

Universidad de las Ciencias Informáticas

Facultad 7



**Sistema para la gestión de los recursos y servicios de
los Centros de Autoaprendizaje y Servicios de
Idiomas Extranjeros**

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

Autor: Lorenzo Bárzaga Meriño

Tutora: Lic. Yaima Margarita Riveri Ruiz

Co-tutora: Ing. Yislenys Suárez Hernández

La Habana, junio de 2011

“Año 53 de la Revolución”

Datos del contacto

Tutora: Lic. Yaima Margarita Riveri Ruiz

Profesora Asistente. Licenciada en Educación, con especialidad en Lengua Inglesa. Graduada en el año 2003. Actualmente imparte la asignatura Idioma Extranjero V en el cuarto año de la carrera y es Asesora de la Lengua Materna en la facultad.

Correo electrónico: maggie@uci.cu

Co-tutora: Ing. Yislenys Suárez Hernández

Ingeniera en Ciencias Informáticas, graduada en el 2009. Estuvo al frente del equipo de Auditorías y Revisiones en el grupo de calidad del CESIM. Actualmente se desempeña como analista en los proyectos SIGICEM y SAGCEM del departamento Sistemas de Apoyo a la Salud.

Correo electrónico: ysuarezh@uci.cu.

Agradezco a mis padres por haberme dado la vida, por ser mi sostén en todo momento y por darme su ejemplo en todo.

A mi hermano Laly por ser mi compañero y amigo en todas las circunstancias que nos hemos encontrado.

A mi novia por apoyarme en cada momento y por darme todo su amor y cariño incondicionalmente.

A mis amigos Yasma, Migue... por ser como mis hermanos en todo momento y a todos los demás que estudiaron conmigo por estar siempre cuando me han hecho falta.

A mis compañeros de primer año y a todos los que conocí a lo largo del tiempo de estudio, así como a mis compañeros de mi grupo por brindarme su amistad.

A mi compañera de tesis Mileidy por brindarme su ayuda incondicional y apoyo en el desarrollo de este trabajo.

A Yanelis por ayudarme a corregirme los errores del documento de tesis y guiarme por el buen camino.

A las tutoras Yaima y Yislenis por ayudarme en el desarrollo de la tesis.

A Mojena por su ayuda en el desarrollo del sistema.

A mi suegra por brindarme su cariño como si fuera uno más de sus hijos en todo este tiempo.

A la Revolución por permitirme estudiar en esta escuela y formarme como ingeniero en Ciencias Informáticas.

A todas las personas que me han ayudado de alguna forma en la realización de este sueño.

A todos, muchas gracias.

Loren.

Dedico este trabajo de diploma en primer lugar a mis padres; por demostrarme la importancia de hacerse un profesional y brindarme todo su apoyo en los años de estudio y en todos los momentos de la vida.

A mi novia Yanisleydis por darme su apoyo incondicional y soportarme en todo este tiempo junto.

A mi hermano por estar guiándome siempre y estimulándome a que estudie.

Loren.

Resumen

El presente trabajo surge como resultado de un estudio de la gestión de información de los recursos de los Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros (CASIE). Se evidenció que el trabajo en estos centros, se realiza de forma manual o semi-automatizada, lo que trae consigo diferentes problemas tal como la demora en el procesamiento de los datos. Dada esta situación surgió la necesidad de crear un sistema que facilite la gestión de los recursos materiales y humanos de los Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros.

Para el desarrollo del sistema se utilizó Proceso Unificado de Desarrollo (RUP) como metodología de desarrollo, como herramienta para el modelado se empleó Visual Paradigm 6.4. Se utilizó el framework Symfony 1.3.3, basado en el lenguaje de programación PHP 5 y como sistema gestor de base de datos MySQL 5.0.45. Para el diseño de las interfaces de la aplicación se utilizó Ext JS 2.2.

El Sistema para la gestión de los recursos de los Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros que propone este trabajo, permitirá agilizar la elaboración y entrega de los reportes semanales y los profesores realizarán reservaciones *en línea* para hacer uso de los salones de los CASIE. Además, se informatizará el control de los recursos materiales y humanos de los CASIE, lo que garantiza la integridad de la información y hace su gestión más eficiente.

Palabras claves: CASIE (Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros); recursos materiales y humanos.

Tabla de contenidos:

Introducción 1

Capítulo 1: Fundamentación Teórica..... 5

 1.1 Sistemas para la gestión de recursos..... 5

 1.2 Estudio de la Tecnología..... 9

Capítulo 2: Características del Sistema..... 24

 2.1 Procesos del negocio 24

 2.2 Objeto de informatización..... 25

 2.3 Modelo del negocio..... 25

 2.4 Especificación de los requisitos de software..... 31

 2.5 Modelo del Sistema 34

Capítulo 3: Análisis y Diseño del Sistema 38

 3.1 Modelo de análisis 38

 3.2 Modelo del diseño 40

 3.4 Diagramas de Clases del Diseño 43

 3.5 Modelo de datos 46

 3.6 Diagrama de despliegue..... 50

Capítulo 4: Implementación 52

 4.1 Modelo de implementación..... 52

 4.2 Seguridad..... 54

Conclusiones 55

Recomendaciones 56

Referencias Bibliográficas..... 57

Bibliografía..... 60

Anexos: 62

 Anexo 1: Especificación de los casos de uso del sistema 62

 Anexo 2: Descripción ampliada de los casos de uso del sistema..... 62

 Anexo 3: Diagramas de clases del análisis 70

 Anexo 4: Diagramas de clases del diseño 71

 Anexo 5: Diagrama de clases persistentes 72

Introducción

En la actualidad es aconsejable conocer al menos un idioma extranjero, no solo como un medio de enriquecimiento cultural, sino por la razón práctica de ponerse en contacto con la actividad social y económica-científica de otras culturas. La disciplina¹ Idioma Extranjero forma parte del plan de estudios de los educandos que se forman como futuros ingenieros en Ciencias Informáticas en la UCI (Universidad de las Ciencias Informáticas). Esta disciplina es de gran importancia para la formación del futuro profesional y su desempeño en esta rama.

Con el objetivo de favorecer el estudio y conocimiento de otros idiomas, se concibieron en la UCI los Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros (CASIE). Estos centros constituyen un recurso importante para apoyar el desarrollo de estrategias que propicien el aprendizaje autónomo, a través de las experiencias educativas de las diferentes asignaturas de la disciplina antes mencionada. Los CASIE también contribuyen al estudio de idiomas extranjeros por parte del claustro de profesores.

Los centros de autoaprendizaje surgieron en el curso 2002-2003 cuando se decidió, por el Departamento de Idiomas, presentar este proyecto. Actualmente la UCI cuenta con cuatro CASIE, distribuidos en los diferentes edificios docentes de la Universidad. Cada uno de ellos tiene tres salones: de lectura, de computación y audio-visual.

Los CASIE constituyen el lugar donde los usuarios, de acuerdo con sus necesidades, preferencias, estilos de aprendizaje y disponibilidad de tiempo, pueden mantener, ampliar y/o profundizar el conocimiento del idioma. Además, permiten a los usuarios practicar aspectos que les resulten más interesantes o difíciles, tomar conciencia de sus estrategias y aprender otras nuevas. Así como trabajar individualmente a su ritmo, en el horario de su conveniencia y complementar las actividades de clase. Para lograrlo, se brindan servicios personalizados con la

¹ El término **disciplina** utilizado se corresponde con el concepto empleado en la Educación Superior en Cuba, para la parte del diseño general del currículo de una profesión determinada, en la que se planifica y organiza el contenido de una rama del saber.

supervisión directa de un profesor-asesor y el aprovechamiento de los distintos recursos didácticos, físicos y digitales. Entre ellos se encuentran:

- ✓ Servicios de diagnóstico (necesidades, estilos y estrategias de aprendizaje, niveles de conocimiento).
- ✓ Servicios de asesoría (presencial).
- ✓ Servicios de preparación para exámenes internacionales (TOEFL, IELTS).
- ✓ Servicios de copias de CDs, video y audio casetes.
- ✓ Servicios de traducción.

En cada uno de estos centros existe un técnico general, que es el responsable de los materiales y servicios que se ofrecen. Esta persona es la encargada de llevar el control de los usuarios, grupos docentes y asesores, así como de realizar inventarios a los materiales y medios cada cierto tiempo. También registra las reservaciones realizadas por los profesores para hacer uso de cada uno de los salones de los CASIE.

El trabajo que se realiza en los centros de autoacceso, así como la información que se procesa en ellos, se hace de forma manual o semi-automatizada. La información se recopila por parte del técnico general en varios documentos de tipo Excel. Tanto el tipo de documento donde se recoge la información como su cúmulo, demora el procesamiento de los datos y hace engorrosa la labor de ese trabajador.

La documentación generada es archivada en la computadora del técnico general, con el objetivo de elaborar los reportes semanales que deben ser enviados al Jefe de Departamento Docente Central. Dada la magnitud de esta documentación se dificulta la localización de determinados documentos, que también están expuestos al peligro de extraviarse ante fallas inesperadas de la computadora. Conjuntamente, no existe una política de resguardo que garantice una adecuada recuperación en caso de desastre.

Dada la situación problemática anteriormente planteada se tiene como **problema a resolver** ¿Cómo optimizar la gestión de la información de los recursos materiales y humanos y servicios de los Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros?

La investigación tiene como **objeto de estudio** el proceso de gestión de la información de los Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros; se delimita como **campo de acción** el proceso de gestión de la información de los recursos materiales y humanos y servicios de los Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros.

Para la solución del problema se plantea como **objetivo general** optimizar la gestión de la información de los recursos materiales y humanos y servicios de los Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros a través del desarrollo de un sistema informático.

Para darle cumplimiento al objetivo general se definieron las siguientes **tareas de la investigación**:

- ✓ Analizar los procesos de gestión y control de la información de los recursos de los CASIE.
- ✓ Realizar un estudio de los sistemas informáticos que gestionan información de recursos materiales y humanos en los ámbitos nacional e internacional.
- ✓ Seleccionar la tecnología a utilizar para el desarrollo del sistema.
- ✓ Diseñar los principales artefactos que propone la metodología seleccionada.
- ✓ Implementar las funcionalidades correspondientes al sistema para la gestión de los recursos y servicios de los CASIE.

Una vez culminado el *Sistema para la gestión de los recursos de los Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros* y se encuentre a disposición de los clientes, se espera como resultado que se obtengan los siguientes beneficios:

- ✓ Gestionar los usuarios que visitan los CASIE.
- ✓ Generar los reportes semanales que contemplen los reportes de incidencias tecnológicas y los reportes de asistencia.
- ✓ Permitir a los profesores realizar en línea, las reservaciones, de los salones de los CASIE.
- ✓ Gestionar los materiales digitales y en formato duro existentes en los CASIE.
- ✓ Controlar los inventarios realizados a los medios y materiales de los CASIE.

El presente documento está estructurado en cuatro capítulos. A continuación se expone una breve descripción:

Capítulo 1: Fundamentación Teórica

Se recogen los principales conceptos que serán tratados durante el desarrollo de la investigación. Se hace referencia a las tendencias actuales, así como las tecnologías, herramientas y la metodología que se aplican durante el proceso de investigación y desarrollo del sistema.

Capítulo 2: Características del Sistema

Se describe el funcionamiento del negocio a través del diagrama de casos de uso del negocio y la descripción textual de cada caso de uso. Se definen además las reglas del negocio, los requerimientos funcionales y no funcionales del sistema, a través del diagrama de casos de uso del sistema y las descripciones de los casos de uso para comprender mejor su funcionamiento.

Capítulo 3: Análisis y Diseño del Sistema

Está relacionado con el diseño del sistema; proporciona las clases del diseño; así como los diagramas correspondientes a cada una de ellas. Se construye además el modelo del diseño para aplicaciones web, lo cual constituye una ayuda para la implementación del sistema y brindar los recursos para la base de datos. Se describe el diagrama de despliegue y el modelo de datos.

Capítulo 4: Implementación

Está enfocado a la fase de implementación para dar solución a los requerimientos funcionales identificados. Se modela los diagramas de componentes; se describen los estándares, codificación y el tratamiento de errores para darle solución al sistema propuesto.

Capítulo 1: Fundamentación Teórica

En el presente capítulo se realiza un análisis sobre diferentes sistemas que permiten la gestión de recursos materiales y humanos en el ámbito nacional e internacional que pueden ser de ayuda para el desarrollo del sistema que se desea implementar.

Se muestra también un estudio de las tendencias, tecnologías y metodologías más usadas en la actualidad, así como las herramientas a utilizar para el desarrollo del Sistema para la gestión de los recursos y servicios de los CASIE en la Universidad de las Ciencias Informáticas.

1.1 Sistemas para la gestión de recursos

Llevar el control de los recursos de cualquier institución es de gran importancia para garantizar su preservación. A continuación se muestra un estudio realizado a sistemas utilizados para el control de recursos materiales y humanos que puede ser de ayuda para la implementación de la solución propuesta.

1.1.1 Sistemas para el control de recursos materiales

Sistema Computarizado para el Manejo de Inventario y Control de Pedidos en una Industria de Elaboración de Perfume

Este sistema se desarrolló en la Universidad de San Carlos en Guatemala en el año 2004 como un trabajo de graduación. Busca optimizar los recursos con que una persona cuenta al trabajar, mientras se reducen tiempo y costos.

El sistema permitirá obtener información de una manera rápida y exacta al tener definido cómo se controlarán físicamente los productos en la bodega y partiendo de ello para su fácil localización en el sistema. La estructura del sistema permite consultar tanto la cantidad existente de cierto producto, como su ubicación física en los diferentes estantes a través del código. [1]

Software para la Organización Nacional de Bufetes Colectivos (ONBC)

Este sistema fue desarrollado en la provincia de Ciego de Ávila, Cuba. Tiene como principal objetivo llevar el control de todos los recursos como bombillos, materiales de oficina entre otros.

Este sistema es propio de la organización, es un sistema obsoleto y está montado sobre la base de MS-DOS.

Sistema Económico para las Unidades de Salud (MEBUS)

Su principal objetivo es llevar el control de todos los medios básicos de todas las unidades de salud, este incluye información de los submayores de los medios básicos, brinda la depreciación por cada medio y da la posibilidad de revalorizar el medio cuya depreciación haya caducado. Además, contiene una base de datos que recoge la información de todos los materiales a los cuales se les haya dado de baja en un periodo de 5 (cinco) años.

1.1.2 Sistemas de gestión de la información de recursos humanos

Empleados MSD Multiusuario

Es un avanzado gestor de información, que permite manejar datos personales, familiares, historial, documentos y titulaciones entre otros servicios. Este sistema está estructurado por cuatro módulos: Módulo Empleados, Módulo Información de Empleados, Módulo Resumen y Módulo Estadísticas.

Es una aplicación fácil de utilizar y permite configurar muchas de sus características para adaptarse a los gustos del usuario. Brinda un alto grado de privacidad y aporta herramientas de suma utilidad tales como: copias de seguridad, procesador de textos, calculadora y otras. [2]

Wikimetal RRHH

Es un sistema destinado a la gestión del personal de una empresa, de simple y fácil manejo que ha sido creada con el fin de ayudar a las empresas en la difícil tarea del control de sus empleados. Entre otras funciones; brinda la posibilidad de gestionar las vacaciones de los empleados de forma mensual, bimestral o trimestral. Puede efectuar el control por departamentos e incluso visualizar inmediatamente los retrasos, ausencias o sanciones de los empleados. [3]

@GestRRHH versión 2.0

Es un sistema diseñado para la gestión de Recursos Humanos compuesto de los siguientes módulos: agenda, plantilla, actividades y nóminas. Entre las principales funcionalidades que presenta están las siguientes: permite la realización de un gran número de informes, cada módulo posee un color individual para mejorar la información y permite la copia de seguridad automática. [4]

Registro Personal de la Salud (RPS)

Es un sistema que está basado en software libre, en las nuevas tecnologías y en el uso de servicios web que garantizan una explotación integrada y compatible. Utiliza como soporte de esta aplicación la infraestructura de la Red Telemática de Salud (INFOMED). Cumple con la arquitectura definida por el Grupo de Arquitectura MINSAP–MIC y hace uso de la Plataforma de Servicios (PlaSer).

Este sistema tiene las siguientes características:

- ✓ Almacena datos en una base de datos única, sólida y confiable.
- ✓ Realiza la actualización del Registro Personal de la Salud (profesionales y técnicos) de manera rápida y segura.
- ✓ Permite gestionar al personal de salud pendiente de ubicar en una unidad de salud; abarca los movimientos de traslado, jubilación y baja, así como los movimientos de los recién graduados.
- ✓ Busca profesionales por sus características, que puede ser: nombre, tipo de profesional y nivel técnico. [5]

Fastos

Es un sistema en el ámbito nacional que gestiona la información de los recursos humanos, conformado por módulos que ejecutan los procesos y operaciones de: registro de empleados, control de la plantilla, control de asistencia, informes y modelos. [6]

ASSETS (Sistema de Registro Integral)

Sistema que registra todo el personal que es plantilla en la UCI, este sistema es del tipo de Planificación de Recursos Empresariales (ERP), de factura foránea. Se introdujo en Cuba desde el año 1997. Controla plenamente los recursos laborales, dígame: empleados, estructura organizativa de la entidad y plantilla. Este sistema solo gestiona el personal que es plantilla del centro por lo que no sirve de ayuda a la hora de encontrarle una solución a la problemática planteada.

