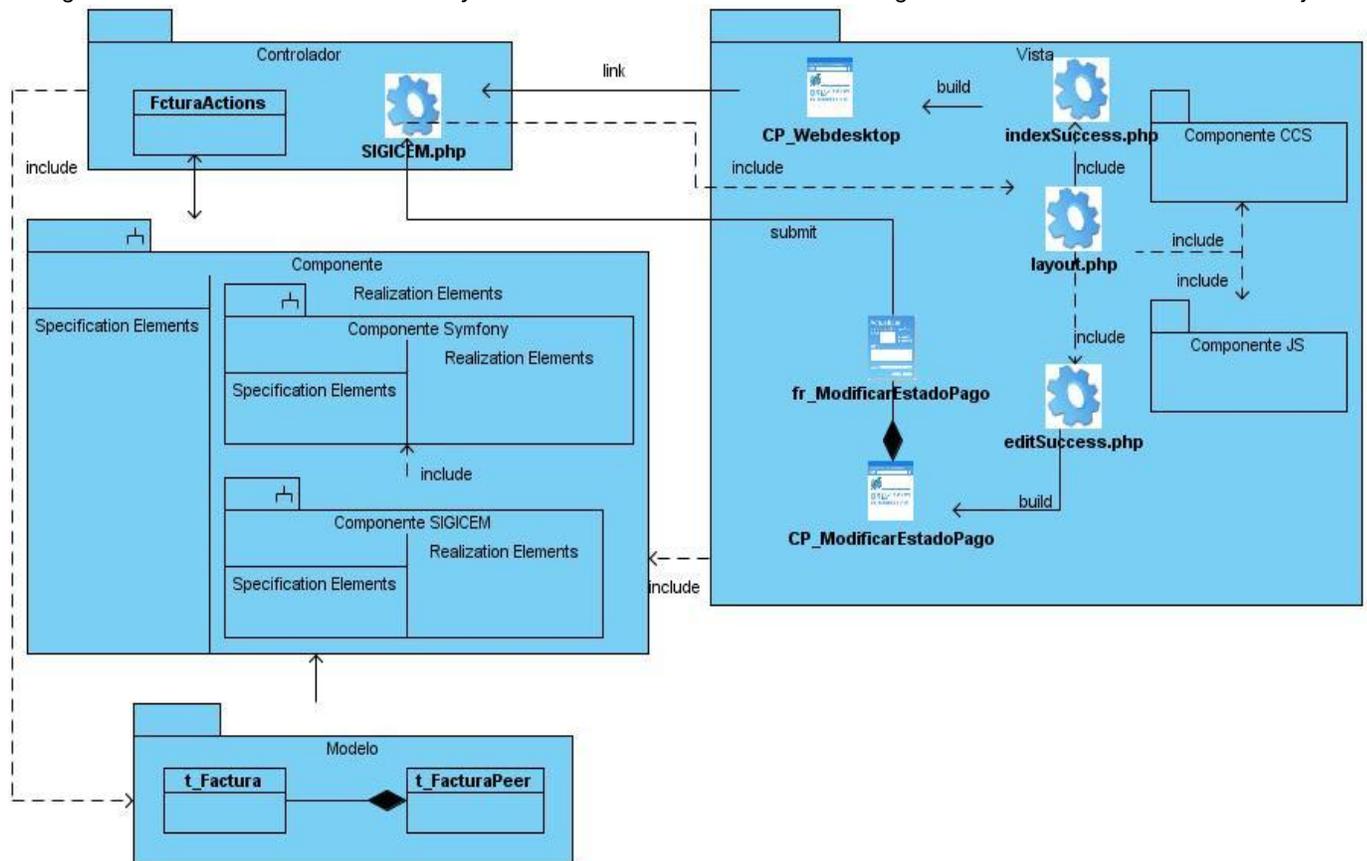


Diagrama. 6 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: ModificarEstadoPago.

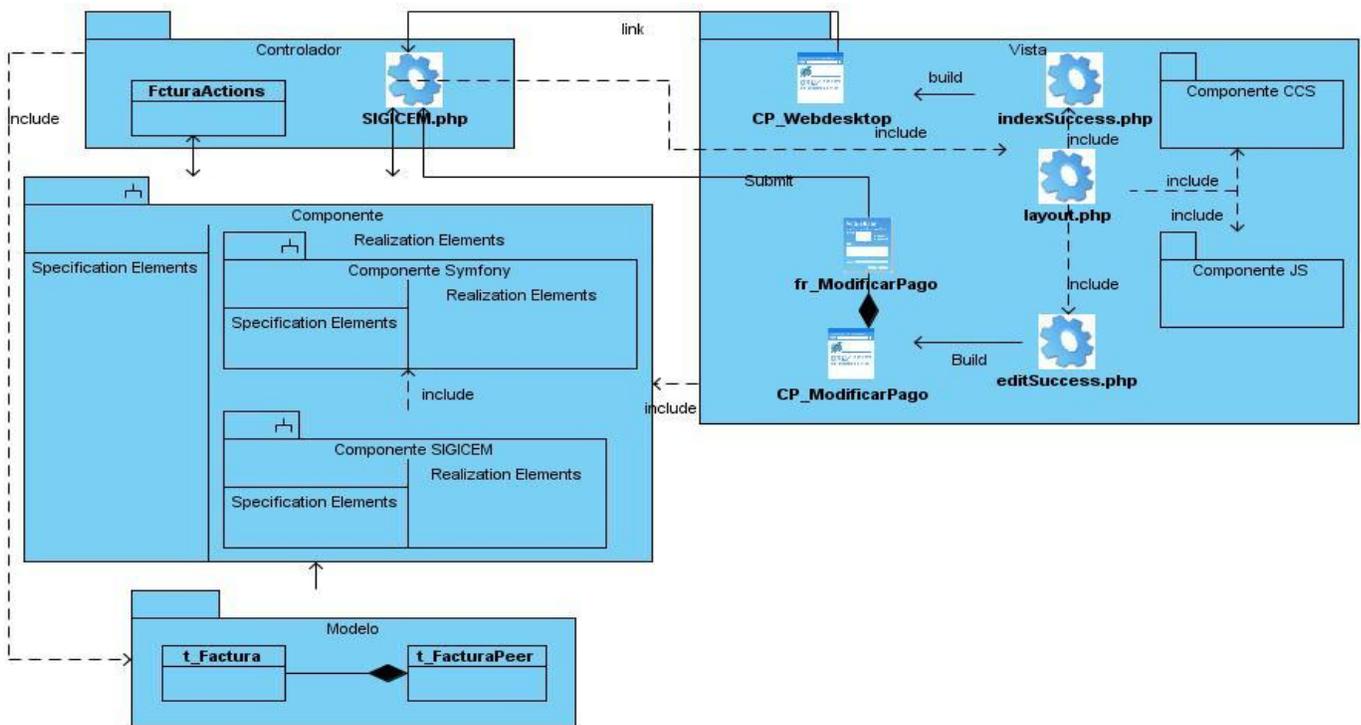


Diagrama. 7 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: ModificarPago.

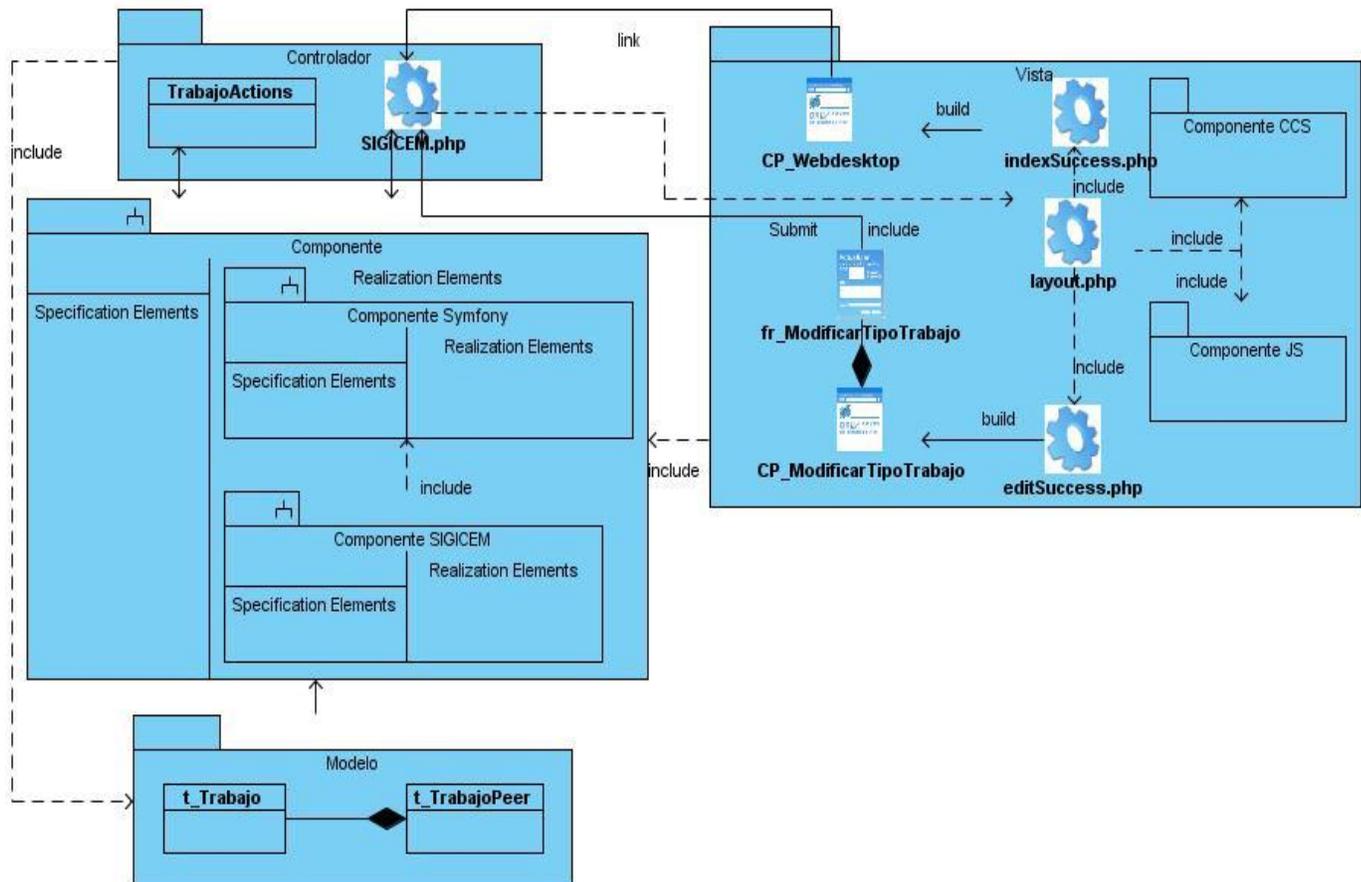


Diagrama. 8 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: ModificarTipoTrabajo.

