

Universidad de las Ciencias Informáticas
“Facultad de Telecomunicaciones, Seguridad Informática y
Administración de Redes.”



Título: “Sistema para la automatización del
movimiento de equipos, partes y piezas en el local
de tránsito.”

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor(es): Yomaily Hernández Portilla.

Tutor(es): Deivis Ricardo Álvarez Mendoza.

Co-tutor: Aliennys Mercedes Hurtado.

Junio, 2008
Ciudad de La Habana.

DECLARACIÓN DE AUTORÍA

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

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

Yomaily Hernández Portilla

Deivis R. Álvarez Mendoza.

Firma del Autor

Firma del Tutor

Opinión del usuario del trabajo de diploma.

El Trabajo de Diploma titulado “Sistema para la automatización del movimiento de equipos, partes y piezas del local de tránsito”, fue desarrollado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

- Totalmente
- Parcialmente en un ____ %

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

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

Representante de la entidad

Cargo

Firma

Cuño

Opinión del tutor del trabajo de diploma.

Título: Sistema para la automatización del movimiento de equipos, partes y piezas en el local de tránsito.

Autor: Yomaily Hernández Portilla.

El tutor del presente Trabajo de Diploma considera que durante su ejecución la estudiante mostró las cualidades que a continuación se detallan:

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de ____ .

Firma

_____de junio del 2008.

AGRADECIMIENTOS

Existen muchas personas que durante esta etapa me han brindado su mano para sostenerme, a todas ellas mis más sinceros agradecimientos.

Existen otras que más que una mano han sido mi sostén:

Mi "*mochi*", "*papi*" y "*hermano*" por estar siempre a mi lado.

Jey, Lilo y Yara, por estar presente en los momentos difíciles.

Eternamente a Makay, por ayudarme a convertirme en ingeniera.

A Pavel, por el PostgreSQL.

En fin, a todos los que de forma incondicional me iluminaron con su apoyo.

DEDICATORIA

A mi madre, por ser la mujer más completa de este mundo,

A mi familia, toda, que siempre confió en mí,

A mis compañeros de estudio por estos 5 años de travesía,

A mi tutor,

A mi Patria.

RESUMEN

En la Universidad de Ciencias Informáticas no existe una solución para llevar el proceso de entradas y salidas de equipos, partes y piezas en el local de tránsito perteneciente al grupo de Asistencia Técnica, razón que dificulta el trabajo y funcionamiento en esa área.

Con motivo de automatizar los procesos manuales del local se toma como iniciativa la creación de un sistema para la automatización del movimiento de equipos, partes y piezas en el local de tránsito.

Este sistema representa, sin dudas, una mejora notable en el manejo de información referente a los artículos, control riguroso y actualización inmediata de datos, sustituyendo todo el trabajo manual por tecnología.

Para el desarrollo de la aplicación se modelaron los flujos de trabajo basados en los principios del Proceso de Desarrollo Unificado (RUP) y el Lenguaje de Modelado Visual (UML) y se construyeron los diagramas referentes a cada flujo con la ayuda del Visual Paradigm. De forma similar se utilizan otras tecnologías actuales que brindaron soporte y estabilidad a la aplicación.

PALABRAS CLAVE

Equipos, piezas, partes, local de tránsito, automatización, sistema.

TABLA DE CONTENIDOS

AGRADECIMIENTOS	I
DEDICATORIA	II
RESUMEN.....	III
INTRODUCCIÓN.....	8
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	10
1.2. Estado del arte.....	10
1.2.1. Ámbito internacional.....	11
1.2.2. Estudios en el país.	11
1.2.3. Situación en la UCI.....	12
1.3. Tendencias y tecnologías actuales.	12
1.3.1. Aplicaciones Web.	12
1.3.2. Arquitectura Cliente-Servidor.....	13
1.3.3. Lenguaje PHP.....	14
1.3.4. Servidor web Apache.....	15
1.3.6. Lenguaje JavaScript.	16
1.3.7. Gestor de base de datos PostgreSQL	17
1.4. Patrones de arquitectura.....	18
1.5. Descripción del framework que se utiliza. Codelgniter.....	20
1.6. Metodología empleada en la ingeniería de software.	21
1.6.1. Herramienta y lenguaje aplicado en la metodología.	22
1.7. Herramientas empleadas para la programación y diseño web.....	22
1.8. Conclusiones:.....	22
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	24
2.1. Introducción.	24