Sistema de Registro de Personal (SRP)

Este sistema está enmarcado en el registro del personal tercero de la UCI. Fue desarrollado a través del lenguaje de programación PHP, PostgreSQL como gestor de base de datos y el servidor web utilizado es Apache. SRP tiene varias funcionalidades como: gestionar personas, atributos por categoría, categorías de persona y roles. Esta solución -a pesar de mantenerse en función y de cierto modo realizar el registro del personal tercero específicamente, que es uno de los tipos de persona con necesidad de registro en la universidad- no es factible, porque no responde al flujo de procesos que resuelve verdaderamente el problema.

Después de realizar un análisis de los sistemas expuestos con anterioridad, se puede decir que en su generalidad no cumplen con la necesidad que se plantea en la problemática. Es decir, la información que gestionan estos sistemas, no coincide con la información que se genera en los CASIE y no manejan los mismos tipos de datos en cada uno de ellos.

El MEBUS gestiona los medios básicos de las unidades de salud; el sistema Registro Personal de la Salud solo maneja los datos del personal de la salud. Wikimetal RRHH es un sistema destinado a la gestión del personal de una empresa. Empleados MSD Multiusuario permite manejar datos personales, familiares, historial, documentos y titulaciones entre otros servicios de los empleados. GestRRHH gestiona los recursos humanos; así como las nominas de los empleados. Todos estos sistemas manejan información propia de su organización, son de software propietario y los costos de mantenimiento y soporte técnico son muy elevados.

La información que se desea gestionar en el sistema está relacionada con los recursos y servicios de los CASIE: materiales, usuarios, grupos docentes, reservaciones e inventarios realizados. El sistema a desarrollar debe ser multiplataforma (Windows o Linux).

1.2 Estudio de la Tecnología

1.2.1 Metodologías de desarrollo de software

Actualmente el desarrollo de software ha alcanzado un alto nivel debido a la competencia que existe, por lo que los desarrolladores se han visto en la necesidad de buscar técnicas mediante las cuales se logren estandarizar el trabajo de las aplicaciones que se desarrollan. Es por ello que se ha creado un conjunto de metodologías para el desarrollo del software; que hace posible que todo el personal de un proyecto se vincule y pueda entenderse.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayuda a la documentación para el desarrollo de productos software, en la que se indican, paso a paso, todas las actividades a realizar para lograr informatizar el proceso deseado; muestran qué personas deben participar en el desarrollo de las actividades y el papel que desempeñan. Además, detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla. [7]

Las metodologías se pueden agrupar en tradicionales o formales y ágiles. Las tradicionales centran su atención principalmente en desarrollar una documentación completa de los proyectos, así como el cumplimiento de un plan de proyecto, el cual debe ser definido al comienzo del mismo. Mientras que para las metodologías ágiles es de vital importancia la respuesta a los cambios y mantener una buena relación con el cliente.

Programación Extrema (XP)

XP es una de las metodologías de desarrollo de software más exitosas. Se utiliza para proyectos de corto plazo y equipo de desarrollo pequeño. Es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software; promueve el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores y propicia un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define especialmente para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. [8]

Se considera como una adopción de las mejores metodologías de desarrollo a partir de lo que se pretende realizar con el proyecto, y de manera dinámica aplicarlo durante el ciclo de vida del software.

SCRUM

Es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza ejecutable que incorpora una nueva funcionalidad. Es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

El equipo, se enfoca solamente en construir software de calidad. La gestión de proyecto se focaliza en definir las características del software a construir y remover cualquier obstáculo que pudiera entorpecer el trabajo del equipo de desarrollo. Se busca que los equipos de desarrollo, sean lo más productivos posible.

Se realizan entregas parciales y regulares del resultado final del proyecto, priorizadas por el beneficio que aportan al receptor del proyecto. Está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos y donde la innovación, la competitividad y la productividad son fundamentales.

SCRUM también se utiliza para resolver situaciones en que no se entrega al cliente lo que necesita, cuando las entregas se alargan demasiado, los costos se disparan o la calidad no es aceptable; cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar con un proceso especializado en el desarrollo del producto. [9]

Proceso Unificado de Rational (RUP)

El Proceso Unificado Rational constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos, basado íntegramente en lenguaje unificado de modelado (UML) como soporte a la metodología.

Esta es una metodología de la ingeniería de software el cual logra disciplinar el trabajo en un equipo de desarrollo pues el mismo define quién debe hacer qué, cómo y cuándo. RUP es una metodología de desarrollo de software y junto con el lenguaje unificado de modelado UML. [10]

Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo 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.

RUP pretende implementar las mejores prácticas actuales en ingeniería de software:

- ✓ Desarrollo iterativo del software.
- ✓ Administración de requerimientos.
- ✓ Uso de arquitecturas basadas en componentes.
- ✓ Modelación visual del software.
- ✓ Verificación de la calidad del software.
- ✓ Control de cambios.

Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. En lo que se refiere al ciclo de vida de RUP, es una implementación del desarrollo en espiral, este divide el proceso de desarrollo del software en ciclos, donde estos ciclos son la clave que tiene el modelo para crear un proyecto de buena calidad. El ciclo de vida organiza las tareas en fases e iteraciones.

Está compuesto por cuatro fases: Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones. A su vez está compuesto por nueve flujos de trabajo, seis de ellos de ingeniería y tres de apoyo: Modelado del negocio, Requerimientos, Análisis y

Diseño, Implementación, Pruebas, Despliegue, Administración de cambio y configuración, Administración de proyecto y Entorno. En cada una de sus fases se emplean todos los flujos de trabajo pero con diferente énfasis.

Después del análisis de las metodologías anteriormente mencionadas se decidió escoger para el proceso de desarrollo RUP ya que es iterativo e incremental, está centrado en la arquitectura y guiado por los casos de uso. Además está compuesto por cuatro fases y flujos de trabajo que permiten guiar de forma detallada el proceso de desarrollo de software.

1.2.2 Lenguajes de modelado

El lenguaje de modelación es independiente del lenguaje de programación que se desea emplear en el desarrollo del software. Con el transcurso de los años se ha producido un auge en la programación orientada a objeto (POO), lo que ha conllevado a la creación de diferentes lenguajes de modelado orientado a objetos.

Lenguaje de modelado OO i*

La notación i* fue creada en la primera mitad de la década de los 90. Permite expresar de forma clara y sencilla los objetivos de los actores que aparecen en los modelos y la dependencia entre ellos. Consta con una notación gráfica que permite tener una visión intuitiva y unificada del entorno modelado que muestra tales actores y dependencias.

Tiene la desventaja de no tener una definición única del lenguaje. Además, las definiciones existentes no son tan claras como se desearía ya que contienen ambigüedades y contradicciones. [11]

Lenguaje de modelado OO UML

Es un lenguaje de modelado visual que se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema de software. [12]

UML, (por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y

documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo). Incluye aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.[13]

Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. UML 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 también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo. [14]

Después del estudio realizado a los diferentes lenguajes de modelado se propone usar UML en su versión 2.1, pues 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.

1.2.3 Herramientas de modelado

Las herramientas de modelado o CASE se pueden definir como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

Rational Rose

Rational Rose es una herramienta de modelado visual con plataforma independiente, que permite la comunicación entre los miembros del equipo. Tiene la ventaja que utiliza la notación estándar en la arquitectura de software (UML), posibilita a los arquitectos y desarrolladores de software visualizar todo el sistema mediante el uso de un lenguaje común. Mantiene la consistencia de los modelos de sistemas de software.

Permite chequear las sintaxis UML y realizar ingeniería inversa. Posibilita la generación de documentación automáticamente y la generación de código a partir de los modelos. A través de él los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con los demás componentes del proyecto. [15]

Visual Paradigm

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: Análisis y Diseño orientados a objetos, Construcción, Pruebas y Despliegue.

El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. [16]

Posibilita la representación gráfica de los diagramas. Esto propicia ver el sistema desde diferentes perspectivas, como el de componentes, despliegue, secuencia, casos de uso, clase, actividad, estado, entre otros. Además, identifica requisitos y comunica información. Se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, además, permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo a través de un lenguaje gráfico. Tiene disponible, distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community. Facilita licencias especiales para fines académicos.

Entre sus características más significativas se puede resaltar que su licencia es gratuita, posee varios idiomas, sus ediciones son compatibles y es de fácil uso para la creación de aplicaciones web. [17]

Después del análisis realizado a las diferentes herramientas de modelado se escogió Visual Paradigm UML en su versión 6.4. La misma soporta el ciclo de vida completo del desarrollo de software y es de fácil uso para la creación de aplicaciones web.

1.2.4 Lenguajes de programación

En la actualidad existen diferentes lenguajes de programación, estos han surgido debido a las tendencias y necesidades de las plataformas existentes. A continuación se hará un análisis de las más importantes.

Desde el comienzo de Internet, surgieron diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. Con el transcurso del tiempo, las tecnologías se desarrollaron y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la web dinámica, que permitieran interactuar con los usuarios y utilizaran sistemas de bases de datos.

ASP.NET

ASP.NET es un conjunto de tecnologías de desarrollo de aplicaciones Web comercializado por Microsoft. Es usado por programadores para construir sitios Web domésticos, aplicaciones Web y servicios XML. Forma parte de la plataforma .NET y es la tecnología sucesora de Active Server Pages (ASP).

ASP.NET es una nueva estructura de programación y fue diseñado con el objetivo de que dichas aplicaciones respondan rápidamente a las solicitudes de los usuarios, sin importar la cantidad de datos que se estén procesando en el servidor, debido a que en una página ASP se pueden incluir: HTML plano, código de scripting y texto; no hay una distinción formal entre el contenido de una página y su comportamiento. Impone un cierto orden sobre el modelo de programación estándar ASP y proporciona diversas mejoras en las cuales se destacan:

- ✓ **Rendimiento:** se compila desde el código nativo, lo que permite mucho mejor rendimiento y un almacenamiento de la caché en el servidor.
- ✓ **Rapidez en programación:** mediante diversos controles, se logra con unas pocas líneas y en menos de 5 (cinco) minutos, mostrar toda una base de datos y hacer rutinas complejas.
- ✓ **Servicios Web:** posee herramientas para compartir datos e información entre distintos sitios.
- ✓ **Seguridad:** posee diversas herramientas que garantizan la seguridad de nuestras aplicaciones. [18]

PHP

Es un lenguaje de programación del lado del servidor, tiene como característica que es fácil de aprender, y está pensado para desarrollar páginas web eficientes. Se utiliza para escribir el contenido dinámico de las páginas. Soportado por la mayoría de los servidores web entre ellos

Apache. Se utiliza en cualquier sistema operativo. Permite manipular la información almacenada en la base de datos como MySQL. [19]

Es gratuito e independiente de plataforma, rápido, con una amplia librería de funciones y mucha documentación. Fue creado en 1994 por Rasmus Lerdorf.

Es un lenguaje orientado a objeto, lo que permite la reutilización de código y un rápido desarrollo. Posee compatibilidad con las bases de datos más comunes, como MySQL, Oracle y PostgreSQL, entre otros. Proporciona soporte para diferentes protocolos de comunicación conocidos entre los cuales se tienen:

- ✓ HTTP (Protocolo de Transferencia de Hipertexto).
- ✓ IMAP (Protocolo de Acceso a Mensajes de Internet).
- ✓ FTP (Protocolo de Transferencia de Archivos).
- ✓ LDAP (Protocolo Ligero de Acceso a Directorios), entre otros.

Es publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Java

Java es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido. Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma, esto en alguna medida es una ventaja significativa para los desarrolladores de software pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux y Apple.

Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente. Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador, como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria. [20]

Después de analizar los lenguajes de programación anteriormente mencionados se escogió PHP, pues es un lenguaje liviano para desarrollo de aplicaciones Web, uno de los más utilizados en

el mundo, presenta una amplia documentación, es gratuito e independiente de la plataforma donde se quiera utilizar.

1.2.5 Framework de desarrollo

Un framework de desarrollo es una estructura de trabajo y soporte definida, mediante la cual otro software puede utilizar para llevar un trabajo más organizado y desarrollado en la realización de software. Con los framework de desarrollo, se pueden encontrar soportes de programas, bibliotecas y un lenguaje interpretado, entre otros software, para facilitar el desarrollo y unir las diferentes estructuras y componentes de un proyecto.

Symfony

Es un framework para aplicaciones web desarrollado en PHP5 por Fabien Potencier y patrocinado por los laboratorios de Sensio, Agencia francesa Web muy conocida por sus puntos de vista innovador en el desarrollo web. Acelera la creación y mantenimiento de las aplicaciones Web, y permite la reutilización de código, lo cual agiliza el desarrollo de la aplicación.

Su diseño permite optimizar las aplicaciones y mantener una organización de la misma. Permite reducir el tiempo de desarrollo de una aplicación pues proporciona varias herramientas y clases las cuales pueden ser reutilizadas por los desarrolladores. Permite el desarrollo por capas, ya que se basa en el patrón arquitectónico Modelo Vista Controlador (MVC). Es compatible con la mayoría de los gestores de base de datos como MySQL. [21]

Kumbia

Es un framework para desarrollo de aplicaciones Web escrito en PHP5. Se basa en las prácticas de desarrollo Web como “Don't Repeat Yourself” (DRY) y el Principio “Keep It Simple, Stupid” (KISS). Fomenta la eficiencia y velocidad en la creación y mantenimiento de aplicaciones Web mediante el remplazo de códigos repetitivos. Intenta proporcionar un medio fácil para el desarrollo de aplicaciones robustas para entornos educativos y comerciales, gracias a que es flexible y configurable. [22]

CakePHP

Es un framework para aplicaciones Web desarrollado en PHP. Empezó su desarrollo en el 2005 cuando Ruby On Rails² estaba ganando popularidad y desde entonces ha generado muchos proyectos y su comunidad ha crecido. Facilita el usuario la interacción con las Bases de Datos mediante el uso de Active Record, se basa en el patrón arquitectónico Modelo Vista Controlador (MVC), lo que le permite dividir la aplicación en capas.

Entre sus características fundamentales se encuentran:

- ✓ Es compatible con PHP4 y PHP5.
- ✓ Presenta URLs amigables.
- ✓ Sistema de plantillas rápido y flexible.
- ✓ Trabaja en cualquier subdirectorio del sitio.
- ✓ Validación integrada.
- ✓ Componentes de seguridad y sesión. [23]

Después de analizar los Framework existentes para PHP, se selecciona Symfony en su versión 1.3.3, por ser un framework gratuito y de código abierto y por poseer una gran variedad de plug-ing, que permite agilizar el trabajo. Es compatible con el Sistema Gestor de Base de Datos (SGBD) que se va a utilizar. Symfony es independiente del SGBD, prácticamente debido a que internamente usa Propel o Doctrine. Además tiene una amplia comunidad a nivel mundial y mucha documentación en inglés y español.

1.2.6 Herramientas de desarrollo

Para desarrollar, normalmente solo es necesario un editor de texto, un intérprete o compilador y una terminal de líneas de comando, pero siempre es más rápido y fácil si se usan algunas herramientas o un Entorno de Desarrollo Integrado (IDE). Esto simplifica el trabajo y ahorra tiempo de desarrollo.

NetBeans

² **Ruby On Rails:** Es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC).

NetBeans IDE es 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. Es un producto libre y gratuito sin restricciones de uso.

Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles, resaltado de sintaxis y por si fuera poco sus funcionalidades son ampliables mediante la instalación de [24].

Entre las características de la plataforma están:

- ✓ Administración de las interfaces de usuario (ejemplo: menús y barras de herramientas).
- ✓ Administración de las configuraciones del usuario.
- ✓ Administración del almacenamiento (guarda y carga cualquier tipo de dato).
- ✓ Administración de ventanas.
- ✓ Framework basado en asistentes (diálogos pasos a paso). [25]

Zend Studio

Zend Studio es uno de los ambientes de desarrollo integrado, disponible para desarrolladores profesionales que agrupa todos los componentes de desarrollo necesarios para el ciclo de desarrollo de aplicaciones PHP. Zend Studio acelera los ciclos de desarrollo y simplifica los proyectos complejos. Proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

Consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. La del cliente contiene la interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP, o en caso de que estén instalados, los configura para trabajar juntos en depuración. [26]

Después del análisis de los diferentes entornos de desarrollo integrados mencionados anteriormente se escogió NetBeans IDE, para llevar a cabo el desarrollo en PHP, en su versión 6.8,

ya que esta versión incorpora un módulo para el trabajo con Symfony, muy amigable y fácil de utilizar.

1.2.7 Sistemas Gestores de Base de Datos

Una base de datos o banco 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.

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 (acceso a usuarios autorizados).
- ✓ Protección de información, de modificaciones, inclusiones, consulta. [27]

PostgreSQL

Es un sistema de gestión de base de datos relacional orientada a objetos, de software libre, publicado bajo la licencia BSD (Berkeley Software Distribution), con características de los mejores sistemas de bases de datos comerciales. PostgreSQL es libre y su código fuente completo está disponible. [28]

Es un motor de base de datos surgido en 1986 con el lanzamiento de su primera versión. Este gestor es altamente potente y posee prestaciones y funcionalidades equivalentes a otros de carácter comercial. Es más completo que MySQL ya que permite métodos almacenados, restricciones de integridad y vistas.