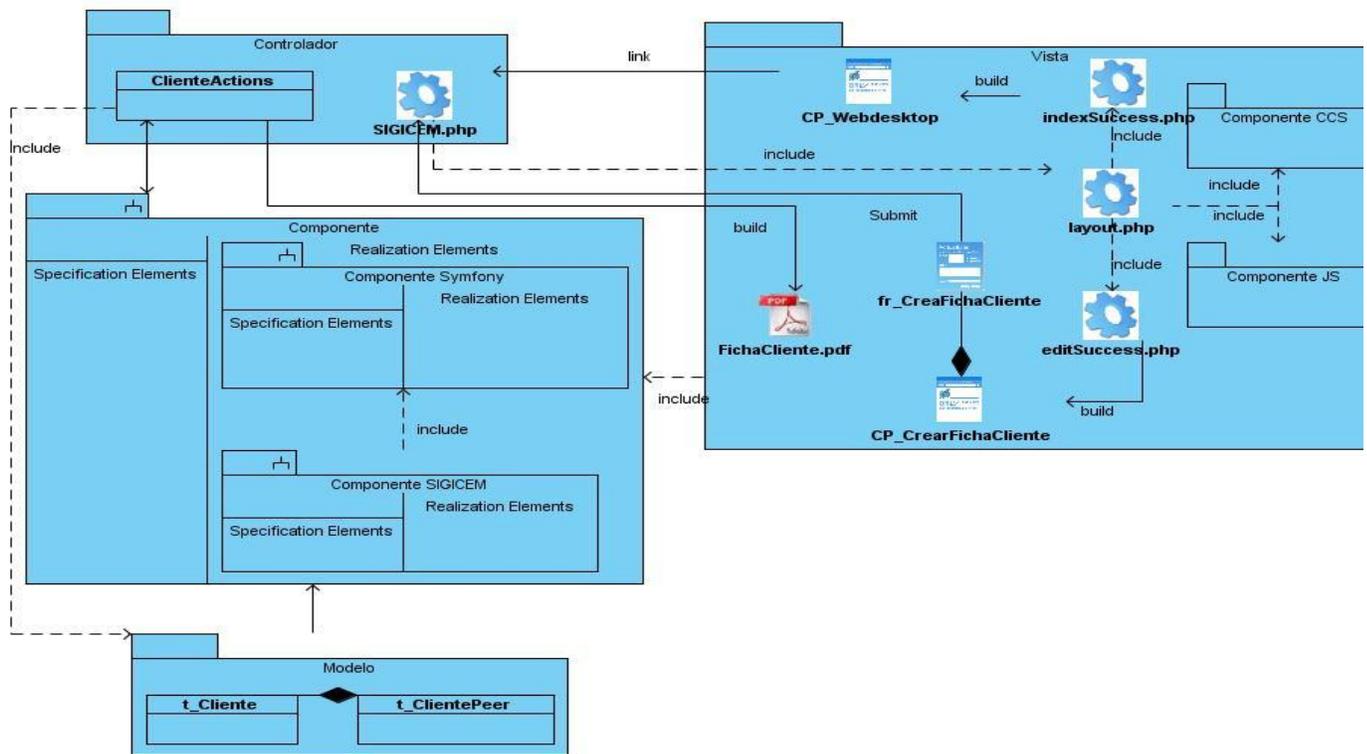


Diagrama. 9 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: Crear Ficha Cliente.

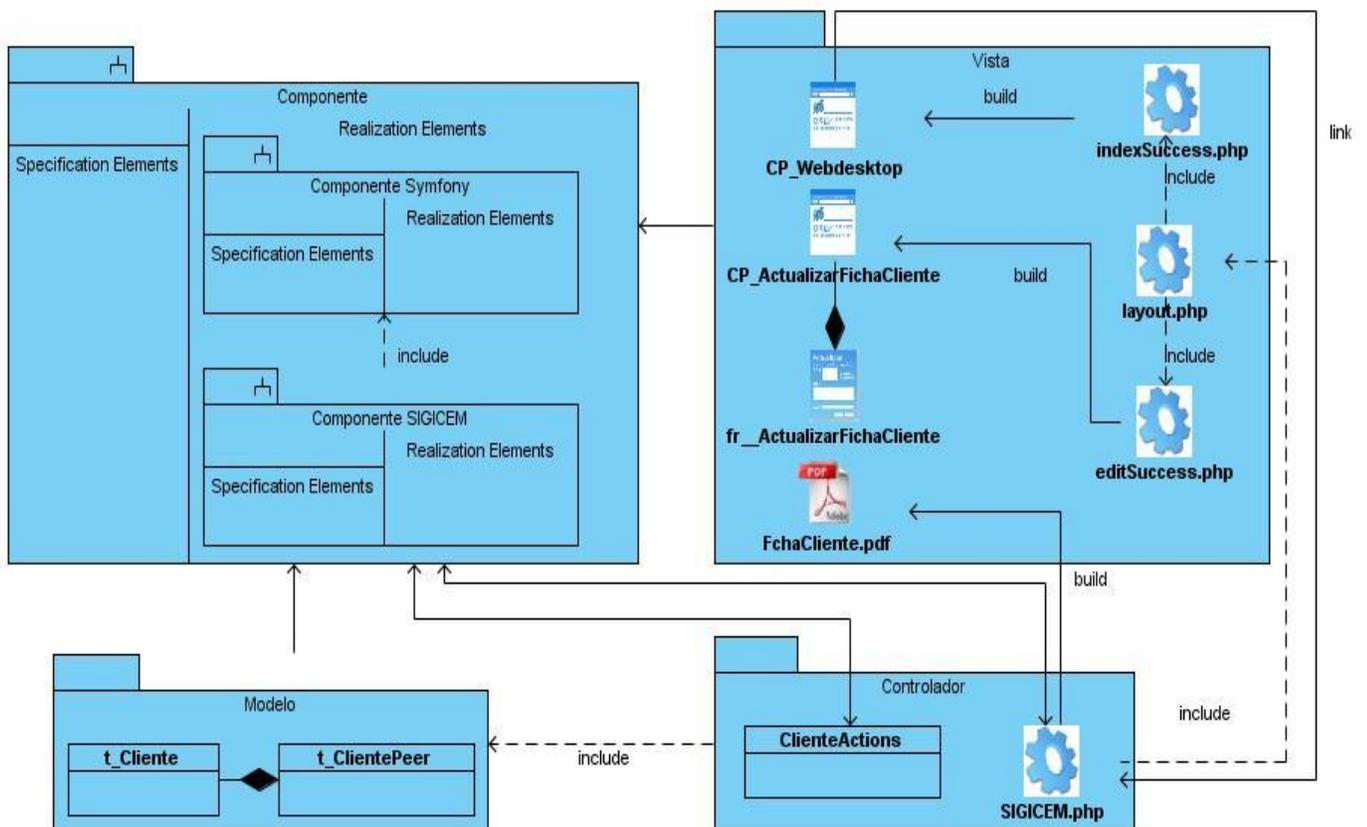


Diagrama. 10 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: Actualizar Ficha Cliente.

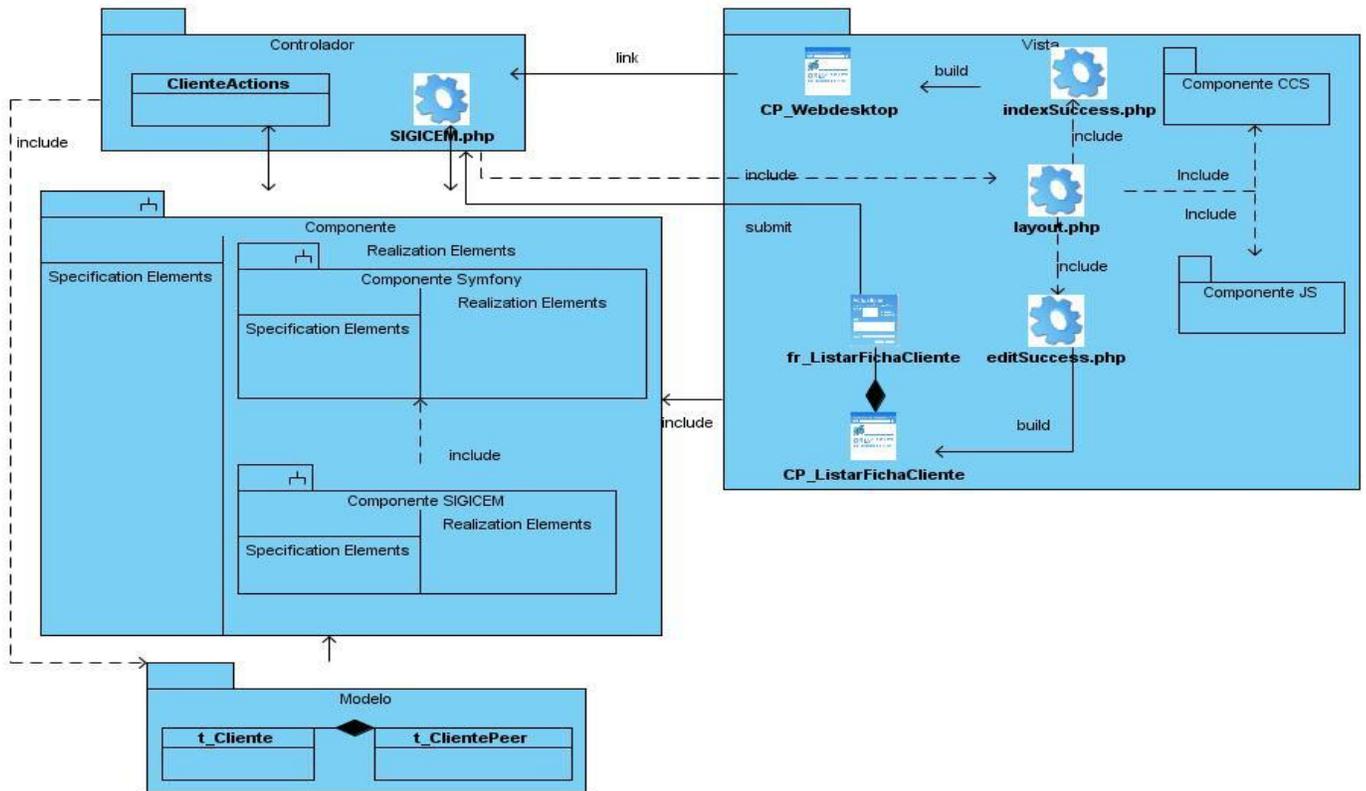


Diagrama. 11 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: ListarFichaCliente.