2.2. Objeto de estudio.....	24
2.2.1. Problema y situación problemática.....	24
2.2.2. Objeto de automatización.....	25
2.2.3. Información que se maneja.....	25
2.2.4. Propuesta del sistema.....	26
2.3. Modelo del negocio actual.....	27
2.3.1. Reglas del negocio.....	28
2.3.2 Determinación y justificación de actores y trabajadores del negocio.....	29
2.3.3 Diagrama de casos de uso del negocio.....	30
2.3.4 Descripción de los casos de uso del negocio.....	30
2.3.5 Diagrama de actividades.....	31
2.3.6 Modelo de objetos.....	31
2.4 Captura de requerimientos.....	31
2.4.1 Requerimientos funcionales:.....	32
2.4.2 Requerimientos no funcionales.....	33
2.5. Modelo de casos de uso del sistema.....	34
2.5.1. Definición del actor del sistema.....	34
2.5.2 Diagrama de casos de uso del sistema.....	35
2.5.2 Descripción de los casos de uso del sistema.....	35
2.6. Conclusiones.....	36
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	37
3.1. Introducción.....	37
3.2. Análisis.....	37
3.2.1. Definición del modelo de análisis.....	37
3.2.2. Diagramas de clases del análisis.....	37

3.3. Diseño.....	42
3.3.1. Diagramas de interacción.....	43
3.3.1.2. Diagramas de secuencia.....	43
3.3.2. Diagramas de Clases del Diseño por cada caso de uso.....	43
3.4. Diseño de la base de datos.....	49
3.4.1. Modelo lógico de datos.....	49
3.4.2. Modelo físico de datos.....	50
3.4.3. Diagrama de despliegue.....	51
3.5. Conclusiones.....	52
CAPÍTULO 4: IMPLEMENTACIÓN.....	53
4.1. Introducción.....	53
4.2. Diagrama de componentes.....	53
4.3 Conclusiones.....	55
CAPÍTULO 5: ESTUDIO DE LA FACTIBILIDAD.....	56
5.1. Introducción.....	56
5.2. Planificación basada en puntos de Casos de Uso.....	56
5.3. Beneficios tangibles e intangibles.....	63
5.4. Análisis de costos y beneficios.....	63
5.5. Conclusiones.....	63
CONCLUSIONES.....	64
RECOMENDACIONES.....	65
BIBLIOGRAFÍA.....	66
ANEXOS.....	67
Anexo I.....	67
Anexo II.....	70

Anexo III.....	73
Anexo IV.....	74
Anexo V:.....	90
GLOSARIO.....	100

INTRODUCCIÓN

La Universidad de las Ciencias Informáticas (UCI) es una universidad de nuevo tipo con una dinámica diferente a la del resto de las universidades del país.

Surgida en el año 2002 y con el objetivo de crear profesionales en el campo de las ciencias de la informática e ingenieros integrales comprometidos con los principios de la Revolución, sus necesidades de avance y de colocarse en el mercado mundial del software, la UCI ha ido informatizando cada uno de los servicios que en esta se ofrecen.

Uno de estos servicios es el manejo de equipos, piezas y partes de computadoras. La gestión de estos artículos corresponde al local de tránsito perteneciente al Grupo de Asistencia Técnica (GAT).

Este proceso se divide en dos actividades fundamentales: la de control de entradas y la de control de salidas. Ambas consisten en el almacenamiento y actualización de datos de los artículos. De cada pieza, parte o equipo que existe en el local se va a registrar su salida, ya sea por préstamo de artículos a un cliente o por un documento que solicite artículos. De igual forma se registrarán las entradas, las que pueden ser compra, abastecimiento del almacén central o simplemente cuando se realiza la devolución de un préstamo.