Algunas de sus características principales:

- ✓ Alta concurrencia.
- ✓ Amplia variedad de tipos nativos.

- ✓ Uso de disparadores.
- ✓ Funciones de Ventanas.
- ✓ Expresiones de tablas comunes y consultas recursivas.
- ✓ Instalaciones ilimitadas.
- ✓ Estabilidad y confiabilidad.
- ✓ Extensible.
- ✓ Multiplataforma.
- ✓ Diseño para ambientes de amplio volumen.

MySQL

MySQL es un SGBD SQL de código abierto, lo que significa que cualquier programador puede remodelar el código de la aplicación para mejorarlo. Existen versiones de MySQL para Linux y Windows, esto lo hace que sea multiplataforma. [29]

MySQL es multihilo y multiusuario licenciado bajo la GNU GPL. MySQL fue creado por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Es una popular herramienta de código abierto del mundo de software de base de datos, con más de 100 millones de copias de su software descargado o distribuidos a lo largo de su historia. Con su velocidad, fiabilidad y facilidad de uso. Se ha convertido en la opción preferida para la Web, Web 2.0, SaaS y empresas de telecomunicaciones, debido a que elimina los problemas más importantes asociados con el tiempo de inactividad, mantenimiento y administración y de aplicaciones en línea.

Las principales características de este gestor de bases de datos son las siguientes:

- ✓ Aprovecha la potencia de sistemas multiprocesador a la hora de realiza las búsquedas de datos.
- ✓ Soporta gran cantidad de tipos de datos para las columnas.
- ✓ Gran portabilidad entre sistemas.
- ✓ Soporta hasta 32 índices por tabla.

- ✓ Posee API's³ que permiten el desarrollo en gran cantidad de lenguajes (C, C++, Java, PHP, entre otros.).
- ✓ Gestión de usuarios y contraseñas, que mantiene buen nivel de seguridad en los datos. [30]

Realizado el estudio de los gestores de bases de datos anteriores se escogió para el desarrollo del sistema MySQL en su versión 5.0.45, pues es muy sencillo de utilizar, se integra perfectamente con el framework Symfony y es una de las más utilizadas en sistemas que no requieran de mucho procesamiento de información por parte del gestor de bases de datos.

Para el desarrollo del sistema se propone como servidor web Apache en su versión 2.2 ya que es multiplataforma; es modular pues puede ser adaptado a diferentes entornos y necesidades, es extensible. Es también el servidor web más utilizado hoy en día y acapara casi el 60 % de la cuota del mercado.

Para el diseño de las interfaces del sistema se propone Ext JS en su versión 2.2 por ser un framework JavaScript del lado del cliente, que permite crear aplicaciones web con interfaces muy similares a la de una aplicación de escritorio. Entre sus características se pueden encontrar que es multiplataforma, completamente orientado a objetos, y posee múltiples posibilidades para el trabajo con las validaciones y manejo de errores en el cliente.

En el capítulo se realiza un análisis sobre la forma en que se gestiona y controla la información de los recursos de los CASIE, el cual evidenció que el trabajo realizado en estos centros se ejecuta de forma manual o semi-automatizada lo que provoca la existencia de diferentes problemas. Las aplicaciones estudiadas que permiten la gestión de los recursos materiales y humanos a los niveles internacional, nacional e institucional, no satisfacen las necesidades que se presentan en los Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros. Esto se sustenta en que la información que gestionan estos sistemas no coincide con la que se genera en los CASIE y con los datos que se desea procesar en el sistema propuesto.

³ API por sus siglas en inglés (Application Programming Interface) o **interfaz de programación de aplicaciones** es conjunto de funciones y procedimientos (métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usados generalmente en las bibliotecas.

Se plantea realizar un sistema para la gestión de los recursos de los Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros para el cual se seleccionó: como metodología de desarrollo Proceso Unificado de Desarrollo (RUP). Como lenguaje de modelado se propone usar UML 2.1, para el modelado UML, se empleó Visual Paradigm 6.4. También se escogió el framework Symfony 1.3.3, basado en el lenguaje de programación PHP 5.2.3 y como sistema gestor de base de datos MySQL 5.0.45. Para el diseño de las interfaces de la aplicación se utilizó Ext JS 2.2 y como entorno de desarrollo integrado (IDE) se seleccionó NetBeans IDE. 6.8.

Capítulo 2: Características del Sistema

En el presente capítulo se especifican las reglas generales que se deben cumplir en el negocio, el modelado del negocio, para lo cual se identifican y describen cada uno de los actores y trabajadores. Se plantean los casos de uso del negocio, los Diagramas de Actividad y el Modelo de objetos. Se definen los requisitos funcionales y no funcionales del sistema a desarrollar. Se detallan cada uno de los casos de uso del sistema.

2.1 Procesos del negocio

Thompson, R.J y Redstone, L en 1997 definen un proceso como una serie de actividades interconectadas en busca de un propósito. En un contexto de negocios, el propósito de los procesos será proveer a sus clientes internos con sus requerimientos de una manera oportuna. [31]

2.1.1 Descripción de los Procesos del negocio

En las visitas realizadas a los Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros (CASIE), se pudieron identificar como procesos fundamentales: proceso de atención a los diferentes usuarios que visitan los CASIE, el proceso de reservaciones de los salones y el proceso de la realización de los inventarios de estos centros.

Para llevar a cabo el primer proceso, cada uno de estos centros cuentan con una recepcionista (también puede ser llevado a cabo por el técnico general), la cual registra los datos de todos los usuarios que diariamente visitan los CASIE. El llenado de los datos se realiza de forma manual, actividad muy difícil de cumplir debido al volumen de información a procesar.

En el segundo proceso, en los CASIE se llevan un registro de todas las reservaciones realizadas por los profesores para hacer uso de alguno de los salones. El llenado del listado de reservaciones se realiza de forma manual, lo que resulta muy complejo debido al gran volumen de información que se almacena. Todo este trabajo ocasiona duplicación de datos, que exista algún error a la hora de llenar los informes o que ocurra su pérdida.

El proceso de inventarios es realizado por el técnico general. Este se realiza cada vez que el centro cierra sus servicios por un periodo determinado y cuando se realiza la apertura o cierre del

mismo. El técnico general debe detallar cada uno de los medios que tiene en el CASIE y verificar los datos de estos. Todos los datos se registran en el modelo de medios básicos.

2.2 Objeto de informatización

El sistema automatizará el registro de los datos de los usuarios (estudiantes, alumnos ayudantes, profesores y trabajadores) que diariamente visitan los CASIE. Esta información podrá ser insertada y modificada por el técnico general de cada uno de los centros. Los profesores podrán realizar la reservación en línea para hacer uso de alguno de los salones. Así como también registrar y mantener actualizados los materiales y recursos que se ofrecen en los CASIE.

La aplicación posibilitará la obtención de reportes necesarios para la creación de documentos oficiales como los reportes semanales que se le deben enviar al Jefe de Departamento Central, modelo de medios básicos de los diferentes salones de los CASIE.

2.3 Modelo del negocio

2.3.1 Reglas del Negocio

✓ La manipulación de los equipos y recursos de los CASIE, se hará solamente por el personal de la institución, directamente encargado de dichas operaciones.

✓ Todas las semanas hay que elaborar los reportes semanales de asistencia para enviárselos al Jefe de Departamento Central.

✓ Los salones pueden ser reservados por profesores, alumnos o trabajadores no docentes.

✓ Si un salón está reservado no puede ser utilizado en ese momento.

✓ Los materiales que se encuentran en los CASIE solo son de consulta.

2.3.2 Actores y Trabajadores del Negocio

Actor del negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externa con los que el negocio interactúa. Lo que se modela como actor es el papel que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. [32]

Actor	Descripción
Usuario	Es cualquier persona que llega al CASIE y solicita alguno de los servicios que allí se brindan. Este usuario puede ser un estudiante, Alumno Ayudante, trabajador docente o no docente, etc.
Profesor	Es la persona encargada de solicitar una reservación para utilizar algunos de los salones de los CASIE. Las reservaciones pueden ser de forma presencial o por vía telefónica.
Jefe Departamento Central	Es la persona encargada de solicitar un inventario de los recursos materiales de los CASIE.

Tabla # 1: Actores del Negocio

Trabajador del negocio

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. [33]

Trabajador	Descripción
Técnico General	Es la persona encargada de llevar el control de los usuarios que visitan los CASIE. Registra las reservaciones realizadas para utilizar alguno de los salones. Responsable del control de los materiales y servicios que se ofrecen en los CASIE. Además de realizar inventarios -cada cierto tiempo- a los materiales y recursos con que cuenta cada uno de los CASIE.

Tabla # 2: Trabajadores del Negocio

2.3.3 Modelo de casos de uso del negocio

El modelo de Casos de Uso del Negocio, es un modelo que describe los procesos de un negocio (casos de uso del negocio) y su interacción con elementos externos (actores). Describe

las funciones que el negocio pretende realizar y su objetivo básico es describir cómo el negocio es utilizado por sus clientes y socios. [34]

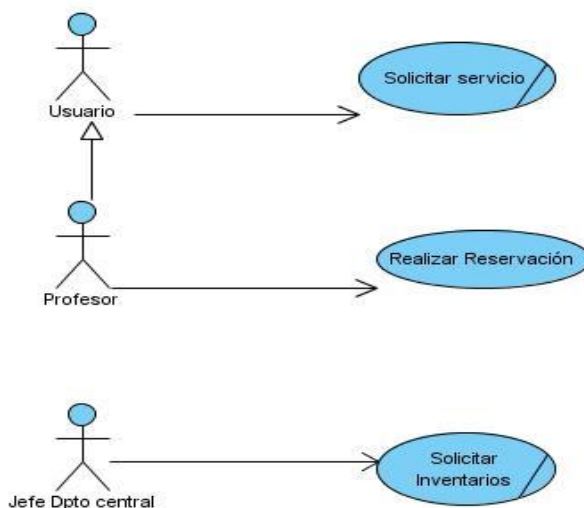


Figura # 1: Diagrama Casos de Uso del Negocio

2.3.4 Especificación de los Casos de Uso del Negocio

Caso de uso: Solicitar servicio: Descripción textual

Caso de Uso	Solicitar Servicio	
Actores	Usuario (inicia)	
Resumen	<p>El caso de uso inicia cuando el usuario llega al CASIE y solicita algún servicio de los que se brindan.</p> <p>Durante este proceso el técnico general le pide el solapín y registra los datos (especificar qué datos) del usuario en un documento Excel, para así brindarle el servicio solicitado. Así finaliza el caso de uso.</p>	
Casos de Uso asociados	-	
Acción del actor	Respuesta del proceso de negocio	
1. El usuario solicita un servicio de los que se brindan en el CASIE.	2. El técnico general le pide el solapín y algunos datos personales para brindarle el servicio solicitado.	

3. El usuario provee los datos.	4. El técnico general guarda los datos en un documento Excel, donde registra la asistencia de los usuarios al CASIE.
	5. El técnico general le brinda el servicio solicitado.
Otras secciones	2.1- Si el usuario no tiene el solapín, se le niega la solicitud del servicio.

Tabla # 3: Caso de Uso del Negocio: Solicitar Servicio

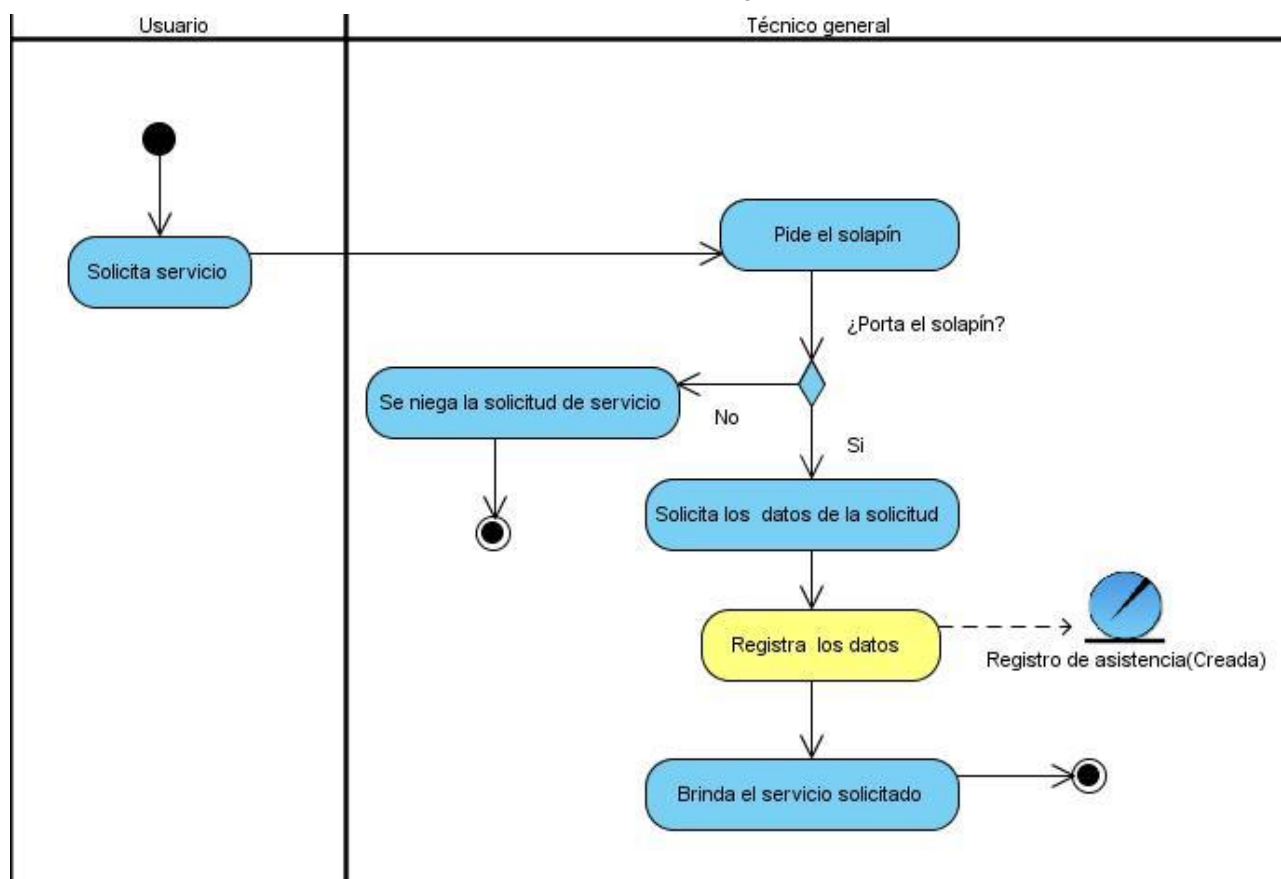


Figura # 2: Diagrama de Actividad Caso de Uso del Negocio: Solicitar Servicio

Caso de Uso: Realizar Reservación: Descripción Textual

Caso de Uso	Realizar Reservación
Actores	Profesor(inicia)

Capítulo 2: Características del Sistema

Resumen	El caso de uso se inicia cuando el profesor realiza una reservación para utilizar uno de los salones de los CASIE. El profesor debe realizar la reservación de forma presencial o por vía telefónica. El técnico general registra los datos pertinentes y guarda los datos de la reservación en un documento Excel. Finaliza así el caso de uso.	
Casos de Uso asociados	-	
Acción del actor	Respuesta del proceso de negocio	
1. El profesor realiza la reservación para utilizar uno de los salones de los CASIE.	2. El técnico general verifica que el salón que desea reservar esté disponible.	
	3. Le pide los datos correspondientes para la reservación.	
4. El profesor provee los datos.	5. Guarda los datos en un documento Excel, donde se registran las reservaciones.	
Otras secciones	2.1- Si no hay salón disponible, no se realiza la reservación.	