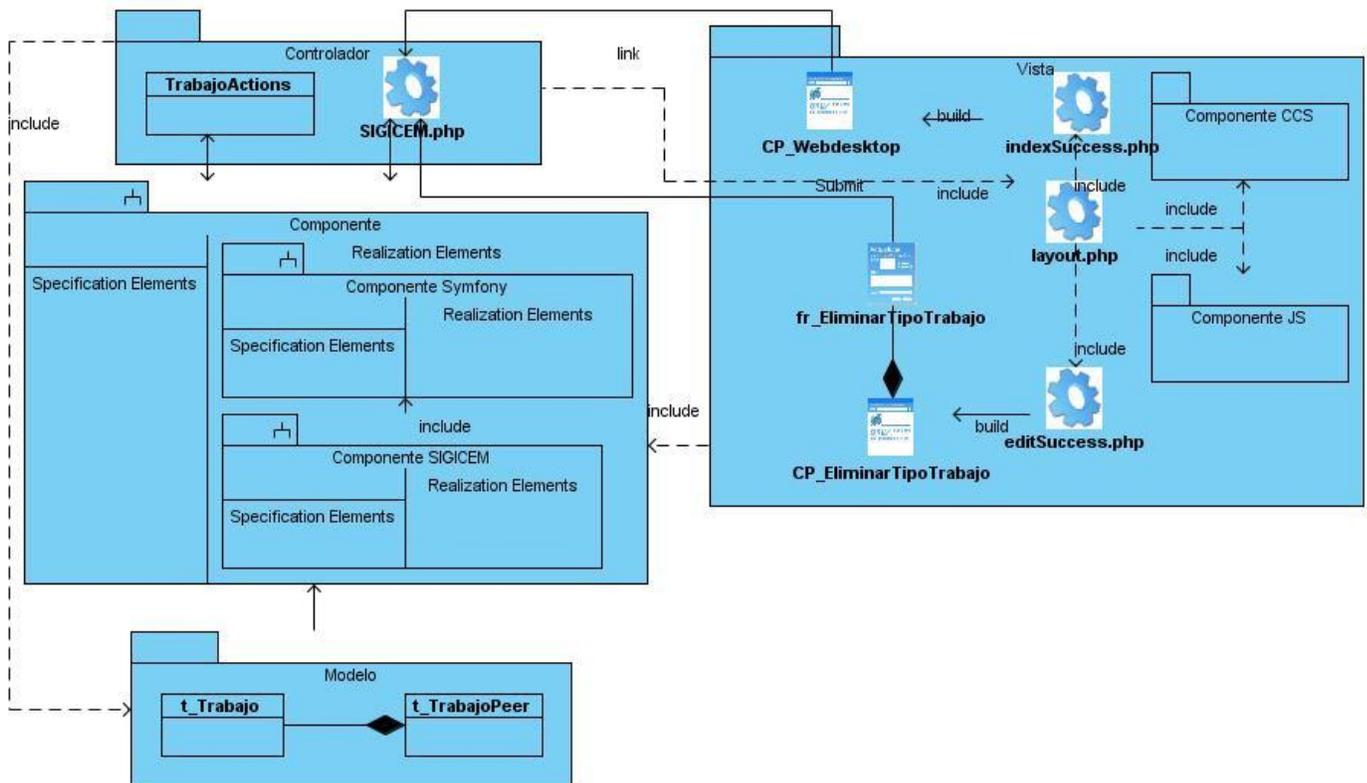


Diagrama. 12 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: EliminarTipoTrabajo.

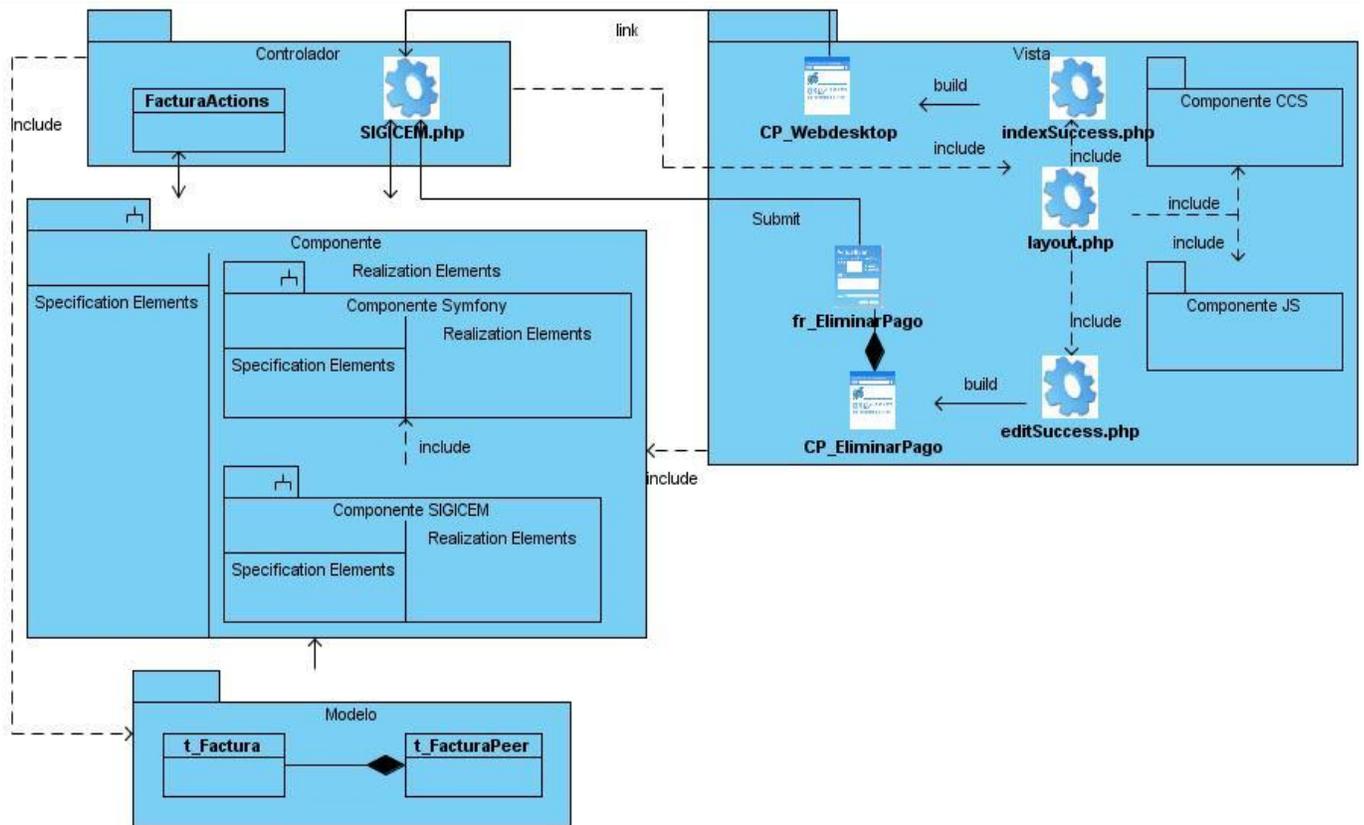


Diagrama. 13 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: EliminarPago.

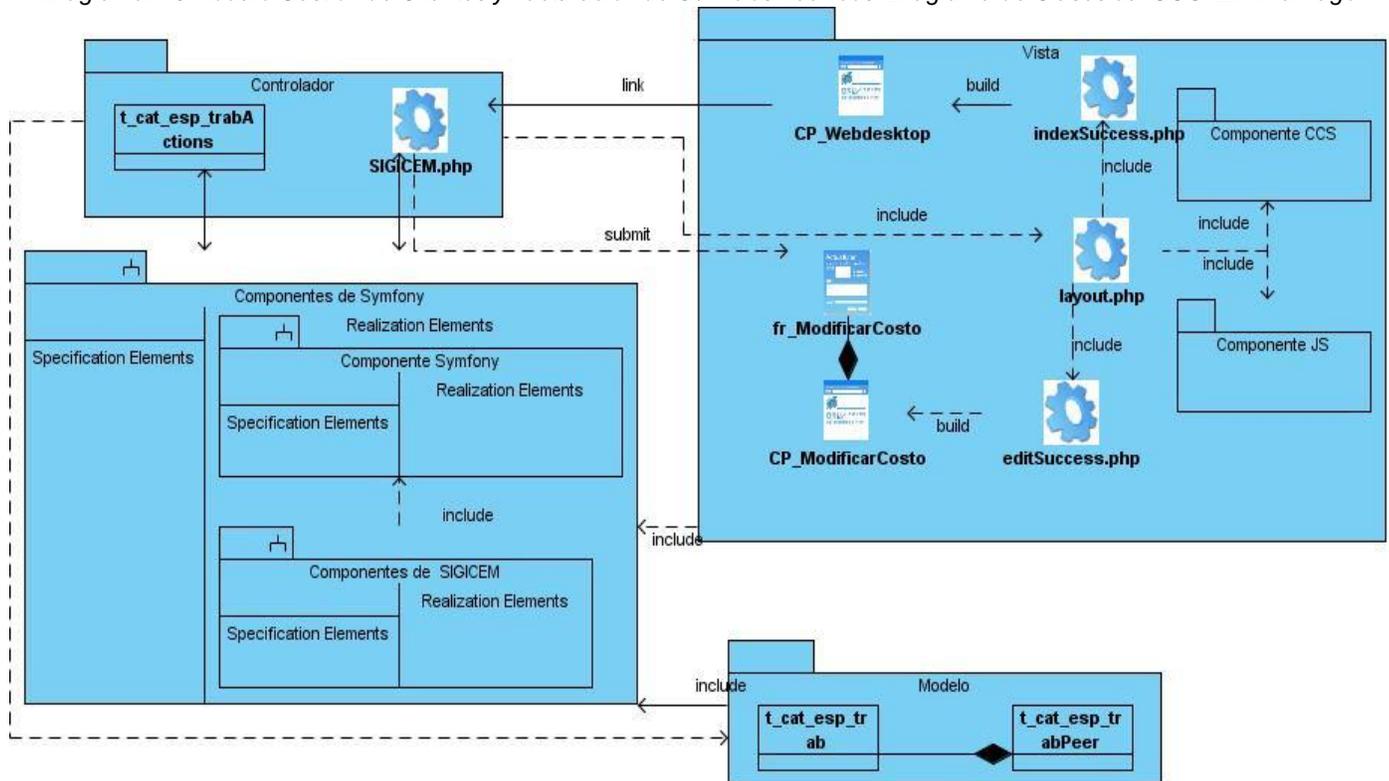


Diagrama. 14 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: ModificarCosto.