Estas funciones son desempeñadas manualmente, lo cual no proporciona una actualización inmediata de los cambios que se realizan, limita la interactividad de los datos, propicia demoras de entrega de información y en un sentido más general viola los principios de seguridad, confidencialidad, usabilidad y disponibilidad concurrente de la gran cantidad de información que constantemente se requiere manejar.

Las diversas problemáticas planteadas conllevan a determinar como nuestro **problema científico** el *¿Cómo automatizar el proceso de control de entradas y salidas de equipos, piezas y partes de computadoras del local de tránsito?* Definiendo como **objeto de estudio** *los procesos referentes a gestión de entradas y salidas de equipos, piezas y partes de computadoras en el local de tránsito*, de igual modo se toma como **campo de acción** *la automatización de los procesos referentes a gestión de entradas y salidas de de equipos, piezas y partes de computadoras en el local de tránsito de la UCI.*

De esta forma el **Objetivo general** consiste en diseñar e implementar una aplicación capaz de gestionar el funcionamiento del local de tránsito. De ahí los siguientes **objetivos específicos**:

- Realizar un estudio de la estructura actual del local de tránsito.
- Declarar los requisitos que debe cumplir el módulo a implementar.
- Descripción de los procesos que se van a implementar en el módulo.
- Modelar conceptualmente las clases y diagramas que están implicados en el módulo.
- Diseñar e implementar el módulo necesario para el funcionamiento adecuado de la aplicación.

Analizando el tema en cuestión se pretende que con el desarrollo de una aplicación que gestione todas estas operaciones de entrada/salida se incremente la eficiencia, organización, control de piezas, equipos y partes de computadoras; reflejando de esta forma la **idea a defender**.

El presente documento se encuentra estructurado en capítulos como se describe a continuación:

Capítulo 1: Describe la fundamentación teórica de la realización del trabajo de diploma. Incluye un estado del arte del tema tratado, a nivel internacional, nacional y de la Universidad, describe las tendencias, tecnologías, metodologías y software usados en la actualidad y en las que se apoya para la solución del problema.

Capítulo 2: Ofrece una descripción detallada del negocio y se realiza el análisis del sistema a desarrollar. Se definen las principales funcionalidades describiéndolas detalladamente mediante herramientas de modelación y descripciones de casos de uso.

Capítulo 3: Se describen las etapas de análisis y diseño. Aborda aspectos relacionados con la construcción de la solución propuesta, se modelan los diagramas de clases de análisis y de diseño, se construyen los diagramas de interacción (colaboración y secuencia), los modelos de datos: físico y lógico, culminando el capítulo con la construcción del diagrama de despliegue.

Capítulo 4: Se abordan todos los aspectos relacionados con la implementación del sistema, representándose el diagrama de componentes.

Capítulo 5: Se estudia la factibilidad de la construcción del sistema, enunciándose los beneficios tangibles e intangibles así como un análisis de costo beneficio.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción.

El objetivo fundamental de la documentación de este capítulo es abordar de forma detallada el soporte teórico para la aplicación web. Se enfatiza, además, en la necesidad de la creación de un software que organice y controle el funcionamiento integral del local de tránsito así como en el estudio de los conceptos, lenguajes de programación, herramientas de trabajo y tecnologías a utilizar con el propósito de obtener finalmente un producto depurado y con la calidad requerida.

1.2. Estado del arte.

La aplicación web que se desea desarrollar tiene como precedente el sitio web www.GATServer.uci.cu, perteneciente al Grupo de Asistencia Técnica. Esta aplicación que nace en diciembre del año 2004 implementada sobre ASP y con SQL Server 2000 como gestor de base de datos se encarga de ofrecer servicios online de solicitud de reportes y reclamación.