Tabla # 4: Caso de Uso del Negocio: Realizar Reservación

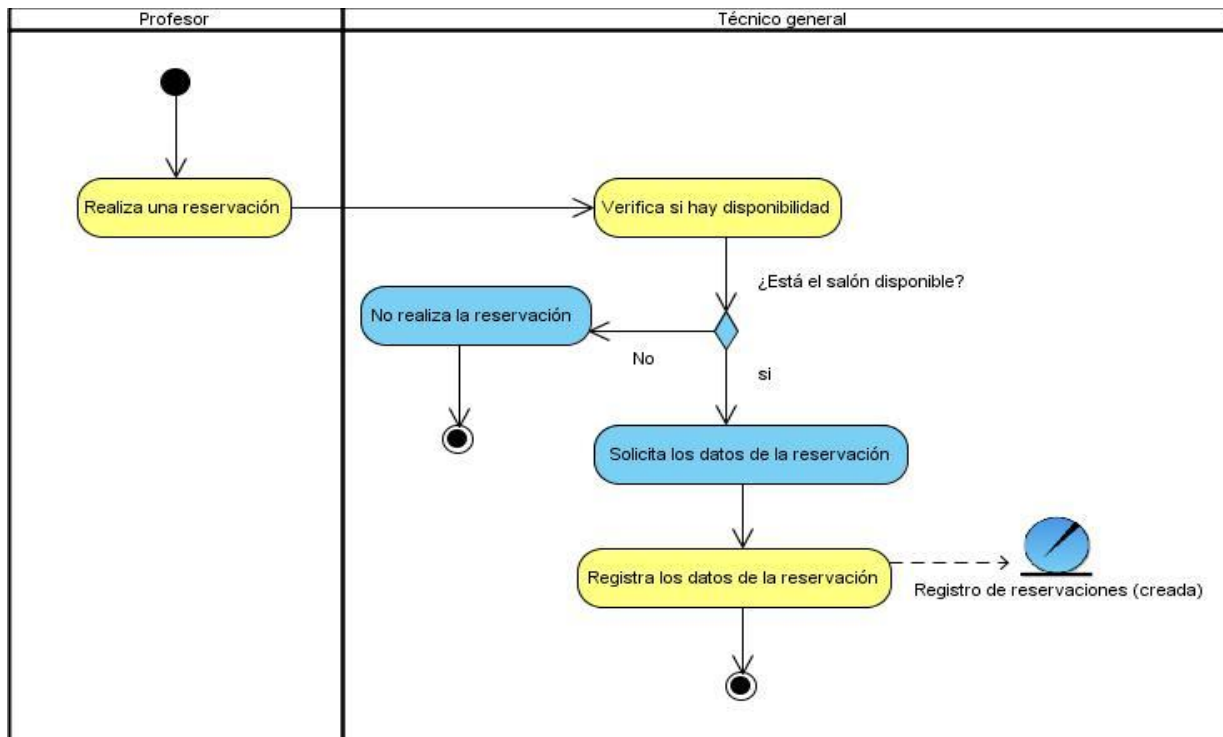


Figura # 3: Diagrama de Actividad Caso de Uso del Negocio: Realizar Reservación

Caso de Uso: Solicitar Inventario: Descripción Textual

Caso de Uso	Solicitar Inventario	
Actores	Jefe de Departamento Central	
Resumen	El caso de uso inicia cuando el Jefe de Departamento Central solicita un inventario, el técnico general realiza una verificación de los medios del CASIE. Durante este proceso el técnico general almacena los datos de todos los medios en un documento Excel. Finaliza así el caso de uso.	
Casos de Uso asociados	-	
Acción del actor	Respuesta del proceso de negocio	
1. El actor solicita realizar un inventario.	2. El técnico general verifica todos los medios del CASIE que se encuentran en los salones de lectura, computación y audiovisual.	
	3. Almacena en el Modelo de Medios Básicos (documento Excel), los datos de todos los medios del CASIE.	

Tabla # 5: Caso de Uso del Negocio: Solicitar Inventario

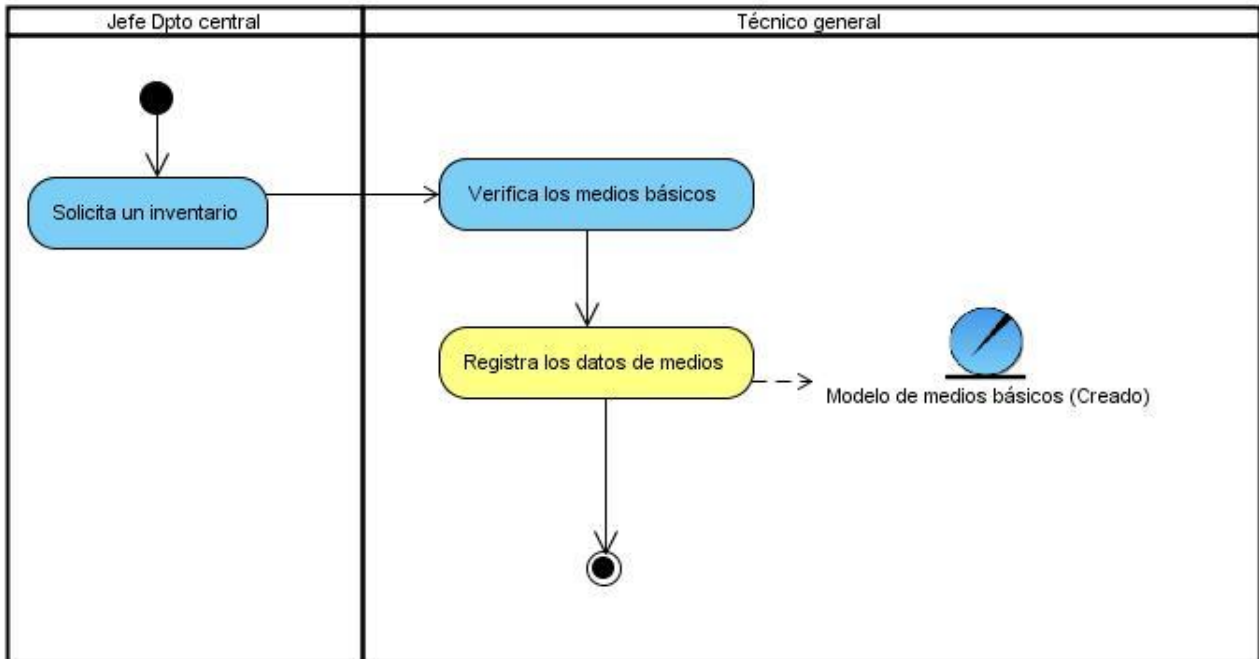


Figura # 4: Diagrama de Actividad Caso de Uso del Negocio: Solicitar inventario

2.3.5 Diagrama de Clases del Modelo de Objeto del Negocio

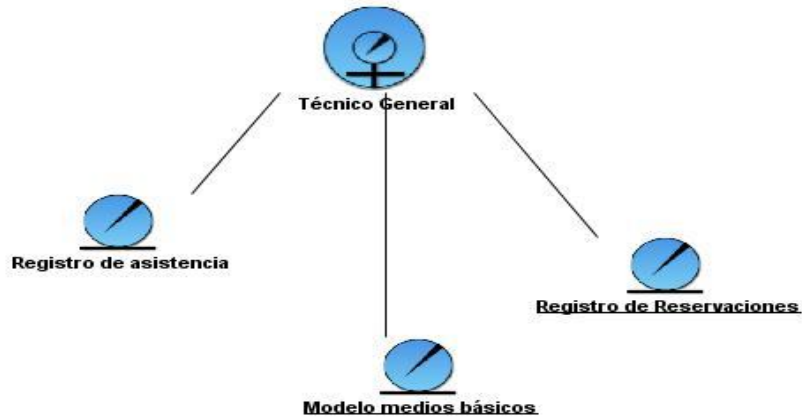


Figura # 5: Modelo de Objeto

2.4 Especificación de los requisitos de software

Un proyecto no puede ser exitoso sin una descripción detallada, correcta y exhaustiva de los requerimientos que definen lo que debe hacer un sistema y la forma en que debe hacerlo. Ellos

permiten una comprensión completa de todo el sistema, fundamentalmente cuando se describe cada uno de ellos. Se dividen en dos grandes grupos: aquellos que permiten plantear las principales funcionalidades que puede llegar a tener el sistema propuesto y los que permiten que la aplicación funcione correctamente y plantean una estrategia acerca de cómo debe hacerse la misma. Los requisitos funcionales y no funcionales permitirán satisfacer las necesidades de los usuarios finales, guiando así el proceso de desarrollo de software.

2.4.1 Requisitos Funcionales

Los requisitos funcionales especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto.

RF1. Autenticar usuario	RF27. Listar inventario
RF2. Adicionar grupo	RF28. Buscar inventario
RF3. Listar grupo	RF29. Reporte inventario
RF4. Buscar grupo	RF30. Adicionar usuario
RF5. Reporte PDF asistencia grupo	RF31. Listar usuario
RF6. Modificar grupo	RF32. Buscar usuario
RF7. Eliminar grupo	RF33. Reporte PDF asistencia usuario
RF8. Adicionar material	RF34. Modificar usuario
RF9. Listar material	RF35. Eliminar usuario
RF10. Buscar material	RF36. Adicionar asesor
RF11. Reporte PDF material	RF37. Listar asesor
RF12. Modificar material	RF38. Buscar asesor
RF13. Eliminar material	RF39. Reporte PDF asistencia asesor
RF14. Adicionar medio básico	RF40. Modificar asesor
RF15. Listar medio básico	RF41. Eliminar asesor
RF16. Buscar medio básico	RF42. Adicionar CASIE
RF17. Reporte PDF medio básico	RF43. Listar CASIE
RF18. Modificar medio básico	RF44. Buscar CASIE
RF19. Eliminar medio básico	RF45. Modificar CASIE
RF20. Adicionar reservación	RF46. Eliminar CASIE
RF21. Listar reservación	RF47. Adicionar permisos
RF22. Buscar reservación	RF48. Listar permisos
RF23. Reporte PDF reservación	RF49. Buscar permisos

RF24. Modificar reservación RF25. Eliminar reservación RF26. Adicionar inventario	RF50. Modificar permisos RF51. Eliminar permisos RF52. Ver detalles de inventario
--	--

2.4.2 Requisitos No Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener, forman una parte significativa de la especificación. Tienen la característica de marcar la diferencia, pues una vez comprobado que el producto cumple con lo requerido por el cliente, las propiedades no funcionales; como cuán usable, seguro, conveniente y agradable, distinguen a un producto bien aceptado de uno con poca aceptación.

Para un correcto funcionamiento el sistema debe tener como características principales para su funcionamiento los siguientes requisitos no funcionales:

Usabilidad: El sistema debe ser diseñado de forma tal que sus usuarios obtengan los conocimientos necesarios, en el menor tiempo posible, para la explotación de sus funcionalidades.

Seguridad: Es el requerimiento más complejo y difícil de garantizar, a su vez permitirá garantizar que la aplicación será utilizada correctamente por cada usuario según sus niveles permitidos.

- ✓ **Confidencialidad:** Se debe tratar el manejo de permisos de forma que solamente se acceda a la información autorizada de acuerdo a los niveles de permisos que debe tener cada usuario del sistema. De esta forma se garantiza que la información no sea expuesta a personal indebido.
- ✓ **Integridad:** Se debe tratar el manejo de la información de forma tal, que la información no sea modificada por personal ajeno. Esto evitará alteraciones en los resultados planteados en la documentación.
- ✓ **Disponibilidad:** Se deberá garantizar el acceso pleno de cada usuario con facultades para el uso de la aplicación las 24 horas del día.

Respaldo y recuperación de base de datos: El sistema debe permitir la realización de copias de respaldo hacia otros dispositivos de almacenamiento para prevenir pérdida de

información. Esta base de datos debe ser posible de recuperar mediante las copias de respaldo previamente creadas.

Soporte: Debe presentar un diseño que sea capaz de la realización de pruebas que permitan la explotación de la aplicación de forma eficiente. Se incluye la posibilidad de brindar asistencia técnica que permita la solución de problemas en tiempo real de ejecución que garantice la solución de fallas que pueda presentar. Debe ser garantizada su adaptabilidad y compatibilidad con los distintos Sistemas Operativos.

Interfaz: El diseño de la interfaz visual debe ser amigable para los usuarios que interactúan con la aplicación, de forma tal que permita el fácil entendimiento de las funcionalidades que brinda, además de poseer colores amigables y refrescantes para una mejor interacción entre el usuario y la aplicación.

El diseño de la interfaz permitirá mostrar mensajes para la guía de usuarios en caso de errores en entradas inválidas de los datos o de confirmación de realización de actividades.

Software: Para la elaboración de la aplicación se utilizará como sistema operativo Windows XP con Service Pack 3.

Hardware: Para el desarrollo de la aplicación propuesta se demanda una estación de trabajo con 80 Gb de disco duro como mínimo, CPU a 3.0 GHz y 512 MB de RAM.

2.5 Modelo del Sistema

2.5.1 Actores del Sistema

Actores	Objetivo
Profesor	Es el encargado de realizar una reservación para utilizar uno de los salones de los CASIE
Técnico General	Es el encargado de llevar la gestión y el control de los grupos docentes, asesores; así como atender a los diferentes usuarios de los CASIE, ya sean estudiantes, profesores, trabajadores, alumnos ayudantes etc. Es el responsable también de gestionar los materiales que se brindan en los CASIE y realizar inventarios a los medios con que cuentan los CASIE.

Capítulo 2: Características del Sistema

Administrador	Es el encargado de gestionar los permisos de los diferentes usuarios del sistema, así como gestionar los CASIE, en caso que se quiera añadir un nuevo CASIE al sistema
Visualizador	Es el encargado de ver la información que desee, buscar persona, material, grupo o cualquier inventario que se haya realizado en los CASIE
Jefe Dpto.	Es el encargado de listar los datos de inventarios, así como los de medios básicos.

Tabla # 6 Actores del Sistema

2.5.2 Diagrama de Casos Uso del Sistema.

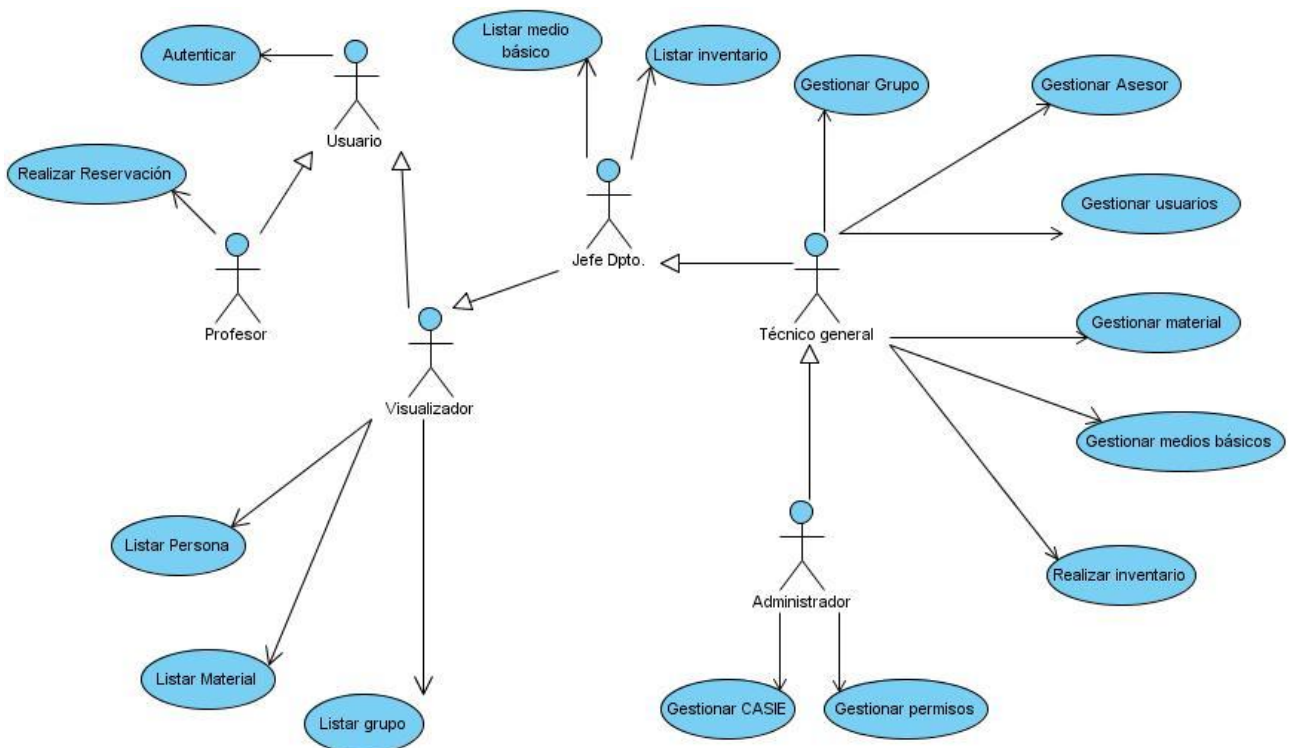


Figura # 6: Diagrama de Casos de Uso del Sistema

2.5.3 Especificación de los Casos de Uso del Sistema

A continuación se muestran las especificaciones de los casos de uso más significativos. El resto de las especificaciones se pueden encontrar en el [Anexo 1](#) así como las descripciones ampliadas de los casos de uso se encuentran en el [Anexo 2](#).

Caso de uso	Realizar Reservación
Objetivo	Realizar una reservación para utilizar un salón del CASIE.
Actores	Profesor
Resumen	El caso de uso inicia cuando el profesor entra en el sistema y selecciona la opción Realizar Reservación, el sistema permite introducir / eliminar o modificar los datos de la reservación y muestra la disponibilidad de los salones. El sistema confirma que se realizó la reservación. El sistema brinda la posibilidad también de generar el listado de reservaciones realizadas. Termina el caso de uso.

Tabla # 7: Caso de uso: Realizar Reservación

Caso de uso	Realizar inventario
Objetivo	Realizar inventario a los medios y materiales de los CASIE.
Actores	Técnico General
Resumen	El caso de uso inicia cuando el técnico general accede a la opción Realizar inventario, el sistema brinda la posibilidad de escoger el tipo de inventario a realizar. El técnico general escoge una de las opciones anteriores y registra los datos del inventario en el sistema. El sistema permite generar el inventario en un documento especificado. El caso de uso termina.