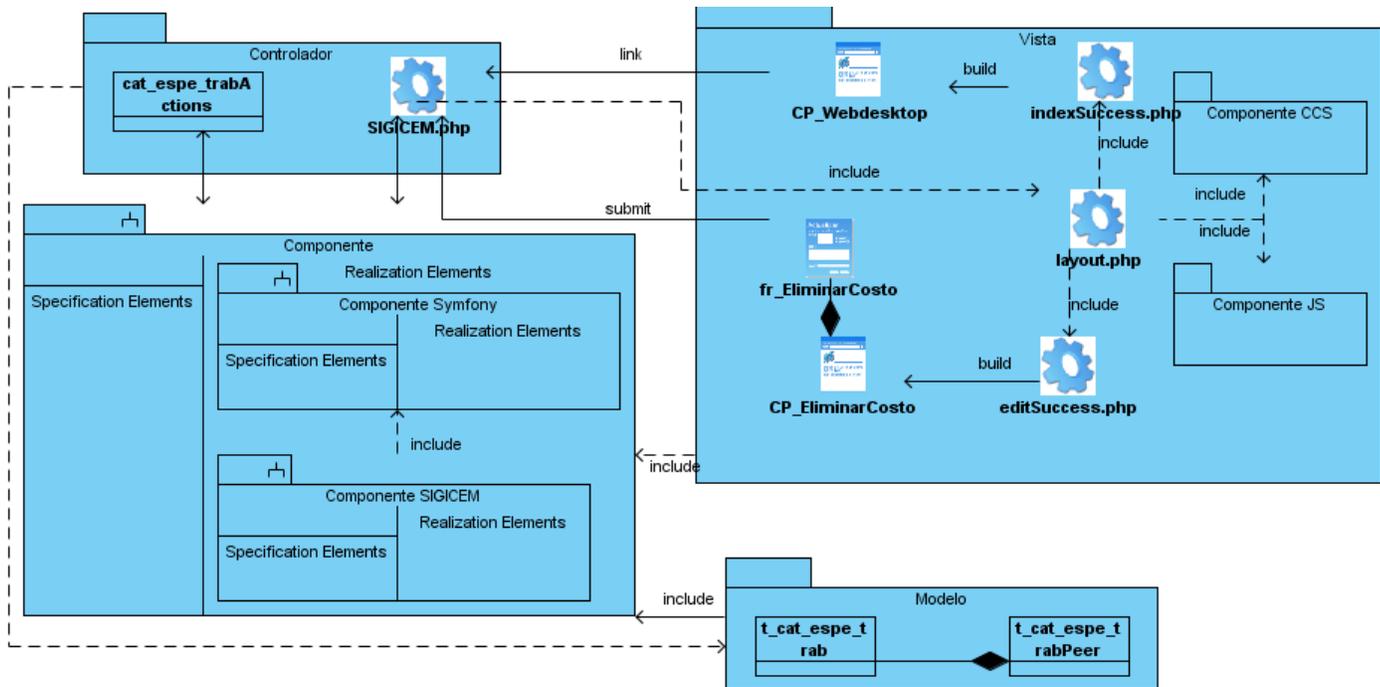


Diagrama. 15 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: EliminarCosto.

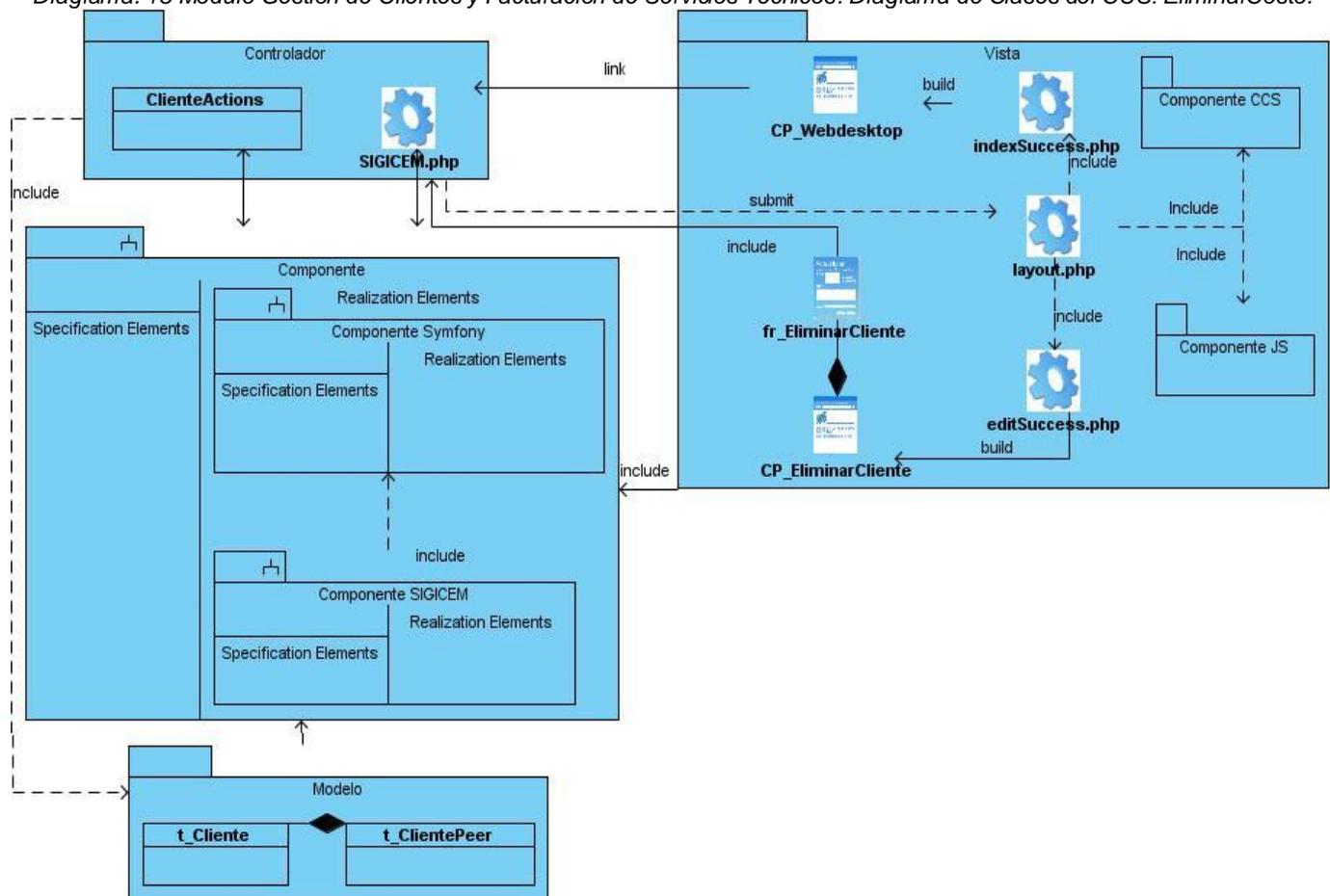


Diagrama. 16 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: EliminarCliente.

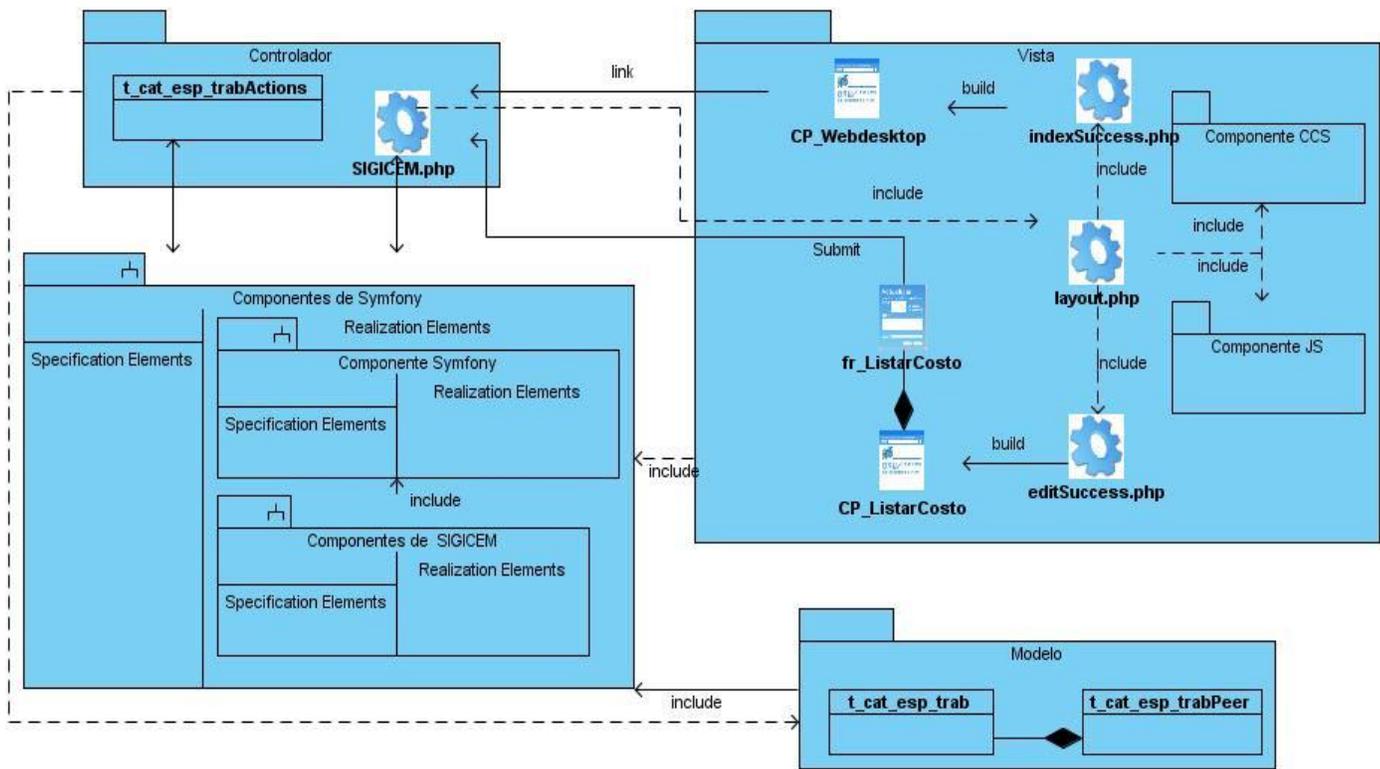


Diagrama. 17 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: Listar Costo.