Actualmente esta aplicación será modificada y re implementada esta vez utilizando el framework CodeIgniter y Apache como servidor Web. Esta idea se toma con el fin de mejorar el servicio que ofrece el sitio y fortalecer sus debilidades, haciendo con esta nueva versión un entorno más amigable y flexible para el usuario y eliminando la programación de códigos innecesarios.

Como resultado de varios años de investigación y desarrollo se ha creado un sistema informático capaz de cubrir las necesidades de cualquier taller de reparaciones. Este producto denominado S.A.T. (Sistema Administrativo para Talleres), ha sido diseñado pensando en la estandarización, la adaptabilidad, y con una muy amplia capacidad de configuración y personalización.

El Sistema de gestión de talleres SAT, es un software de gestión y administración para talleres y servicios técnicos autorizados.

Nace a partir de la necesidad de los establecimientos de servicio técnico de informatizar la administración de sus empresas, para mejorar el servicio, minimizar sus tiempos de respuesta, tener mayor control del taller, de los equipos y de los repuestos para las reparaciones, mejorar el trato y la comunicación con el cliente, organizar las tareas y calcular costos para elegir las mejores opciones de compra.

El sistema SAT es aplicable a todo tipo de talleres y en la propia instalación del producto puede seleccionar si su taller es de artículos electrónicos, de electrodomésticos, computación, motos, autos, fotografía, telefonía, etc. Además, gracias a las capacidades configuración y adaptabilidad del sistema, usted podrá fácilmente aplicarlo a cualquier otro rubro de reparaciones y ventas.

SAT ofrece servicios de desarrollo de sistemas, diseño web, capacitación y asistencia técnica. En este último se ofrece una gama de soluciones a problemas de software y hardware que un usuario pueda presentar y brindando la solución correcta al problema por parte del personal calificado.

1.2.1. Ámbito internacional.

Internacionalmente se puede referenciar nuevamente al Sistema Administrativo de Talleres SAT.

SatNetwork ofrece a sus clientes el desarrollo e implementación de sistemas de computación a medida. Para brindarles el mejor producto cuenta con herramientas avanzadas de programación y equipamiento de última tecnología. El desarrollo de sus productos se realiza con lenguajes visuales, lo que da como resultado programas de muy fácil manejo, con funcionalidad muy intuitiva.

Este un sistema testeado disímiles veces para llegar de forma certera y concreta al cliente y cubriendo sus necesidades.

He aquí lo que impulsa al desarrollo de una aplicación web para el local de tránsito sobre algunos de los principios y servicios que ofrece SAT.

1.2.2. Estudios en el país.

En Cuba no contamos con una aplicación que realice exactamente las mismas funciones de la aplicación que se desea obtener para el local de tránsito, debido a que se va a implementar para una universidad única de su tipo en el país y que no tiene precedencia de similares con sus características. Se puede hacer referencia a sitios que controlen productos por categorías y conserven listados de los mismos y que a su vez los muestren a los clientes de forma online. Uno de estos sitios es el de la empresa cubana Copextel.

La Corporación Cubana Copextel es una sociedad mercantil y de servicios ingenieros, con más de 15 años de trabajo y una amplia y diversificada red de distribución y servicios ingenieros en todo el territorio nacional.

Se distingue por ser un proveedor de soluciones integrales y contando con el personal especializado capaz de asumir con profesionalidad cada uno de sus proyectos.

1.2.3. Situación en la UCI.

En la Universidad no existe una aplicación web en el local de tránsito que recoja detalladamente las funciones que se realizan en el mismo. Tampoco se ha creado un módulo integrado a la aplicación del Grupo de Asistencia Técnica, conocida como GATServer, donde se gestionen piezas, equipos y partes. Todas las operaciones que en dicho local se desempeñan se hacen de forma manual mediante reportes y tarjetas de estiba lo que indudablemente dificulta el control de piezas, equipos y/o partes de computadoras por lo que se hace imprescindible la creación de una aplicación donde se registren todas estas actividades.