Tabla # 8: Caso de uso: Realizar inventario

Caso de uso	Gestionar usuarios
Objetivo	Gestionar los usuarios registrados en el sistema
Actores	Técnico general
Resumen	El caso de uso inicia cuando el técnico general accede a la opción Gestionar usuarios, el sistema brinda la posibilidad escoger las siguientes opciones: adicionar/eliminar/modificar los usuarios en el sistema, el actor realiza una de las opciones anteriores. El caso de uso termina.

Tabla # 9: Caso de uso: Gestionar usuarios

En el capítulo se identificaron los actores del negocio: profesor, usuario y técnico general y como trabajador del negocio: el técnico general. Los casos de uso del negocio identificados son: solicitar servicio, realizar reservación y realizar inventarios.

También se identificaron como actores que intervienen en el sistema: el técnico general, profesor, administrador y visualizador. Se especificaron todos los requerimientos funcionales y no funcionales del sistema. Se identificaron todos los casos de uso del sistema, los cuales fueron descritos detalladamente y reflejan las funcionalidades recogidas previamente en los requerimientos. El desarrollo del flujo de trabajo Requerimiento y los artefactos obtenidos dan paso a la realización del diseño del sistema propuesto.

Capítulo 3: Análisis y Diseño del Sistema


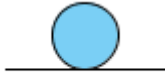
En este capítulo se realiza la descripción del diseño de la aplicación mediante el uso de los distintos diagramas afines para dicha actividad. Se realiza el análisis de cada uno de los casos de usos y se describe, mediante el uso de estereotipos web, la relación entre las distintas páginas que conforman la aplicación que se pretende desarrollar. Se obtiene además el diagrama de clases persistentes para la generación de la base de datos. Se definen los principios de diseño y los patrones y estándares a utilizar y se define el diagrama de despliegue del sistema propuesto.

3.1 Modelo de análisis

El análisis consiste en obtener una visión detallada del sistema, de modo que solo se interesa por los requisitos funcionales, para refinarlos y estructurarlos, además es el principal eslabón para el comienzo de las actividades de diseño e implementación. El objetivo del análisis, se centra en comprender perfectamente los requisitos del software y no en precisar cómo se implementa el producto.

3.1.1 Diagrama de Clases del Análisis.

El diagrama de clase del análisis se realiza para cada caso de uso del sistema. Muestra las clases participantes en dichos casos de uso, así como la relación entre ellas. En los diagramas de clases se identifican tres tipos de clases: Interfaz, Controladora y Entidad.

Clases	Descripción	Representación
Interfaz	Las clases <i>interfaz</i> se utilizan para modelar la interacción entre el sistema y sus actores.	 Interfaz
Entidad	Las clases <i>entidad</i> se utilizan para modelar información que poseen larga vida y que es a menudo persistente.	 Entidad


Control	Las clases <i>control</i> representan coordinación, secuencia, transacciones y control de objetos y son utilizadas para encapsular el control de un caso de uso.	 Control
---------	--	---

Tabla # 7: Representación de los estereotipos a utilizar en el diagrama de clases del análisis.

Diagramas de clases del análisis.

A continuación se muestran los diagramas de clases del análisis de los casos de uso más significativos, el resto de los diagramas de clases del análisis se pueden encontrar en el [Anexo 3](#).

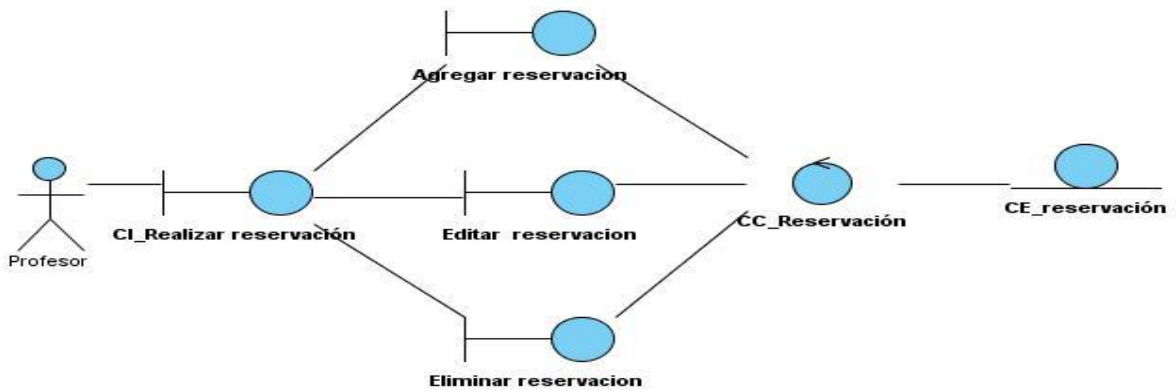


Figura # 10: Diagrama de clases del análisis: Realizar reservación.

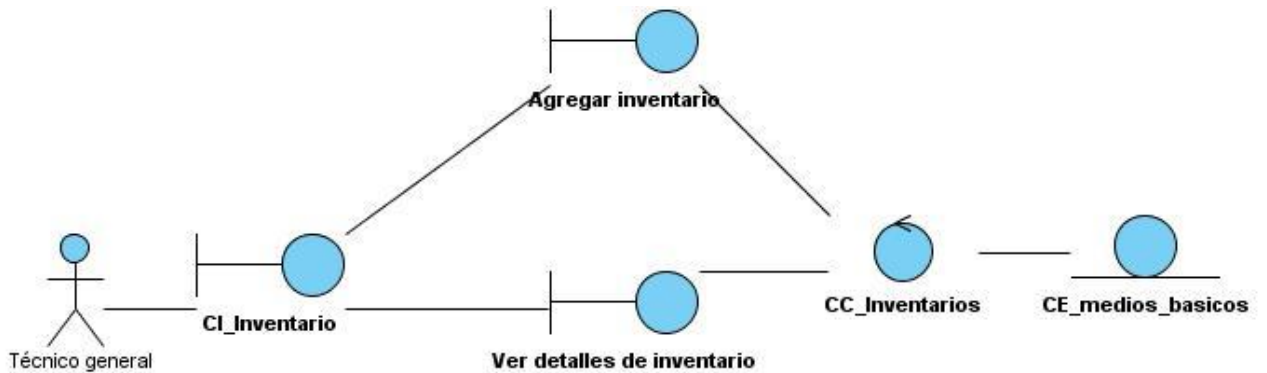


Figura # 11: Diagrama de clases del análisis: Realizar inventario.

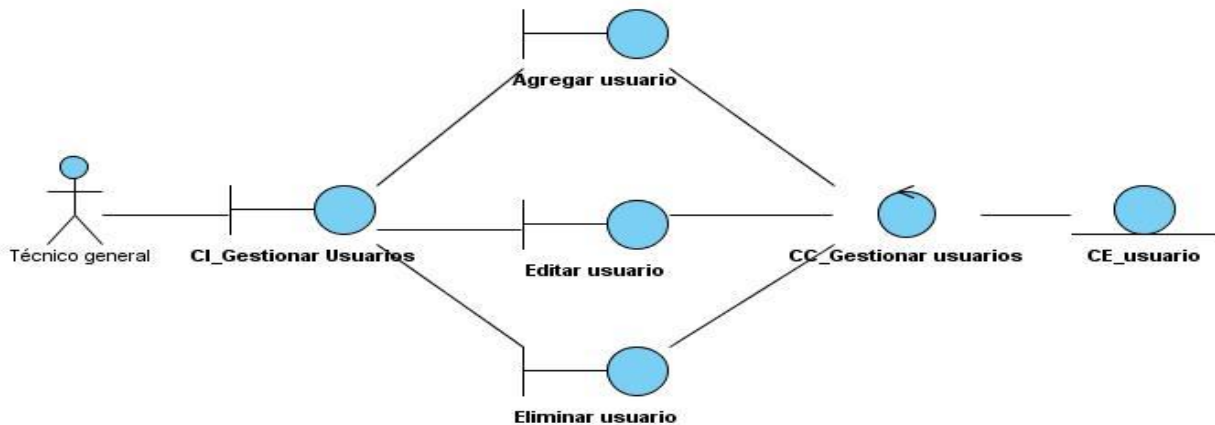


Figura # 12: Diagrama de clases del análisis: Gestionar usuarios.

3.2 Modelo del diseño

El propósito del diseño es modelar el sistema y encontrar la forma para que soporte todos los requisitos definidos. Se planean todos los aspectos relacionados con las restricciones y características del sistema como lo son el lenguaje de programación a utilizar, el sistema operativo donde se podrá ejecutar la aplicación 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.

3.2.1 Descripción de la arquitectura

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, ya que los subsistemas y sus interfaces constituyen la estructura fundamental del producto de software. Se utilizó el patrón de arquitectura Modelo Vista Controlador (MVC), la misma separa la lógica de negocio y la presentación por lo que se consigue un mantenimiento más sencillo de las aplicaciones.

El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la

abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. [35]

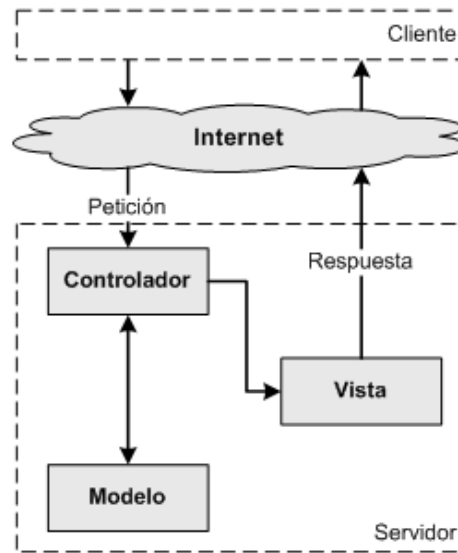


Figura # 13: Patrón Modelo Vista Controlador

El modelo: es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El modelo no tiene conocimiento específico de los controladores o de las vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar a las vistas cuando cambia el modelo. [36]

La vista: el objeto que maneja la presentación visual de los datos representados por el modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio modelo. [37]

El controlador: es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del modelo o por alteraciones de la vista. Interactúa con el modelo a través de una referencia al propio modelo. [38]

Según *Patricio Letelier. Universidad Politécnica de Valencia*, las Ventajas del Patrón Modelo Vista Controlador son: [39]

- ✓ MVC aporta una construcción de software sostenible, en la que se pueden localizar de forma ágil los errores. Supone un diseño modular, y muy poco acoplado, lo que favorece la reutilización.

- ✓ Facilita la labor de todo el equipo: diseñadores gráficos, programadores, diseñadores de base de datos. Las aplicaciones que lo implementan presentan una extensibilidad y una mantenibilidad únicas comparadas con otras aplicaciones basadas en otros patrones.

- ✓ Al separar la presentación de la programación (o lógica de negocio), la aplicación es más fácil de modificar en el futuro.

- ✓ El resultado es más claro, y el reparto de tareas dentro del equipo de trabajo es más fácil; la depuración de la aplicación es más sencilla y, finalmente, puede utilizarse un marco de trabajo (o framework) bien testado.

- ✓ Las modificaciones a las vistas no afectan en absoluto a los otros módulos de la aplicación.

- ✓ Es mucho más sencillo agregar múltiples representaciones de los mismos datos o información.

- ✓ Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.

- ✓ Crea independencia de funcionamiento.

- ✓ Facilita el mantenimiento en caso de errores.

3.3 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo del software y otros ámbitos referentes al diseño de interacción de interfaces. Actualmente el uso de patrones se ha generalizado entre los desarrolladores de aplicaciones en todo el mundo como una manera de estandarizar la creación de los nuevos sistemas informáticos.

Un patrón de diseño es 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 con la solución de 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 Ext JS se encargarán de visualizar las interfaces ya que cuentan con la información para crear los diferentes componentes visuales.

✓ **Creador:** 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 de este patrón permite crear las dependencias mínimas necesarias entre las clases, lo cual favorece al mantenimiento del sistema.

✓ **Bajo Acoplamiento:** Brinda como solución asignar responsabilidades de manera que las clases no dependan fuertemente de otras. Ofrece 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, para evadir un trabajo excesivo. Su utilización mejora la claridad y facilidad con que se entiende el diseño, simplifica el mantenimiento y las mejoras de funcionalidad, generan un bajo acoplamiento, soporta mayor capacidad de reutilización.

3.4 Diagramas de Clases del Diseño

Un diagrama de clases del diseño, es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Además representan 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 se desprende un diagrama de clases.

Para la realización de los diagramas de clases del diseño se presentan como elementos significativos a tres clases UML fundamentales: server page, client page y form, empleadas para

el código servidor, código cliente y formularios respectivamente. Esto permite 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:




Clases	Descripción	Representación
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. Solo puede mantener relaciones con objetos que se encuentren en el servidor.	 Server page
Client Page	Representan páginas que son dibujadas por el navegador web y pueden ser una combinación de algún o algunos lenguajes, scripts del lado del cliente	 Client page
Form	Representa una colección de campos de entrada que forman parte con una página del lado cliente	 Form

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

Relaciones entre las clases:

Clases/Relaciones	Server Page	Client Page	Form
Server Page	Redirect	Build/Redirect	—
Client Page	Link	Link/Redirect	Agregación
Form	Submit	—	—

Tabla # 9: Representación de las relaciones de clases del diseño con estereotipos web.

Capítulo 3: Análisis y Diseño del sistema

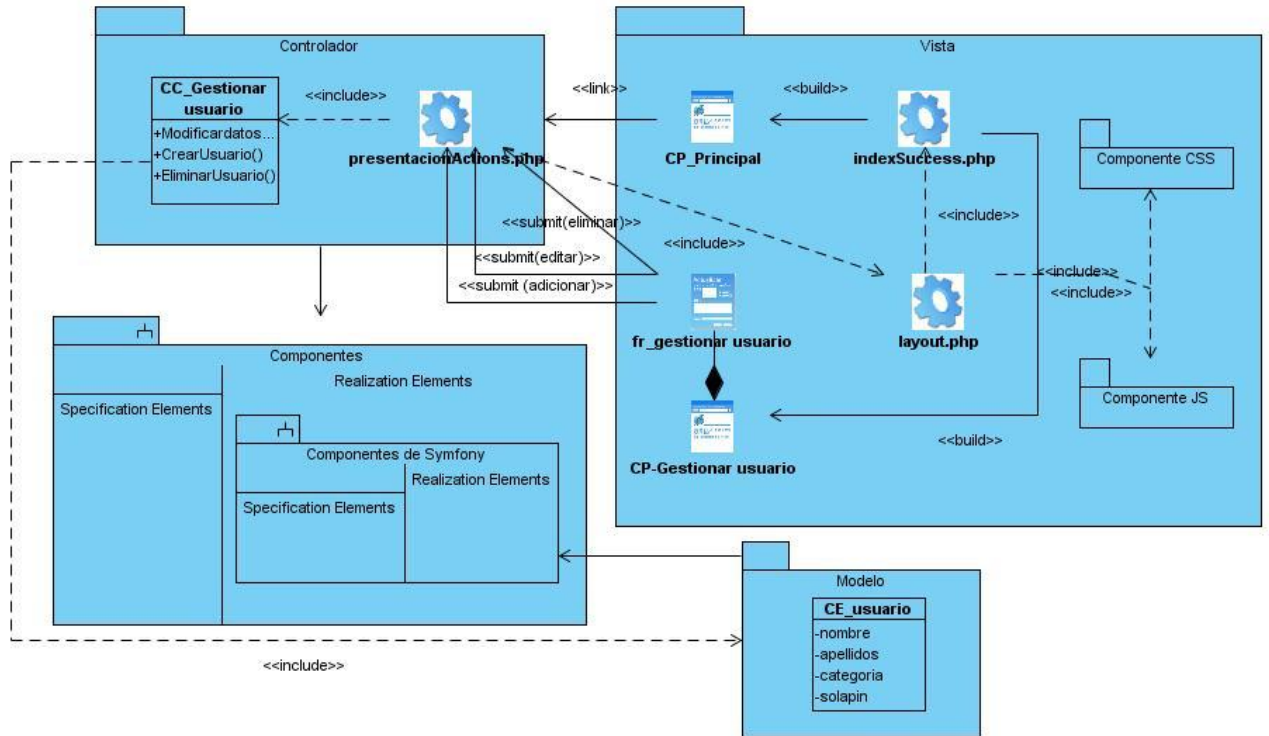


Figura # 14: Diagrama de clases del diseño con estereotipos web: Gestionar usuario.