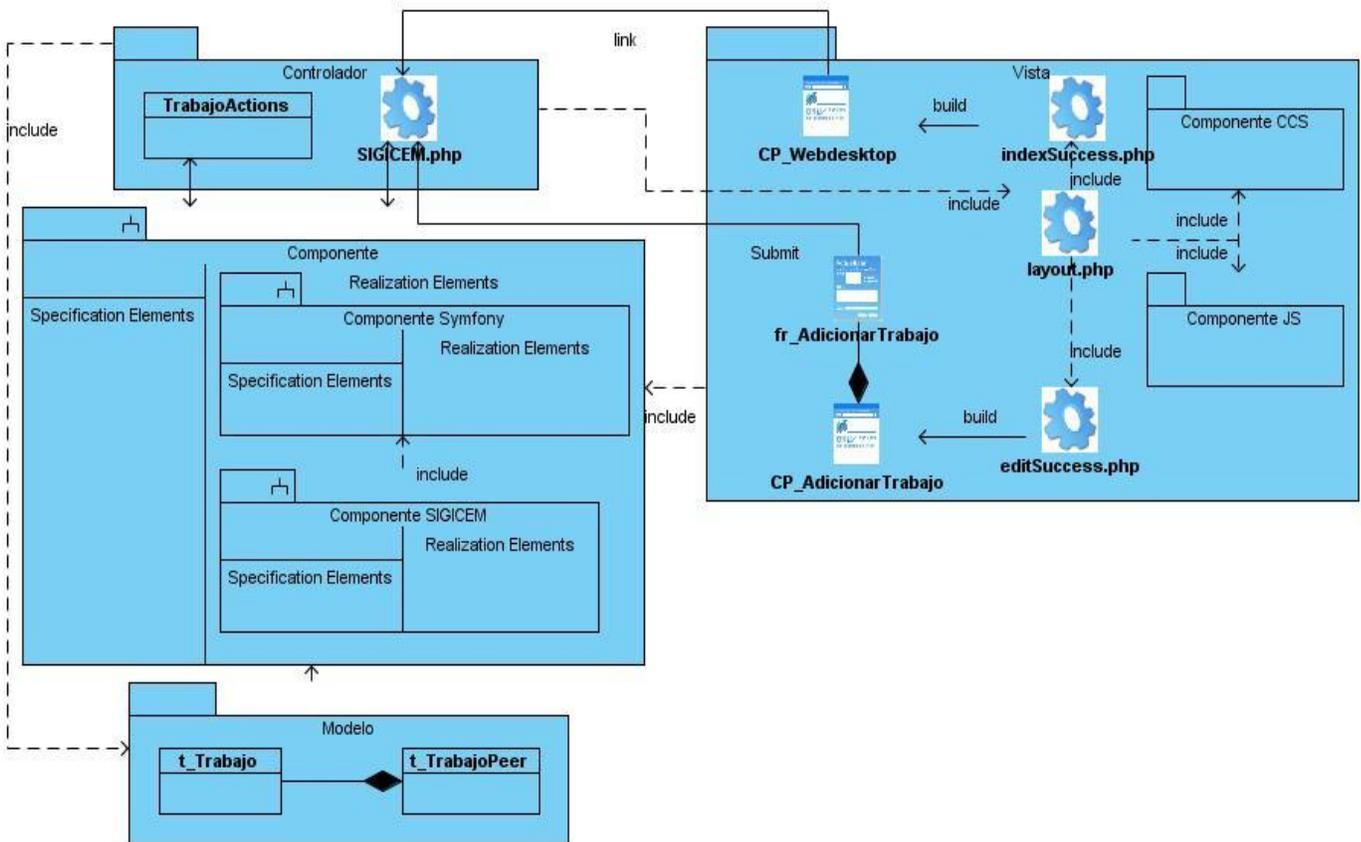


Diagrama. 18 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: Adicionar Trabajo.

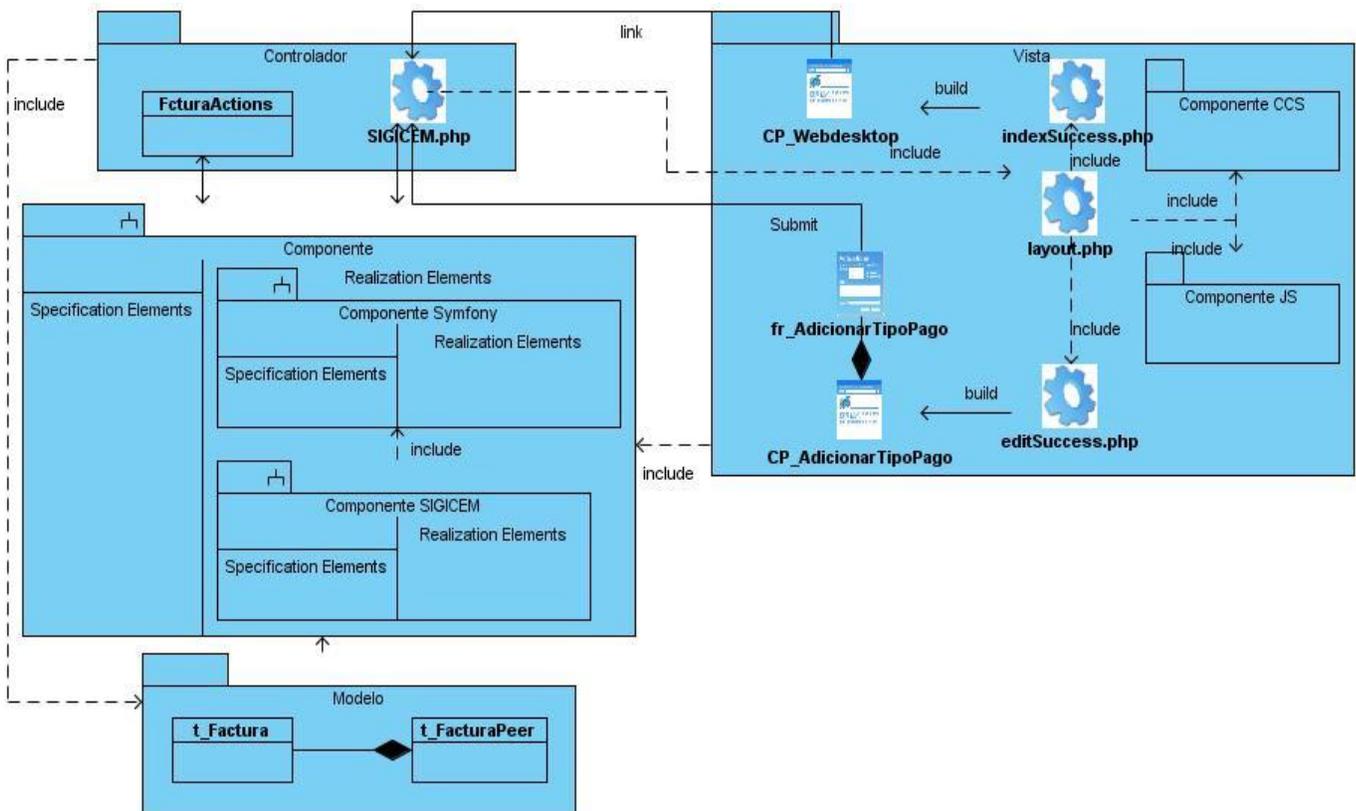


Diagrama. 19 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: AdicionarTipoPago.

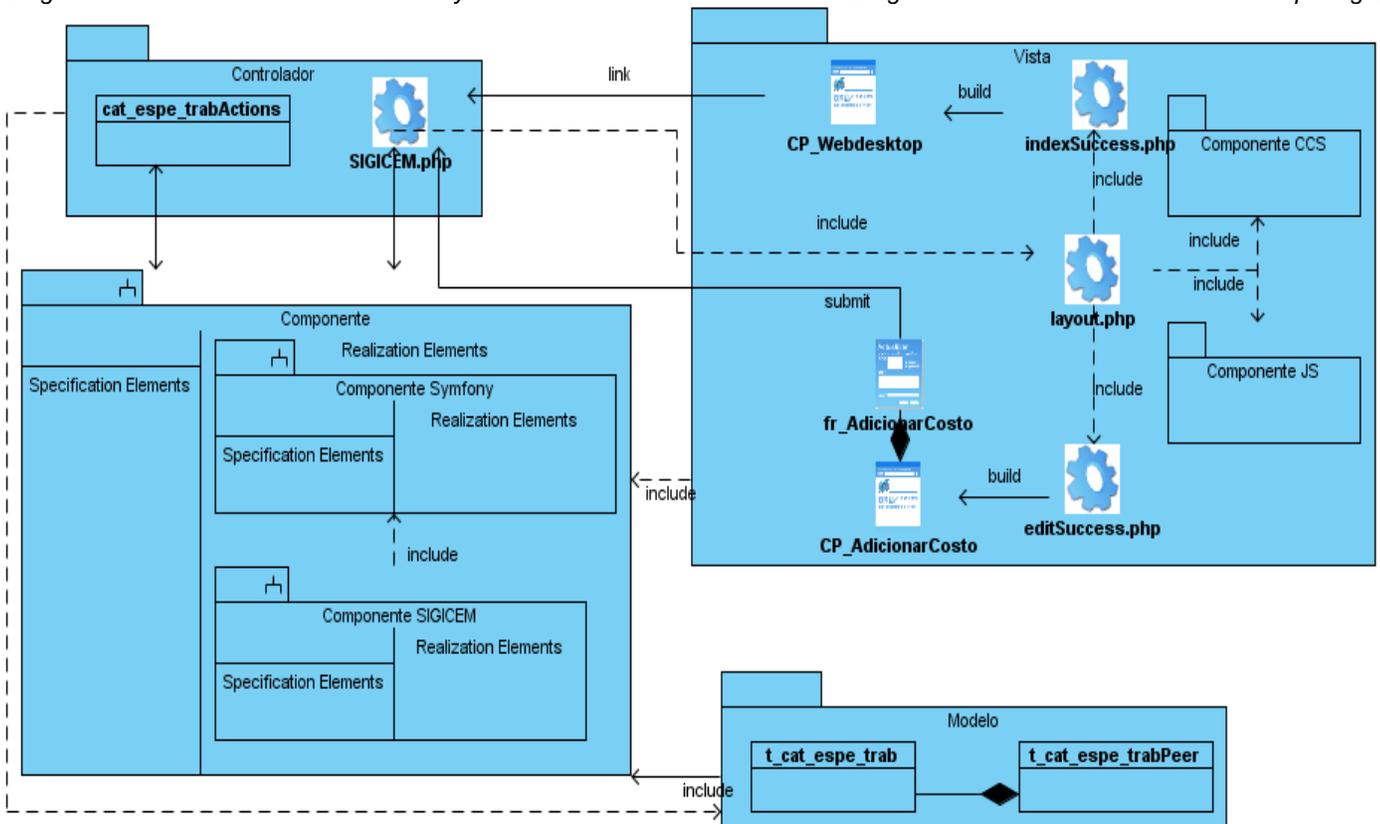


Diagrama. 20 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: AdicionarCosto.