1.3. Tendencias y tecnologías actuales.

1.3.1. Aplicaciones Web.

Una aplicación web es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet. Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero. La facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. Aplicaciones como los webmails, wikis, weblogs, tiendas en línea y la Wikipedia misma son ejemplos bien conocidos de aplicaciones web.

Aunque muchas variaciones son posibles, una aplicación web está comúnmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web es la primera capa, un motor usando alguna tecnología web dinámica (ejemplo: CGI, PHP, Java Servlets o ASP) es la capa de en medio, y una base de datos como última capa. El navegador web manda peticiones a la capa media, que la entrega valiéndose de consultas y actualizaciones a la base de datos generando una interfaz de usuario.

En tiempos recientes se ha usado la estrategia de generalizar esta arquitectura mediante la adición de piezas de hardware que permiten balancear la carga de los servidores web y de aplicación.

Existen numerosos lenguajes de programación empleados para el desarrollo de Aplicaciones Web, entre los que destacan:

- PHP
- ASP/ASP.NET
- Java, con sus tecnologías Java Servlets y JavaServer Pages (JSP)
- Perl
- Ruby
- Python
- HTML
- XML (1)

1.3.2. Arquitectura Cliente-Servidor.

Los elementos principales de la arquitectura cliente servidor son justamente el elemento llamado cliente y el otro elemento llamado servidor.

Esta arquitectura se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes.

Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos (SGBD).

Por otro lado los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red.

Este tipo de arquitectura es la más utilizada en la actualidad, debido a que es la más avanzada y la que mejor ha evolucionado en estos últimos años.

Podemos decir que esta arquitectura necesita tres tipos de software para su correcto funcionamiento:

- Software de gestión de datos: Este software se encarga de la manipulación y gestión de los datos almacenados y requeridos por las diferentes aplicaciones. Normalmente este software se aloja en el servidor.

- Software de desarrollo: este tipo de software se aloja en los clientes y solo en aquellos que se dedique al desarrollo de aplicaciones.
- Software de interacción con los usuarios: También reside en los clientes y es la aplicación gráfica de usuario para la manipulación de datos, siempre claro a nivel usuario (consultas principalmente).

A parte de estos existen más aplicaciones software para el correcto funcionamiento de esta arquitectura pero ya están condicionados por el tipo de sistema operativo instalado, el tipo de red en la que se encuentra, etc. (2)

1.3.3. Lenguaje PHP.

El lenguaje PHP es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles, funciones. No es un lenguaje de marcas como podría ser HTML, XML o WML. (3)

A diferencia de otros lenguajes que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP. (4)

PHP es un lenguaje de programación interpretado usado normalmente para la creación de páginas web dinámicas. PHP es un acrónimo recursivo que significa "PHP Hypertext Pre-processor" (inicialmente PHP Tools, o, Personal Home Page Tools).

Se escoge este lenguaje de programación debido a las ventajas que ofrece:

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones.

1.3.4. Servidor web Apache

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

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

Apache es escogido debido a que ofrece ventajas favorables para el producto, como:

- Modular.
- Open source.
- Multi-plataforma.
- Extensible.
- Popular (fácil conseguir ayuda/soporte).
- Gratuito.

Además de poseer una arquitectura muy modular, una sección *core* y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor web. Algunos de estos módulos son:

- **mod_ssl** - Comunicaciones Seguras vía TLS.
- **mod_rewrite** - Reescritura de direcciones (generalmente utilizado para transformar páginas dinámicas como php en páginas estáticas html para así engañar a los navegantes o a los motores de búsqueda en cuanto a como fueron desarrolladas estas páginas).
- **mod_auth_ldap** - Permite autenticar usuarios contra un servidor LDAP. (4)

1.3.6. Lenguaje JavaScript.

Es un Lenguaje de programación interpretado, o sea, no requiere compilación. Es utilizado especialmente en páginas Web embebido en el código HTML o similares. La mayoría de los navegadores pueden interpretar los códigos JavaScript incluidos en las páginas Web.