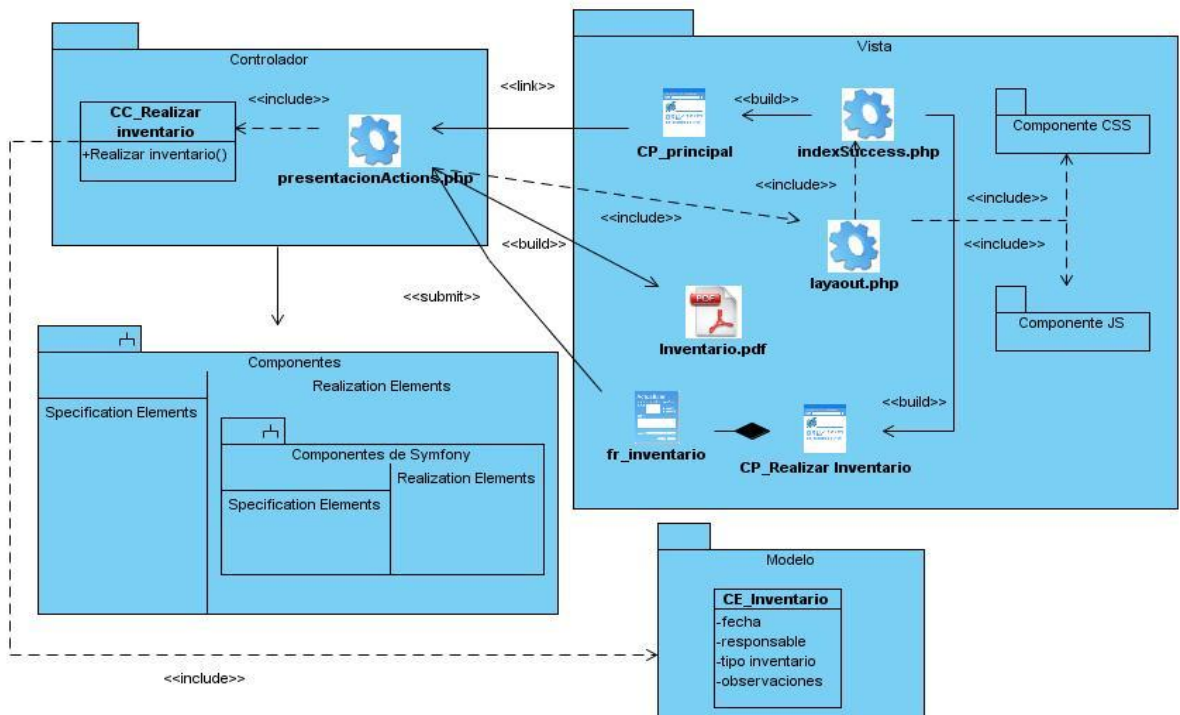


Figura # 15: Diagrama de clases del diseño con estereotipos web: Realizar inventario.

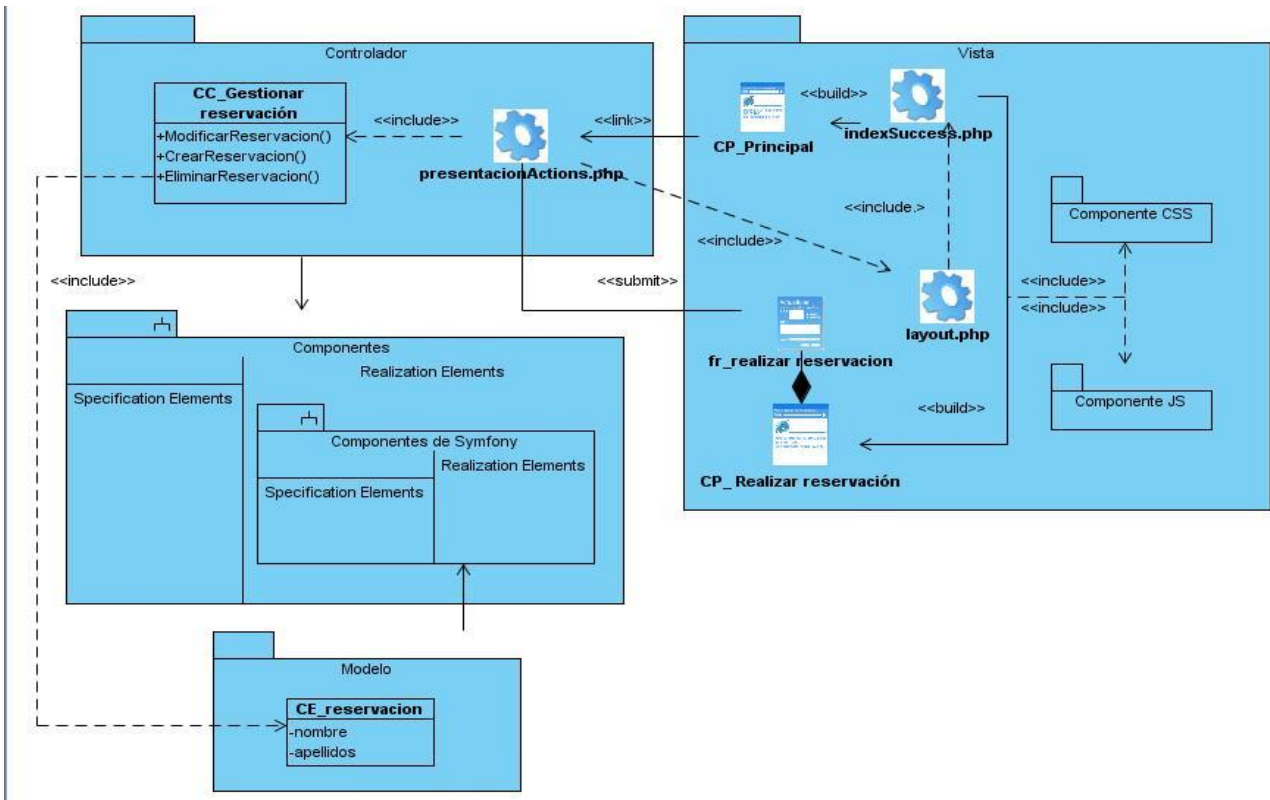


Figura # 16: Diagrama de clases del diseño con estereotipos web: Realizar reservación.

El resto de los diagramas de clases del diseño, se pueden encontrar en el [Anexo 4](#).

3.5 Modelo de datos

El modelo de datos representa todas las tablas de la Base de Datos de la aplicación, permite describir los datos contenidos en la base de datos y las relaciones que existen entre ellos. Para ello se realiza el diagrama de clases persistentes.

3.5.1 Diagrama de clases persistentes

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Lo contrario son las clases temporales que son manejadas y almacenadas por el sistema

en tiempo de ejecución, por lo que dejan de existir cuando termina el programa. El diagrama de clases persistentes se encuentra en el [Anexo 5](#).

3.5.2 Descripción de las tablas del modelo de datos

Nombre: casie		
Descripción: Esta tabla muestra los datos de los CASIE		
Atributo	Tipo	Descripción
id	int	Es el identificador de la tabla CASIE
número_casie	varchar	Es el número del CASIE
jefe	varchar	Es el nombre del técnico general
cant_pc	int	Es la cantidad de PCs que hay en el CASIE

Tabla # 10: Modelo de datos. Tabla casie

Nombre: grupo		
Descripción: Esta tabla muestra los datos de los grupos.		
Atributo	Tipo	Descripción
id	int	Es el identificador de la tabla grupo
grupo	varchar	Representa el número del grupo
profesor	varchar	Es el nombre del profesor que está frente al grupo
solapín	varchar	Representa el número del solapín del profesor
cant_est	int	Es la cantidad de estudiantes que tiene el grupo
fecha	varchar	Representa la fecha en que el grupo visita el CASIE
entrada	varchar	Hora de entrada
salida	varchar	Hora de salida
actividad	varchar	Representa la actividad que va a realizar el grupo
salón	varchar	Representa el salón que utilizara el grupo
casie	varchar	Representa el CASIE donde se agregó el grupo

Tabla # 11: Modelo de datos. Tabla grupo

Nombre: login		
Descripción: Esta tabla guarda lo referente al login		
Atributo	Tipo	Descripción
id	int	Representa el identificador de la tabla login
usuario	varchar	Es el nombre de usuario que desea loguearse.
rol	varchar	Representa el rol que representa el usuario al loguearse
casie	varchar	Representa el CASIE donde se agregó el login

Tabla # 12: Modelo de datos. Tabla login

Nombre: material		
Descripción: Esta tabla muestra todo lo referente a material en formato duro		
Atributo	Tipo	Descripción
id	int	Es el identificador de la tabla material.
titulo	varchar	Representa el titulo del material.
autor	varchar	Es el nombre del autor del material.
colección	varchar	Representa la colección a la que pertenecen los materiales.
fecha	date	Representa la fecha en la cual entro el material al centro.
cant_ejemplares	int	Es la cantidad de materiales existentes.
nivel	varchar	Representa el nivel (elemental, básico, intermedio, avanzado) que tienen los materiales.
Tipo material	varchar	Representa el tipo de formato del material (formato digital o duro).
casie	varchar	Representa el CASIE donde se agregó el material

Tabla # 13: Modelo de datos. Tabla material

Nombre: reservación		
Descripción: Esta tabla muestra todo lo referente a las reservaciones		
Atributo	Tipo	Descripción
id	int	Representa el identificador de la tabla reservación
nombre	varchar	Es el nombre del profesor que va a realizar la reservación
apellidos	varchar	Son los apellidos del profesor que va a realizar la reservación
área	varchar	Representa el área al que pertenece
fecha	int	Representa la fecha en que desea realizar la reservación
entrada	varchar	Representa la hora de entrada
salida	varchar	Representa la hora de salida
salón	varchar	Es el salón que desea reservar el profesor
casie	varchar	Representa el CASIE donde se agregó el reservación
eliminado	bool	Representa si un material fue o no eliminado

Tabla # 14: Modelo de datos. Tabla reservación

Nombre: usuario		
Descripción: Esta tabla muestra todo lo referente a los usuarios		
Atributo	Tipo	Descripción
id	int	Representa el identificador de la tabla usuario.
nombre	varchar	Es el nombre del usuario
apellidos	varchar	Representa los apellidos del usuario
categoría	varchar	Constituye la categoría que presenta el usuario
solapín	varchar	Número del solapín
área	varchar	Representa el área a la que pertenece el usuario
fecha	varchar	Fecha en el que el usuario visita el CASIE

Capítulo 3: Análisis y Diseño del sistema

sesión	varchar	Representa la sesión que utiliza el usuario
entrada	varchar	Representa la hora de entrada del usuario
salida	varchar	Representa la hora de salida del usuario
servicio	varchar	Representa el servicio que solicita el usuario
material	varchar	Representa el tipo de material que solicita el usuario.
salón	varchar	Es el salón que solicita el usuario.
casie	varchar	Representa el CASIE donde se agregó el usuario

Tabla # 15: Modelo de datos. Tabla usuario

Nombre: usuario_asesor		
Descripción: Esta tabla muestra todo lo referente a los asesores		
Atributo	Tipo	Descripción
id	int	Es el identificador de la tabla usuario asesor
nombre	varchar	Es el nombre del asesor
apellidos	varchar	Representa los apellidos del asesor
solapín	varchar	Es el número de solapín del asesor
fecha	varchar	Fecha en que el asesor brindara servicio
entrada	varchar	Representa la hora de entrada del asesor
salida	varchar	Representa la hora de salida del asesor
asesoría	varchar	Representa el tipo de asesoría que brindara el asesor
casie	varchar	Representa el CASIE donde se agregó el usuario asesor

Tabla # 16: Modelo de datos. Tabla usuario asesor

Nombre: medios_básicos		
Descripción: Esta tabla muestra todo lo referente a los usuarios		
Atributo	Tipo	Descripción
id	int	Representa el identificador de la tabla medios_básicos
número inv	varchar	Representa el número de inventario del medio
descripción	varchar	Representa la descripción del medio básico
fecha	date	Representa la fecha en la que entro el medio básico al centro
no serie	varchar	Representa el número de serie del medio básico.
estado técnico	varchar	Representa el estado en el que se encuentra el medio.
casie	varchar	Representa el CASIE donde se agregó el medio básico
eliminado	bool	Representa si un material fue o no eliminado

Tabla # 17: Modelo de datos. Tabla medios_básicos

Nombre: inventario		
Descripción: Esta tabla muestra todo lo referente a los usuarios		
Atributo	Tipo	Descripción
id	int	Representa el identificador de la tabla inventario.
fecha	date	Representa la fecha de realización del inventario.
responsable	varchar	Representa el responsable del inventario.

t_inventario	varchar	Representa el tipo de inventario (Materiales o Medio Básico).
observaciones	varchar	Representa las observaciones existentes al momento de realizar el inventario
casie	varchar	Representa el CASIE donde se agregó el inventario

Tabla # 17: Modelo de datos. Tabla inventario

Nombre: inventario_m		
Descripción: Esta tabla muestra todo lo referente a los usuarios		
Atributo	Tipo	Descripción
id	int	Representa el identificador de la tabla inventario_m.
Id_inventario	int	Representa el identificador del inventario realizado.
Id_m	int	Representa el identificador del (Material o Medio Básico) que se le realizó un inventario previamente.

Tabla # 17: Modelo de datos. Tabla inventario_m

3.6 Diagrama de despliegue

Un diagrama de despliegue muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. Muestra la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes de software y hardware, procesos y objetos que los ejecutan. Este diagrama es útil para ilustrar la arquitectura física de un sistema. [40]

Para el despliegue se requiere de una PC_cliente, que representa la computadora del usuario. Un Servidor de Base de Datos Pentium 4, Service Pack 2 o superior para su uso sobre Windows, con 1Gb de memoria RAM como mínimo, 120G de disco duro para el almacenamiento de datos y un Sistema Gestor de Base de Datos en MySQL montado sobre plataforma Windows o Linux, preferentemente software libre. Para el Servidor de aplicaciones se requiere una PC Pentium 4, Service Pack 2 o superior para su uso en Windows, 1Gb de memoria RAM como mínimo, 80G de disco duro y que permita su uso desde Windows o Linux, preferiblemente sobre software libre. Además desde la PC_cliente se accede a la impresora mediante el puerto USB/Serie.

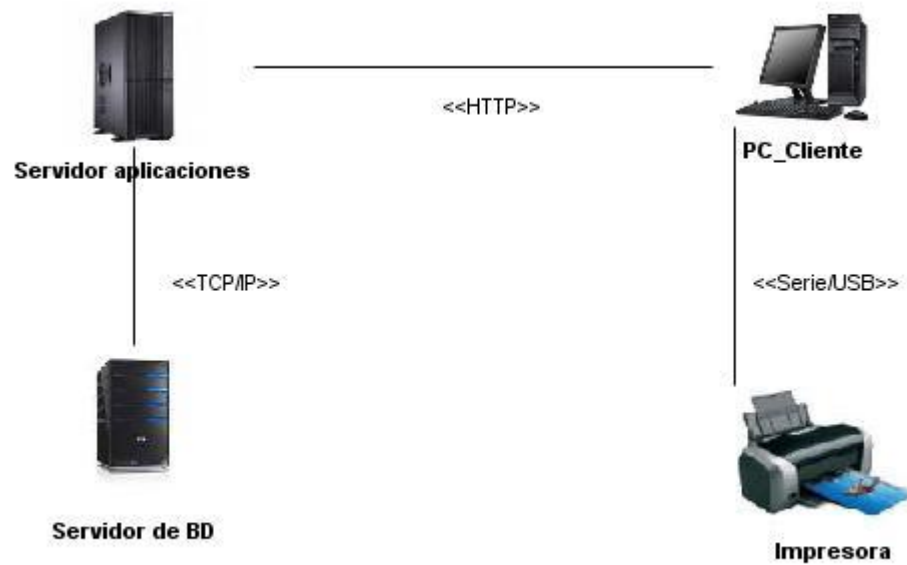


Figura # 16: Diagrama de despliegue

En este capítulo se detallaron las fases de Análisis y Diseño del sistema propuesto, donde se obtuvieron los diferentes artefactos correspondientes a los flujos de trabajo desarrollador acorde a la metodología utilizada. En el análisis se obtuvo los diagramas de clases del análisis así como la descripción de las mismas y en el diseño los diagramas de clases del diseño con estereotipos web. Se describió el patrón arquitectónico Modelo Vista Controlador (MVC).

Además se obtuvo el diagrama de clases persistentes para la generación de la base de datos, así como la descripción de cada una de las tablas de la base de datos. Se realizó el diagrama de despliegue correspondiente al sistema que se propone.

Capítulo 4: Implementación

En el presente capítulo se describe el modelo de implementación con el objetivo de definir la estructura y organización de la aplicación, además de implementar las clases y subsistemas encontrados durante el diseño representado en los diagramas de componentes.