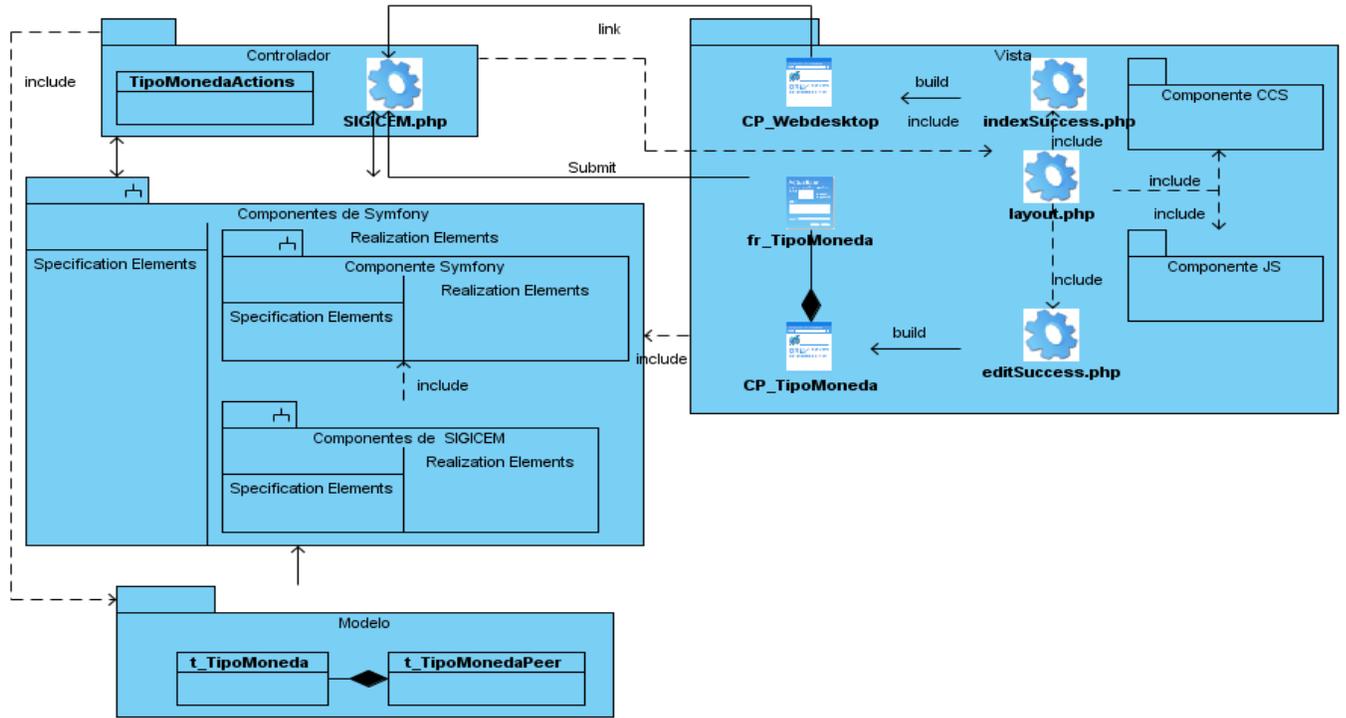


Diagrama. 21 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: ListarTipoMoneda.

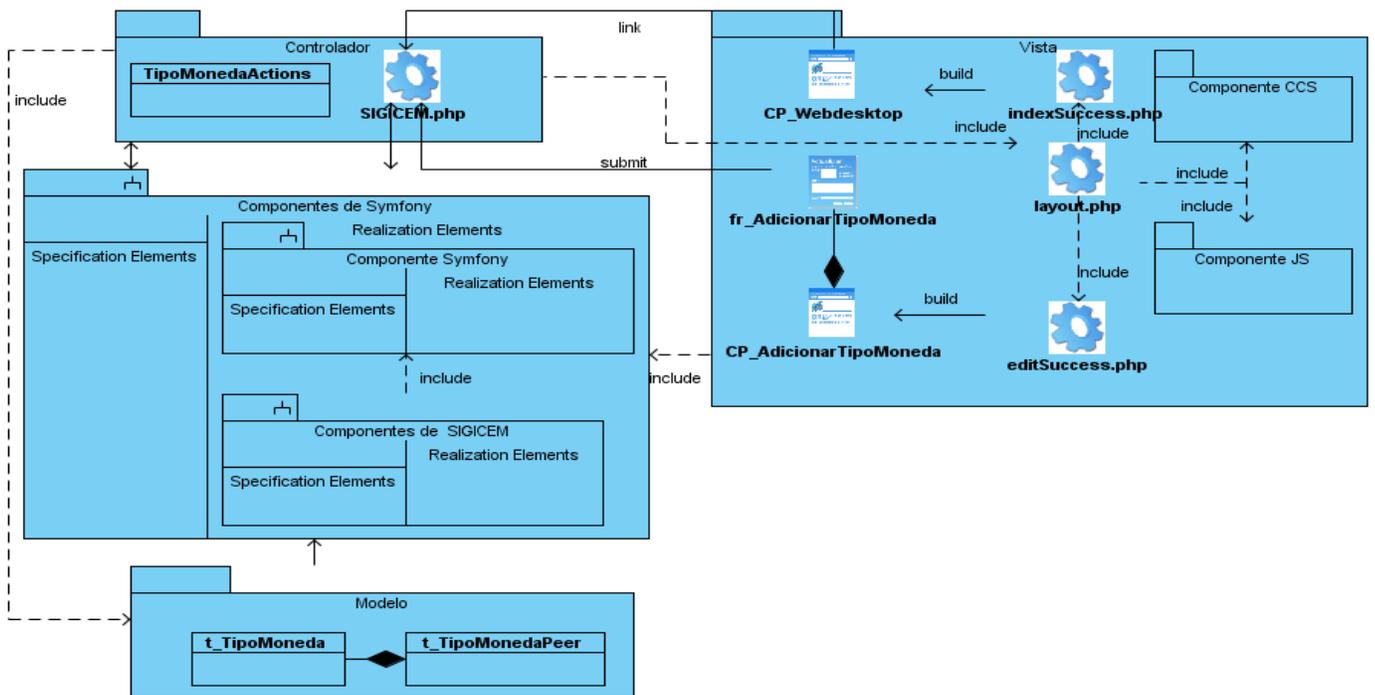


Diagrama. 22 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: AdicionarTipoMoneda

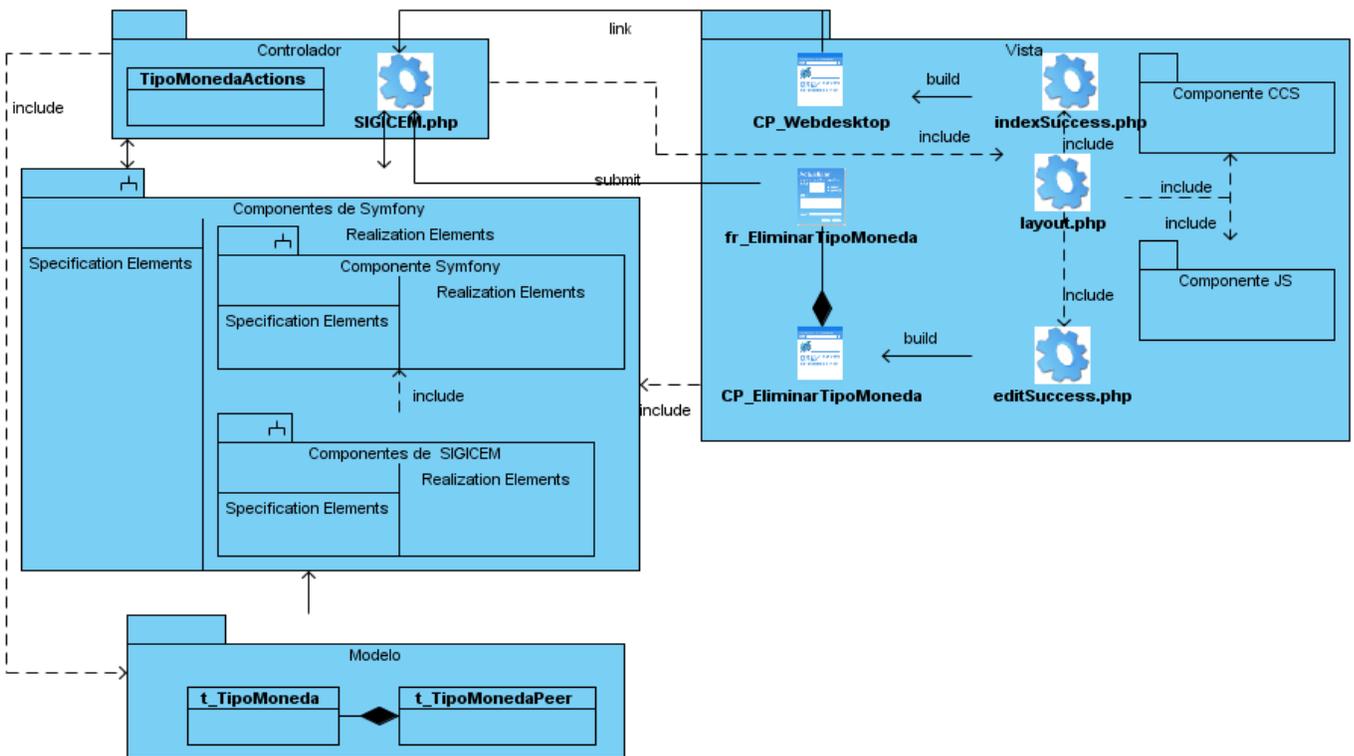


Diagrama. 23 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: *EliminarTipoMoneda*.

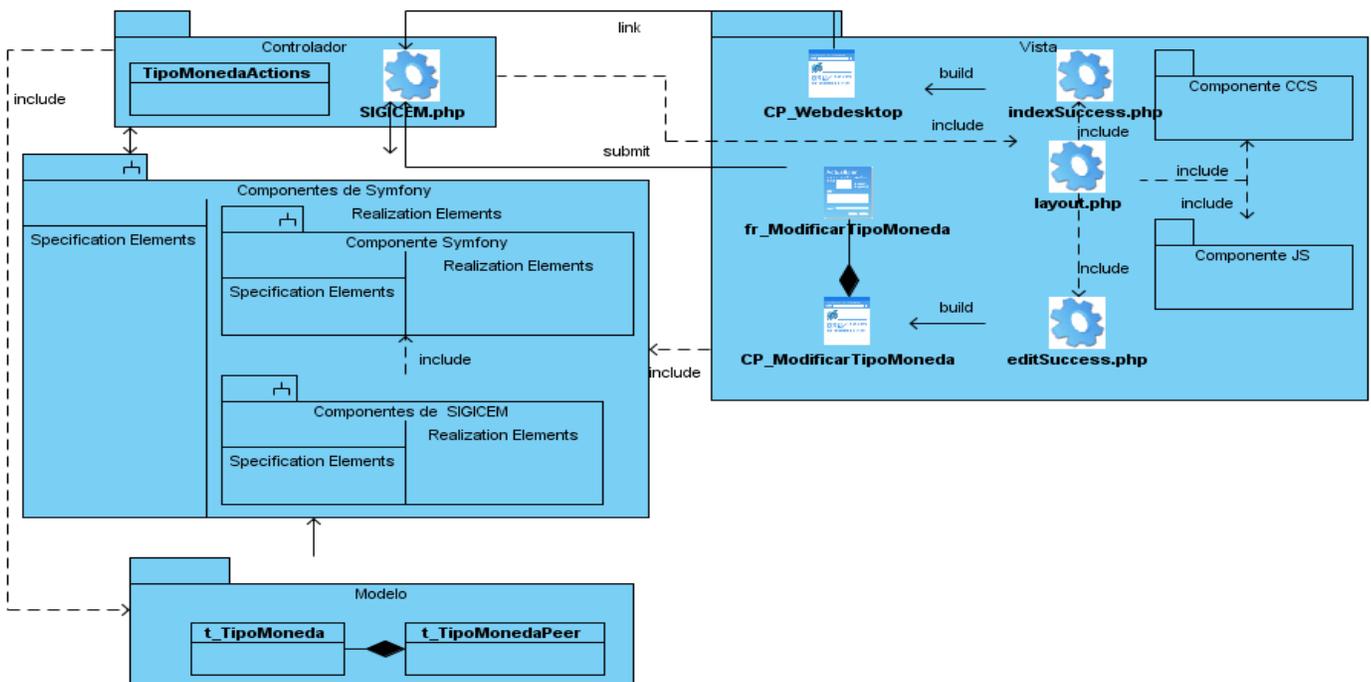


Diagrama. 24 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: *ModificarTipoMoneda*.