4.1 Modelo de implementación

El modelo de implementación es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros de código fuente, y otros tipos de ficheros necesarios para la implantación y despliegue del sistema. Describe también, como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen de los componentes unos de otros. [41]

4.1.1 Diagrama de componentes

Un diagrama de componentes muestra las dependencias entre los componentes de software. Incluye los clasificadores que los especifican (por ejemplo, clases de implementación) y los artefactos que los implementan, tales como, archivos de código fuente, archivos de código binario, archivos ejecutables, scripts. [42]

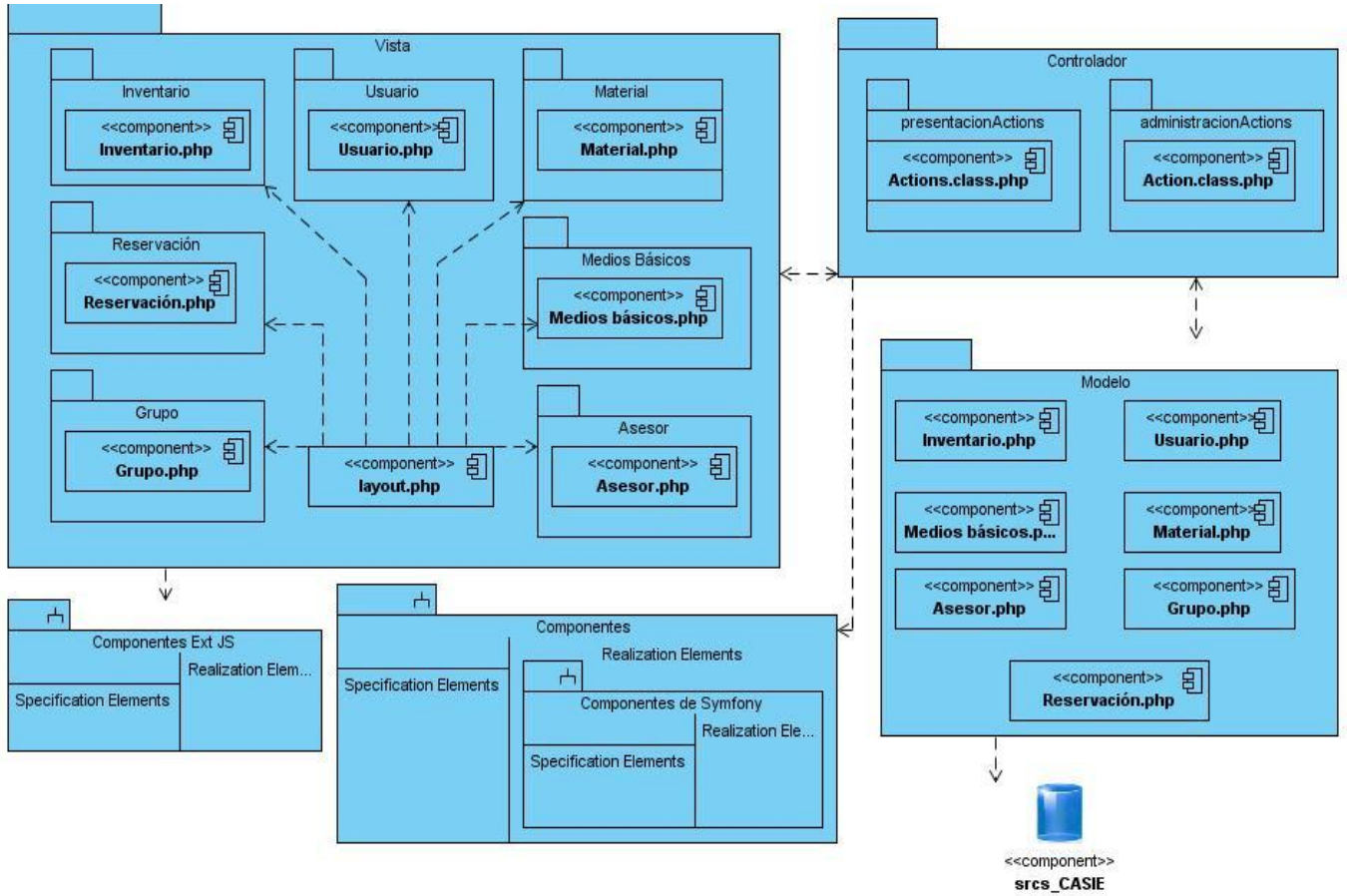


Figura # 17: Diagrama de Componentes

4.2 Seguridad

La seguridad es un elemento esencial para lograr la integridad, confidencialidad y autenticidad de la información almacenada y manejada por el sistema. Para ello el sistema implementa mecanismos que permitan asegurar la misma.

Las herramientas proporcionadas por Symfony para este propósito permiten la creación de aplicaciones seguras, para obtener acceso a áreas restringidas, los usuarios necesitan estar autenticados y/o poseer ciertas credenciales. Se puede extender los privilegios del usuario mediante llamadas a métodos del objeto `sfUser`. El estado autenticado se establece con el método `setAuthenticated()`.

Al sistema y a los módulos solo podrán acceder los usuarios con los permisos previamente establecidos, de acuerdo al nivel de accesibilidad de cada uno.

En este capítulo se presentó el diagrama de implementación con el objetivo de definir la estructura y organización de la aplicación, además de implementar las clases y subsistemas encontrados durante el diseño representado en los diagramas de componentes. También se abordó el tema relacionado con la seguridad de la aplicación.

Conclusiones

Una vez culminada la investigación y después de dar cumplimiento a los objetivos propuestos, se puede arribar a las siguientes conclusiones:

- ✓ Del estudio de las soluciones informáticas para la gestión de recursos materiales y humanos a los niveles internacional, nacional e institucional, se obtuvo que no responden a las necesidades de gestión de la información de los CASIE.
- ✓ La metodología RUP, el lenguaje de modelado UML, la herramienta de modelado Visual Paradigm, el lenguaje de programación PHP, el marco de trabajo Symfony, el gestor de base de datos MySQL y la herramienta de desarrollo NetBeans constituyen la tecnología para el desarrollo del sistema.
- ✓ Entre todos los artefactos que propone la metodología de desarrollo RUP, se encuentran: diagrama de casos de usos del negocio, diagrama del sistema, diagrama de componentes y diagrama de despliegue.
- ✓ Se obtiene un sistema capaz de gestionar los recursos materiales y humanos y servicios de los CASIE, el cual responde a las necesidades de un centro de este tipo.

Recomendaciones

En aras de mejorar el sistema propuesto, se recomienda incluir en futuras investigaciones las siguientes acciones:

- ✓ Realizar pruebas funcionales al sistema.
- ✓ Añadir al sistema la función de biblioteca, que permita llevar un mayor control de los préstamos de materiales, así como la consulta de materiales en línea.
- ✓ Definir usuarios locales, cuando el servicio web no esté disponible.
- ✓ Incluir al sistema, la funcionalidad registrar las trazas de los usuarios.

Referencias Bibliográficas

1. Universidad de San Carlos de Guatemala. [En línea] [Citado el: 9 de noviembre de 2010.] http://biblioteca.usac.edu.gt/tesis/08/08_5871.pdf
2. Empleados MSD Multiusuario. [En línea] [Citado: noviembre 10, 2010.] <http://www.abcdatos.com/programas/programa/z6106.html>.
3. Wikimetal RRHHv3.0, [En línea] [Citado: noviembre 3, 2010.] <http://www.abcdatos.com/programas/programa/z7511.html>.
4. **González Acosta, Roberto.** *SISTEMA PARA LA GESTIÓN DE LOS COLABORADORES DE LA SALUD.*
5. *APLICACIÓN INFORMÁTICA PARA EL CONTROL DE LOS RECURSOS HUMANOS EN EL SISTEMA NACIONAL DE SALUD.* [En línea] [Citado: noviembre 12, 2010.]
6. Fastos. [En línea] [Citado: noviembre 15, 2010.] <http://www.desoft.cu/Productos1/FastosPagus/tabid/442/Default.aspx>.
7. , *Definición de metodología*, Conferencia de historia de la informática “Metodologías de desarrollo de software”, 2008-2009.
8. METODOLOGÍA EXTREME PROGRAMMING [En línea] [Citado el: 9 de enero de 2011.] www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.
9. **Peña, Amarilis y Mesa, Yudisleidy**, *Herramienta para automatizar el proceso de evaluación del programa de mejoras en los proyectos productivos de la UCI, basados en el modelo CMMI: s.n* 2010
10. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software.* Madrid: Pearson Educación S.A, 2000. 84-7829-036-2.
11. *Idem* a la referencia 9
12. *Idem* a la referencia 10
13. Wikipedia de Producción. [En línea] [Citado el: 20 de noviembre de 2010.] <http://ucipedia.uci.cu/index.php/UM>.
14. UML. [En línea] [Citado el: 19 de noviembre de 2010.] <http://usuarios.multimania.es/ooopere/uml.htm>
15. *Idem* a la referencia 9
16. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software.* Madrid: Pearson Educación S.A, 2000. 84-7829-036-2.

Referencias Bibliográficas

17. Sitio de Descarga de software. [En línea] [Citado el: 20 de noviembre de 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
18. **Fahnle, Pablo.** ASP.NET [En línea] [Citado el: 4 diciembre de 2010.] www.programacion.com/asp/articulo/aspnet_queues/.
19. **Potencier, François Zaninotto y Fabien.** *Symfony 1.2, la guía definitiva.* 13: 978-1590597866.
20. **Álvarez, Miguel Ángel.** JAVA [En línea] [Citado el: 5 de diciembre de 2010.] www.desarrolloweb.com/articulos/497.php.
21. *Idem* a la referencia 19
22. *Idem* a la referencia 9
23. *Idem* a la referencia 9
24. _____, *NetBeans 6.8 en Español* [Citado el: 15 de octubre de 2010.] <http://www.taringa.net/posts/downloads/4818143/netbe>
25. *Netbeans* [Online] [Citado: noviembre 23, 2010.]
26. <http://www.dadisokd.net.au.net/index.php/component/content/article/44-software-libre-dadisokd/94-netbeans-dadisokd>.
27. ZEND STUDIO [En línea] [Citado el: 16 de diciembre de 2010.] http://www.ZendStudio.org//Zend_Studio
28. *Lenguaje de programación para páginas web.* [En línea] [Citado: 8 de diciembre de 2010.] <http://www.monografias.com/trabajos7/html/html.shtml#crea>.
29. *Entrono Virtual de Aprendizaje.* [Online] [Citado: 25 de noviembre de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=2756>.
30. **Sánchez, Jorge.** *MySQL guía rápida.* 2004.
31. *Idem* a la referencia 30
32. **Jacobson, Ivar, Booch, Grady y Runbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Madrid: s.n., 2000
33. *Idem* a la referencia 31
34. *Idem* a la referencia 31
35. **Potencier, François Zaninotto y Fabien.** *Symfony 1.2, la guía definitiva.* 13: 978-1590597866.
36. **Tutorial de Java.** *Arquitectura Modelo-Vista-Controlador.* [citado 20 de marzo 2011] http://sunsite.dcc.uchile.cl/java/docs/JavaTut/Apendice/arg_mvc.html
37. *Idem* a la referencia 35

38. *Idem* a la referencia 35
39. **Arquitectura de software**. *Capitulo II. Modelo_Vista_Controlador.p.11*
40. **Letelier Patricio. Universidad Politécnica de Valencia** *Rational Unified Process(RUP)*.
41. **Caraballo, Luis Antonio Salazar**. *Prolegómenos Sobre el Lenguaje de Modelado Unificado (UML)*. 2003.
42. *Idem* a la referencia 40
43. *Idem* a la referencia 40

Bibliografía

- ✓ **Álvarez, Miguel Ángel.** JAVA. Disponible en: www.desarrolloweb.com/articulos/497.php.
- ✓ **Arquitectura de software.** Capítulo II. Modelo_Vista_Controlador.p.11
- ✓ **Centros de Auto-aprendizaje de Servicios de Idiomas Extranjeros (CASIE).** Disponible en: <http://intranet2.uci.cu/node/86/idiomas/casie>.
- ✓ **Caraballo, Luis Antonio Salazar.** Prolegómenos Sobre el Lenguaje de Modelado Unificado (UML). 2003.
- ✓ **Definición de metodología,** Conferencia de historia de la informática “Metodologías de desarrollo de software”, 2008-2009.
- ✓ **Empleados MSD Multiusuario.** Disponible en: <http://www.abcdatos.com/programas/programa/z6106.html>.
- ✓ **Entrono Virtual de Aprendizaje.** [Online] [Citado: 25 de noviembre de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=2756>.
- ✓ **Fahnle, Pablo.** Disponible en: www.programacion.com/asp/articulo/aspnet_ques/.
- ✓ **Fastos.** Disponible en: <http://www.desoft.cu/Productos1/FastosPagus/tabid/442/Default.aspx>.
- ✓ **González Acosta, Roberto.** SISTEMA PARA LA GESTIÓN DE LOS COLABORADORES DE LA SALUD.
- ✓ **Jacobson Ivar, Grady Booch, James Rumbaugh.** El Proceso Unificado de Desarrollo de Software. Madrid: Pearson Educación S.A, 2000. 84-7829-036-2.
- ✓ **José Márquez Díaz, Leonardo Sampedro, Félix Vargas.** Instalación y configuración de Apache, un servidor web gratis. 2002.
- ✓ **Lenguaje de programación para páginas web.** Disponible en: <http://www.monografias.com/trabajos7/html/html.shtml#crea>
- ✓ **Letelier Patricio. Universidad Politécnica de Valencia** Rational Unified Process (RUP).
- ✓ **METODOLOGÍA EXTREME PROGRAMMING.** Disponible en www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.
- ✓ **Netbeans.** Disponible en: <http://www.dadisokd.net.au.net/index.php/component/content/article/44-software-libre-dadisokd/94-netbeans-dadisokd>.

- ✓ **Peña, Amarilis y Mesa, Yudisleidy**, *Herramienta para automatizar el proceso de evaluación del programa de mejoras en los proyectos productivos de la UCI, basados en el modelo CMMI*: s.n 2010
- ✓ **Potencier, François Zaninotto y Fabien**. *Symfony 1.2, la guía definitiva*. 13: 978-1590597866.
- ✓ **Sánchez, Jorge**. *MySQL guía rápida*. 2004.
- ✓ Sitio de Descarga de software. Disponible en:
http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
- ✓ **Stig Sæther Bakken, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf**. *Manual de PHP*. 2002.
- ✓ **Tutorial de Java**. Arquitectura Modelo-Vista-Controlador. Disponible en:
http://sunsite.dcc.uchile.cl/java/docs/JavaTut/Apendice/arq_mvc.html
- ✓ Universidad de San Carlos de Guatemala. Disponible en:
http://biblioteca.usac.edu.gt/tesis/08/08_5871.pdf
- ✓ UML. Disponible en: <http://usuarios.multimania.es/oopere/uml.htm>
- ✓ Wikimetal RRHHv3.0. Disponible en:
<http://www.abcdatos.com/programas/programa/z7511.html>.
- ✓ Wikipedia de Producción. Disponible en: <http://ucipedia.uci.cu/index.php/UM>.
- ✓ ZEND STUDIO. Disponible en http://wwwZendStudio.org/Zend_Studio

Anexos:

Anexo 1: Especificación de los casos de uso del sistema

Caso de uso: Listar Material

Caso de uso	Listar material
Objetivo	Listar materiales registrados en el sistema.
Actores	Visualizador
Resumen	El caso de uso inicia cuando el visualizador el accede nombre del material, el sistema muestra todas los materiales con la información correspondiente a mostrar, permite ordenar o filtrar por algún dato seleccionado. El caso de uso termina.

Caso de uso: Listar inventario

Caso de uso	Listar inventario
Objetivo	Listar los inventarios realizados, que estén registrados en el sistema.
Actores	Visualizador
Resumen	El caso de uso inicia cuando el actor accede a la opción Listar inventario, el sistema muestra todas las entidades con la información correspondiente a mostrar, permite ordenar o filtrar por algún dato seleccionado. El caso de uso termina.

Caso de uso: Listar medios básicos

Caso de uso	Listar medios básicos
Objetivo	Listar los inventarios realizados, que estén registrados en el sistema.
Actores	Visualizador
Resumen	El caso de uso inicia cuando el actor accede a la opción Listar medios básicos, el sistema muestra todas las entidades con la información correspondiente a mostrar, permite ordenar o filtrar por algún dato seleccionado. El caso de uso termina.

Anexo 2: Descripción ampliada de los casos de uso del sistema

Caso de uso: Listar material

Caso de Uso:	Listar Material
Actores:	Visualizador

Resumen:	El caso de uso inicia cuando el actor accede a la opción Listar material, el sistema muestra todos los materiales con la información correspondiente a mostrar. El caso de uso termina.	
Precondiciones:	El visualizador debe tener permiso para poder acceder al sistema.	
Referencias	RF9	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1.El caso de uso inicia cuando el Visualizador selecciona el material	2.El sistema muestra una lista de las entidades con los datos: <ul style="list-style-type: none"> • Colección • Nivel • Tipo de material • Título • Autor • Fecha de entrada • Cantidad de ejemplares • Estado 	
	2. El caso de uso termina.	
Poscondiciones	Muestra los datos de los materiales.	

Caso de uso: Listar Reservación

Caso de Uso:	Listar reservación
Actores:	Visualizador
Resumen:	El caso de uso inicia cuando el actor accede a la opción Listar reservación, el sistema muestra todas las reservaciones con la información correspondiente a mostrar. El caso de uso termina.
Precondiciones :	El visualizador debe tener permiso para poder acceder al sistema.
Referencias	RF21
Prioridad	Crítico
Flujo Normal de Eventos	

Acción del Actor	Respuesta del Sistema
1. El actor accede a la opción "Generar listado de reservaciones"	2.El sistema muestra un listado con los datos correspondientes a las reservaciones: <ul style="list-style-type: none"> ✓ Nombre ✓ Apellidos ✓ Área ✓ Fecha ✓ Entrada ✓ Salida ✓ Salón
	2. El caso de uso termina.
Poscondiciones	Muestra el listado de reservaciones

Caso de uso: Listar medios básicos

Caso de Uso:	Listar medios básicos
Actores:	Jefe Dpto.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Listar medios básicos, el sistema muestra todos los medios básicos con la información correspondiente a mostrar. El caso de uso termina.
Precondiciones :	El visualizador debe tener permiso para poder acceder al sistema.
Referencias	RF15
Prioridad	Critico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el visualizador selecciona la opción "Listar medio básico".	2. El sistema muestra una lista de las entidades con los datos: <ul style="list-style-type: none"> ✓ Descripción ✓ No inventario ✓ No serie ✓ Fecha de entrada ✓ Estado técnico
	3. El caso de uso termina.
Poscondiciones	Muestra los datos del medio básico

Caso de uso: Listar inventario

Caso de Uso:	Listar inventario
Actores:	Jefe Dpto.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Listar inventario, el sistema muestra todos los inventarios con la información correspondiente a mostrar. El caso de uso termina.
Precondiciones :	El visualizador debe tener permiso para poder acceder al sistema.
Referencias	RF27
Prioridad	Critico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el Visualizador selecciona la opción "Listar inventario".	3. El sistema muestra una lista de las entidades con los datos: <ul style="list-style-type: none"> ✓ Responsable ✓ Fecha ✓ Tipo de inventario ✓ Observaciones
	4. El caso de uso termina.
Poscondiciones	Muestra el listado de los inventarios

Caso de uso: Gestionar reservación

Caso de Uso:	Gestionar reservación
Actores:	Profesor
Resumen:	El caso de uso inicia cuando el profesor entra en el sistema y selecciona la opción gestionar reservación, el sistema permite introducir / eliminar/ modificar/ crear reporte o buscar los datos de la reservación. El sistema confirma que se realizó la reservación. El sistema brinda la posibilidad de generar un reporte con el listado de las reservaciones realizadas, entre varias fechas. El caso de uso termina.
Referencias	RF20, RF22, RF23, RF24, RF25
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el profesor accede a la opción	2. El sistema brinda la posibilidad de adicionar,

Realizar reservación.	modificar o eliminar una reservación.
Sección “Agregar reservación ”	
Acción del Actor	Respuesta del Sistema
3. El profesor escoge la opción “ Agregar reservación “	4. El sistema muestra los datos correspondientes: <ul style="list-style-type: none"> ✓ Nombre ✓ Apellidos ✓ Solapín ✓ Área ✓ Fecha ✓ Entrada ✓ Salida ✓ Salón Y permite Salvar y se guarda la reservación Cancelar ver Alternativa 1: “Cancelar operación”
5. Introduce los datos de la reservación.	6. Valida los datos Si hay datos incompletos, ver Alternativa 2: “Existen campos vacios”. Si hay datos incorrectos, ver Alternativa 3: “Existen campos incorrectos.” 7. Termina el caso de uso.
Flujos Alternos: “Cancelar operación.”	
1. Selecciona la opción de Cancelar operación.	2. Regresa a la vista anterior.
	3. El caso de uso termina.
Flujos Alternos: “Existen campos vacios”	
	1. Muestra un indicador al lado de los campos vacíos.
Flujos Alternos: “Existen campos incorrectos.”	
	1. Muestra el mensaje de error “Existen campos incorrectos”
	2. Muestra un indicador sobre los campos incorrectos.
Sección “Modificar reservación ”	
Acción del Actor	Respuesta del Sistema
1.El actor accede a la opción “Editar reservación”	2.El sistema muestra los datos correspondientes: <ul style="list-style-type: none"> ✓ Nombre ✓ Apellidos ✓ Área ✓ Fecha ✓ Entrada ✓ Salida ✓ Salón Y permite Modificar y se guardan los datos de la reservación Cancelar. Ver Alternativa 1: ”Cancelar operación”

3.Modifica los datos deseados	4. Valida los datos Si hay datos incompletos, ver Alternativa 2: “Existen campos vacios”. Si hay datos incorrectos, ver Alternativa 3: “Existen campos incorrectos.” 5. Termina el caso de uso.
Flujos Alternos: “Cancelar operación”	
1. Selecciona la opción de Cancelar operación.	2. Regresa a la vista anterior.
	3. El caso de uso termina.
Flujos Alternos: “Existen campos vacios”	
	1. Muestra un indicador al lado de los campos vacíos.
Flujos Alternos: “Existen campos incorrectos.”	
	1. Muestra el mensaje de error “Existen campos incorrectos” 2. Muestra un mensaje de error sobre los campos incorrectos.
Sección “Eliminar reservación ”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona una reservación del listado. 2. Escoge la opción “Eliminar”	
	3. Muestra un mensaje para confirmar la acción.
4. En el mensaje escoge Aceptar.	5. Se elimina la reservación del sistema.
Flujos Alternos: “Cancelar operación”	
1. En el mensaje escoge Cancelar.	2. Regresa a la vista anterior. 3. El caso de uso termina.
Sección “Reporte PDF reservación ”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción reporte	2. Muestra un formulario para introducir las fechas entre las cuales se van a buscar las reservaciones realizadas.
3. El actor selecciona la opción Generar PDF	4. El sistema muestra un reporte PDF de las reservaciones realizadas en ese periodo. El caso de uso termina.
Sección “Buscar reservación ”	
Acción del Actor	Respuesta del Sistema

1. El actor selecciona el opción buscar	2. Muestra un formulario para introducir los campos entre las cuales se van a buscar las reservaciones realizadas.
3. El actor selecciona la opción Generar PDF	4. El sistema muestra el listado de las reservaciones realizadas. El caso de uso termina.

Caso de uso: Gestionar Inventario

Caso de Uso:	Gestionar inventario
Actores:	Técnico general
Resumen:	El caso de uso inicia cuando el técnico general accede a la opción Gestionar inventario, el sistema brinda la posibilidad de escoger entre las siguientes opciones agregar/ ver detalles/ crear reporte o buscar inventarios. El técnico general escoge una de las opciones anteriores y registra los datos del inventario en el sistema. El sistema permite generar el inventario en un documento PDF. El caso de uso termina.
Referencias	RF26, RF28, RF29, 52
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el técnico general accede a la opción “Realizar inventario”.	2. El sistema permite escoger el tipo de inventario Ver sección: Inventario de materiales. Ver sección: Inventario de medios básicos.
Sección “Inventario de materiales ”	
Acción del Actor	Respuesta del Sistema
1. El profesor escoge la opción “Inventario materiales”.	2. El sistema muestra los datos correspondientes: ✓ Fecha inventario ✓ Responsable ✓ Observaciones ✓ Listado de materiales Y permite Salvar y se guarda el inventario. Cancelar ver Alternativa 1: “Cancelar operación”
3. Introduce los datos del inventario.	4. Valida los datos Si hay datos incompletos, ver Alternativa 2: “Existen campos vacios”. 5. Si hay datos incorrectos, ver Alternativa 3: “Existen campos incorrectos.” 6. Generar inventario en un documento PDF. 7. Termina el caso de uso.
Flujos Alternos: “Cancelar operación.”	
1. Selecciona la opción de Cancelar operación.	2. Regresa a la vista anterior.

	3. El caso de uso termina.
Flujos Alternos: "Existen campos vacíos"	
	1. Muestra un indicador al lado de los campos vacíos.
Flujos Alternos: "Existen campos incorrectos."	
	1. Muestra el mensaje de error "Existen campos incorrectos"
	2. Muestra un indicador sobre los campos incorrectos.
Sección "Inventario medios básicos "	
Acción del Actor	Respuesta del Sistema
1. El actor accede a la opción "inventario de medios básicos"	2. El sistema muestra los datos correspondientes: <ul style="list-style-type: none"> ✓ Fecha inventario ✓ Responsable ✓ Observaciones ✓ Listado de medios básicos <p>Y permite Salvar y se guarda el inventario Cancelar ver Alternativa 1: "Cancelar operación"</p>
3. Introduce los datos del inventario.	4. Valida los datos Si hay datos incompletos, ver Alternativa 2: "Existen campos vacíos". 5. Si hay datos incorrectos, ver Alternativa 3: "Existen campos incorrectos." 6. Generar inventario en un documento especificado. 7. Termina el caso de uso.
Flujos Alternos: "Cancelar operación"	
1. Selecciona la opción de Cancelar operación.	2. Regresa a la vista anterior.
	1. El caso de uso termina.
Flujos Alternos: "Existen campos vacíos"	
	1. Muestra un indicador al lado de los campos vacíos.
Flujos Alternos: "Existen campos incorrectos."	
	1. Muestra el mensaje de error "Existen campos incorrectos"
	2. Muestra un mensaje de error sobre los campos incorrectos.
Sección "Ver detalles"	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona el inventario deseado y selecciona la opción "Ver"	2. Muestra un formulario con los datos del inventario y un listado de los medios básicos o materiales a los

detalles”	cuales se les hizo el inventario. Termina el caso de uso.
Sección “Crear PDF de inventarios ”	
Acción del Actor	Respuesta del Sistema
5. El actor selecciona el opción reporte	6. Muestra un formulario para introducir las fechas entre las cuales se van a buscar los medios básicos.
7. El actor selecciona la opción Generar PDF	8. El sistema muestra un documento PDF con el listado de inventarios realizados entre dichas fechas. El caso de uso termina.
Sección “Buscar inventarios”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona el opción buscar	2. Muestra un formulario para introducir los campos entre las cuales se van a buscar los inventarios. 3. El sistema muestra el listado de los inventarios. El caso de uso termina.
Poscondiciones	Se realizó el inventario

Anexo 3: Diagramas de clases del análisis

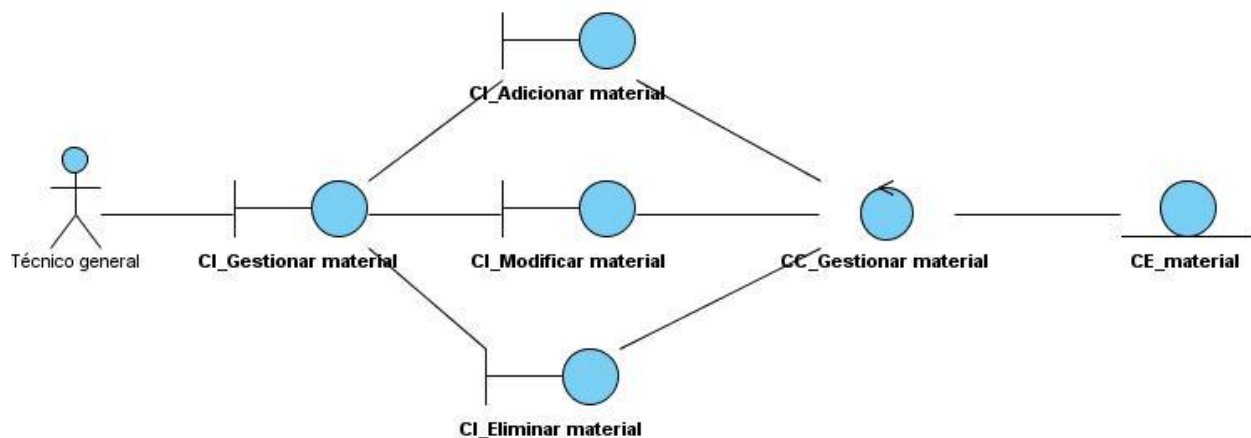


Diagrama de clases del análisis: Gestionar material



Diagrama de clases del análisis: Listar inventario

Anexo 4: Diagramas de clases del diseño

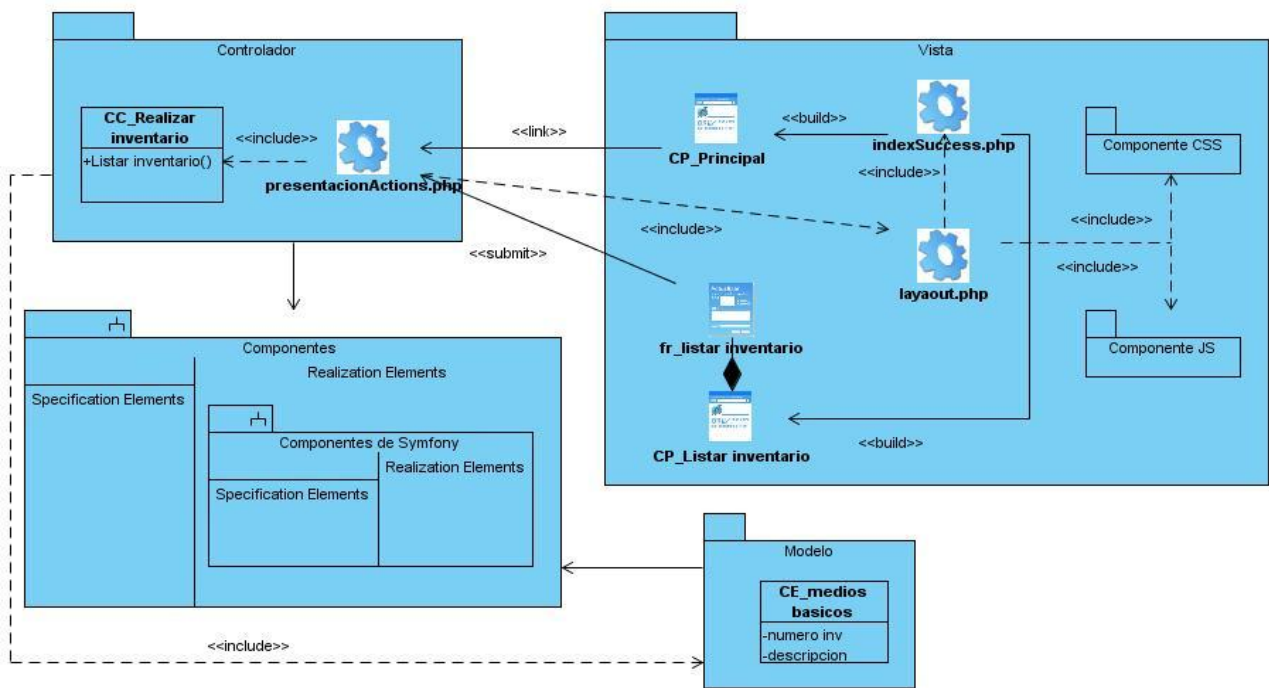


Diagrama de clases del diseño con estereotipos web: Listar inventario

Anexo 5: Diagrama de clases persistentes

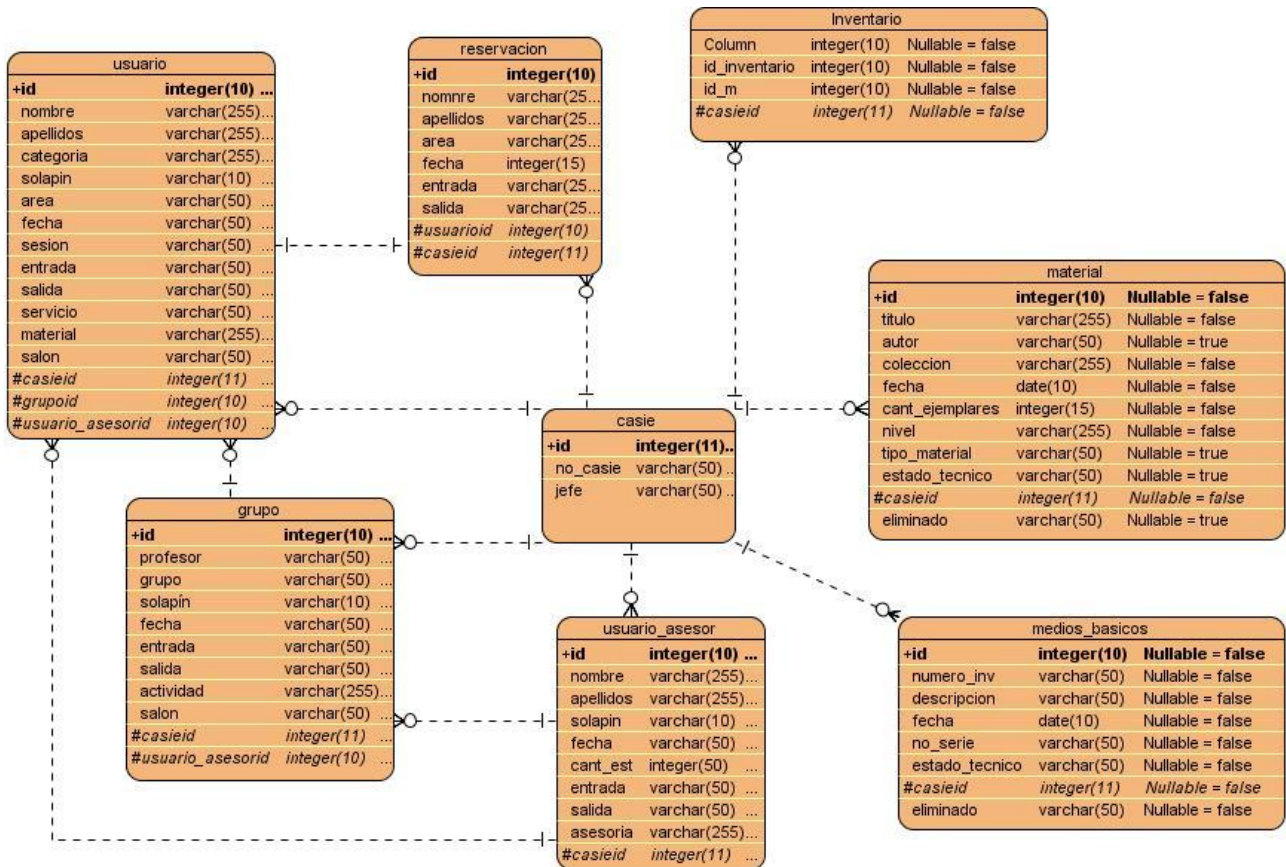


Diagrama de clases persistentes