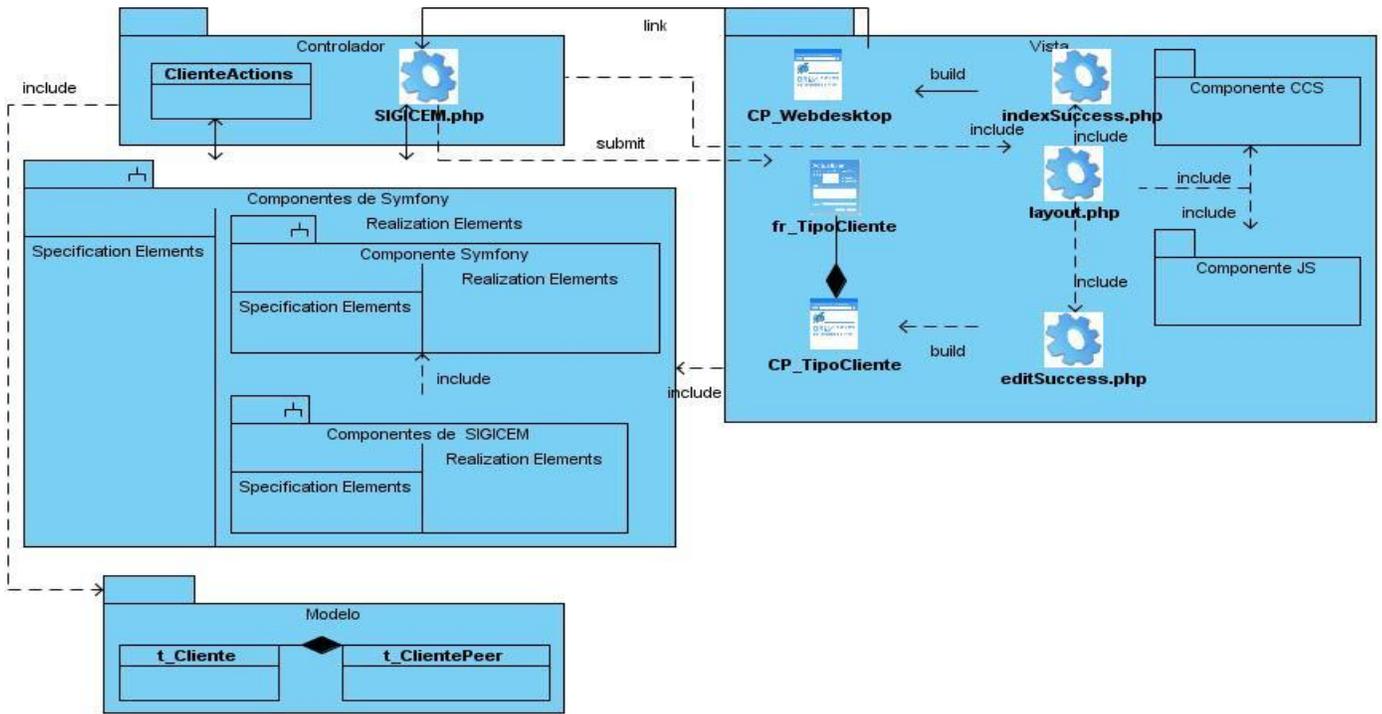


Diagrama. 25 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: ListarTipoCliente.

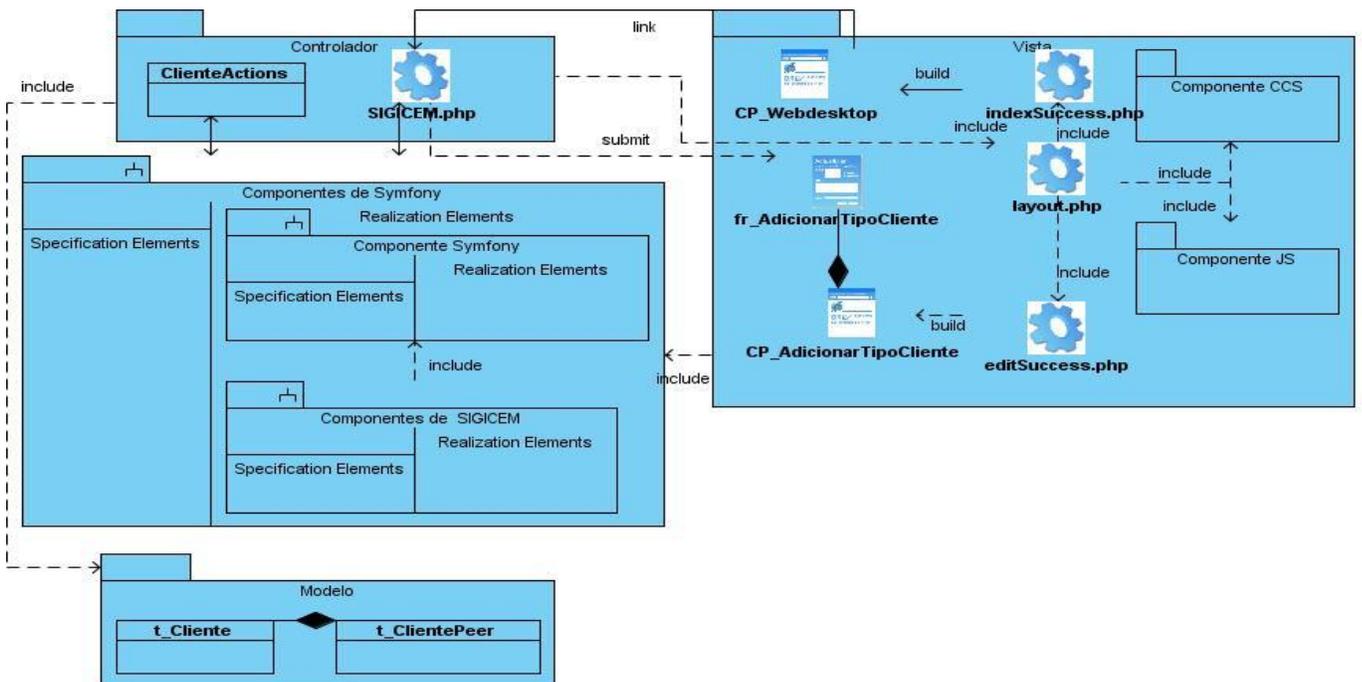


Diagrama. 26 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: AdicionarTipoCliente.

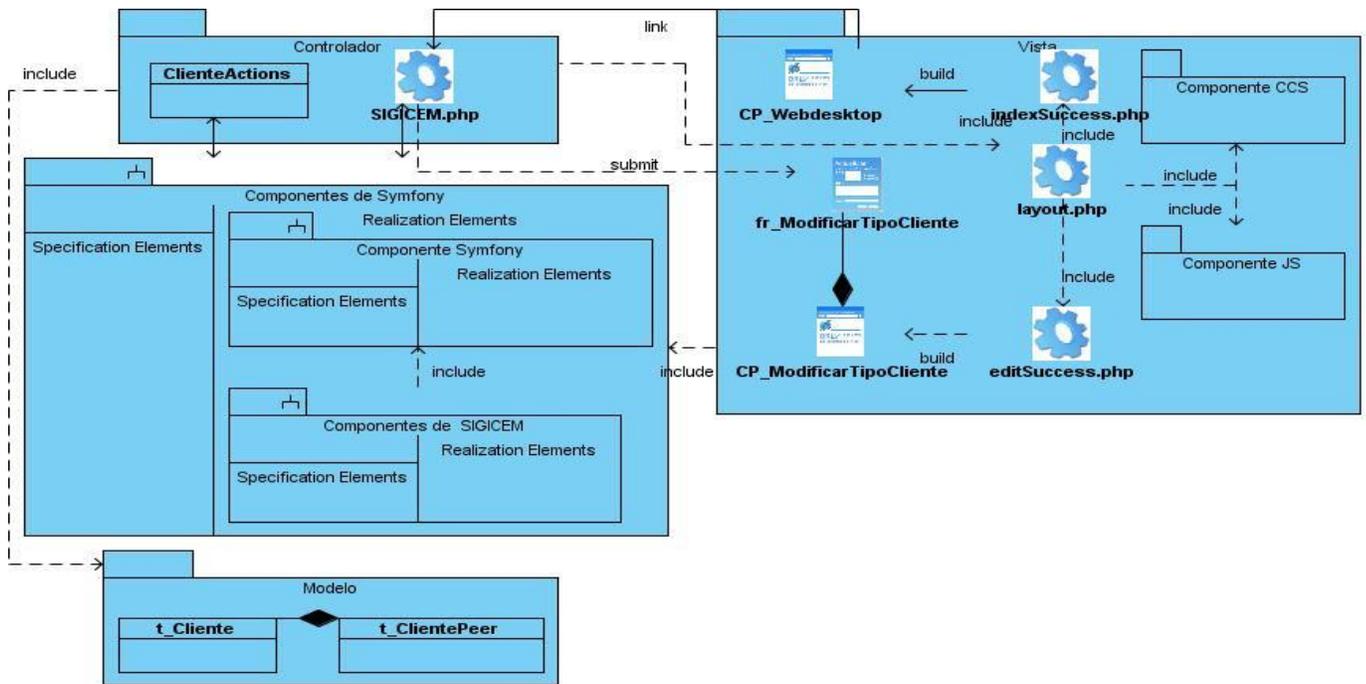


Diagrama. 27 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: *ModificarTipoCliente*.

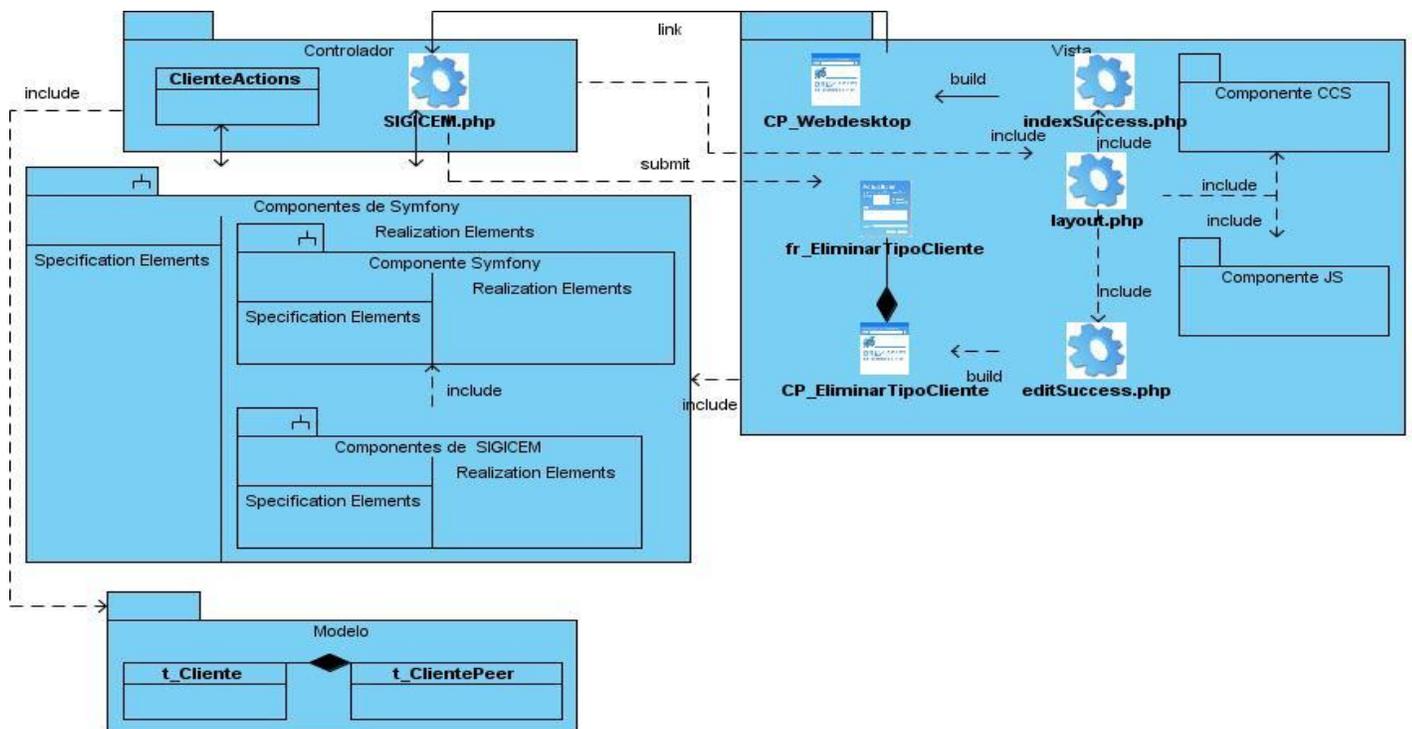


Diagrama. 28 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: *EliminarTipoCliente*.

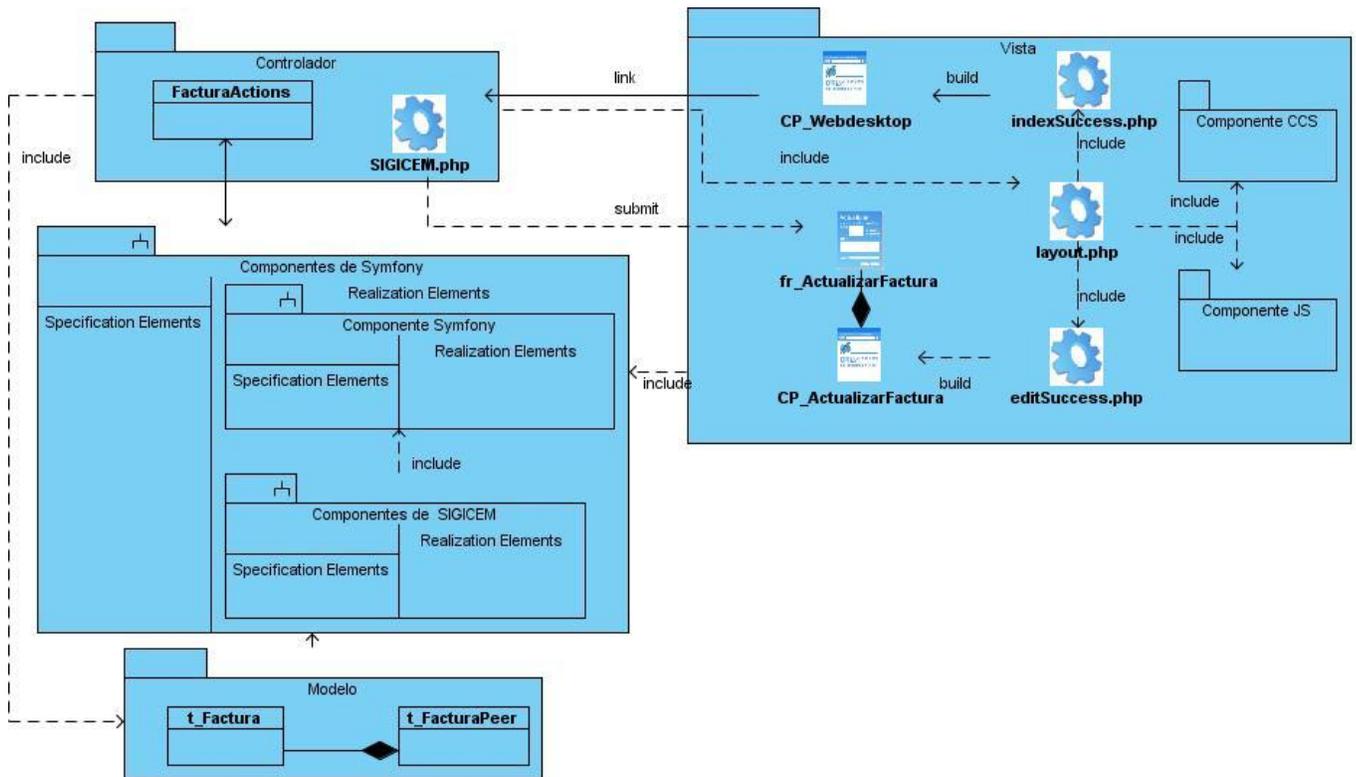


Diagrama. 29 Módulo Gestión de Clientes y Facturación de Servicios Técnicos. Diagrama de Clases del CUS: ActualizarFactura.

Anexo 2

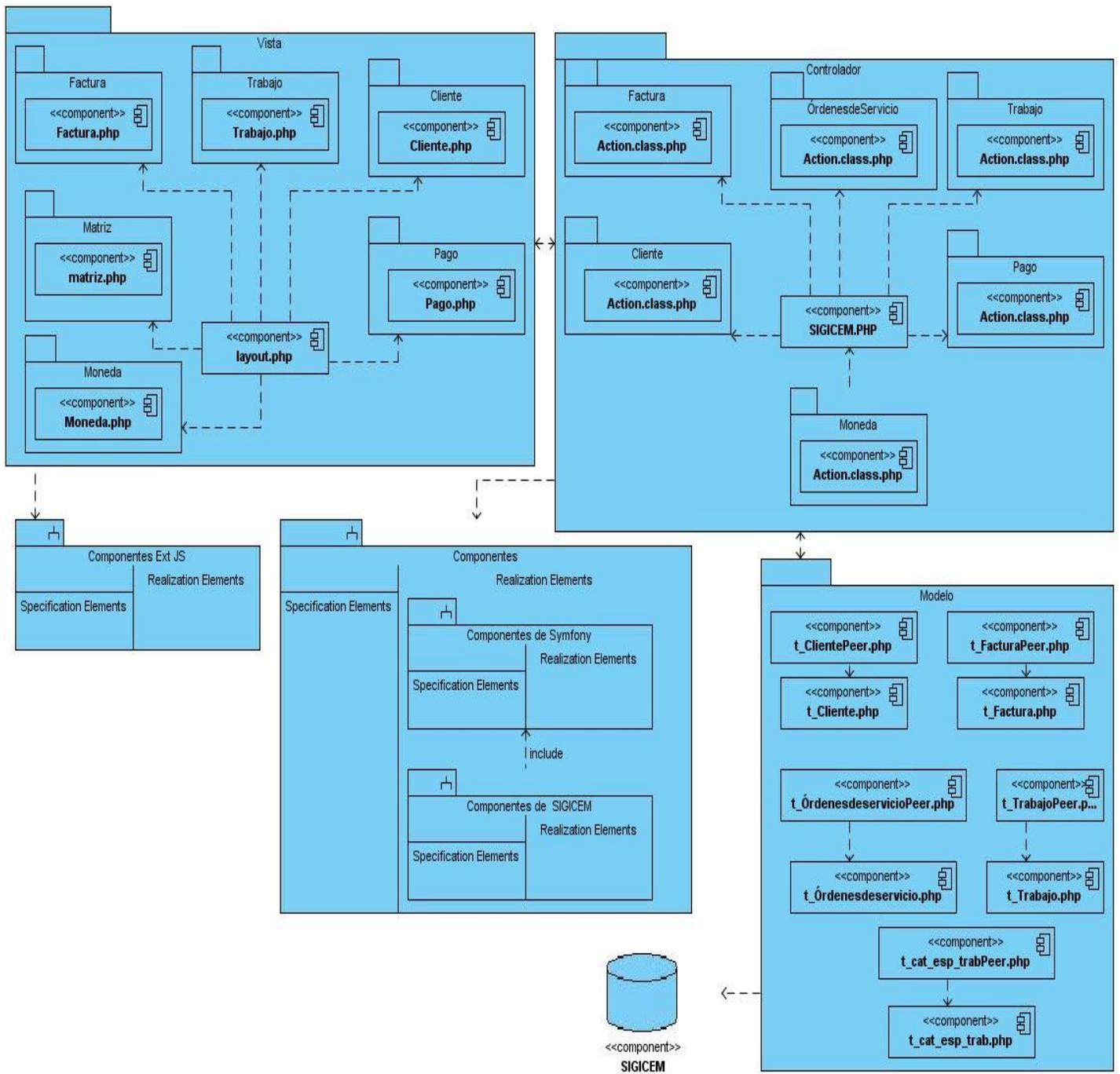


Diagrama. 30 Diagrama de componentes. Módulo Gestión de Clientes y Facturación de Servicios Técnicos.