JavaScript es un lenguaje basado en prototipos, pues las nuevas clases se generan clonando las clases base (prototipos) y extendiendo sus funcionalidades

Este lenguaje se utiliza para las validaciones en la parte cliente del sistema. (5)

1.3.7. Gestor de base de datos PostgreSQL.

PostgreSQL es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD. Funciona en todos los sistemas operativos importantes, incluyendo Linux, UNIX y Windows. Tiene soporte total para foreign keys (llaves foráneas), joins (enlaces), views (vistas), triggers (disparadores), y stored procedures (procedimientos almacenados) en múltiples lenguajes. Tiene interfaces de programación nativos para C/ C++, Java, Perl, Python, Ruby, Tcl, ODBC, entre otros.

Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). (6)

Se escoge PostgreSQL para realizar este sistema porque posee alta concurrencia pues permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo a la tabla. Es extensible, pues el código fuente está disponible para todos sin costo.

A pesar de que este SGBD posee tan buenas características posee una pequeña deficiencia que no es tan notable o mejor dicho es bueno en proyectos grandes, ésta es la velocidad de respuesta que ofrece este gestor con bases de datos relativamente pequeñas puede parecer un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes, cosa que resulta loable.

1.4. Patrones de arquitectura.

Un patrón de arquitectura de software describe un problema particular y recurrente del diseño, que surge en un contexto específico, y presenta un esquema genérico y probado de su solución. (7)

Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma.

El patrón de la arquitectura por el que se regirá el sistema es el Modelo-Vista-Controlador o MVC.

La arquitectura MVC fue diseñada para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

Este modelo de arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa; lo cual nos permite implementarlos por separado.
- Hay un API muy bien definido; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Al incorporar el modelo de arquitectura MVC a un diseño, las piezas de un programa se pueden construir por separado y luego unir las en tiempo de ejecución. Si uno de los Componentes, posteriormente, se observa que funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas. Este escenario contrasta con la aproximación monolítica típica de muchos programas Java. Todos tienen un Frame que contiene todos los elementos, un controlador de eventos, un montón de cálculos y la presentación del resultado. Ante esta perspectiva, hacer un cambio aquí no es nada trivial.

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.

La Vista es 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.

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.

Entre las ventajas del estilo Modelo-Vista-Controlador están las siguientes:

Soporte de múltiples vistas: Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos. (8)

El MVC es un patrón ampliamente utilizado en múltiples plataformas y lenguajes. El sistema se implementó basado en este patrón de arquitectura debido a que posee:

- Mayor facilidad para el desarrollo de clientes ricos en múltiples dispositivos y canales.
- Más claridad de diseño.
- Facilita el mantenimiento.
- Mayor escalabilidad.

1.5. Descripción del framework que se utiliza. CodeIgniter.

El término framework se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se puede añadir las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un framework Web, por tanto, se define como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de aplicaciones Web.

El CodeIgniter es el framework para PHP seleccionado para realizar esta aplicación Web. Está pensado para ofrecer un alto rendimiento, ser ligero y fácilmente instalable (puede usarse en un alojamiento compartido y no es necesario tener acceso a la línea de comando). Además, gracias a la documentación y los foros (en inglés ambos), el aprendizaje es muy rápido por lo que pueden hacerse cosas muy interesantes.

Incluye varias librerías para gestionar el acceso a datos, sesiones de usuarios, formularios, seguridad, etc. Además la comunidad de usuarios ha creado una serie de plugins, clases y librerías para extenderlo que lo hacen aún más interesante.

1.6. Metodología empleada en la ingeniería de software.

El Proceso Unificado de Desarrollo (*Rational Unified Process* en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. (9)

Esta fue la metodología que se escogió para el análisis porque está basado en 5 principios clave que son:

- Adaptar el proceso.
- Balancear prioridades.
- Demostrar valor iterativamente.
- Elevar el nivel de abstracción.
- Enfocarse en la calidad.

El RUP, además, divide el proceso de desarrollo en ciclos, teniendo un producto final al culminar cada una de ellos, estos a la vez se dividen en fases que finalizan con un hito donde se debe tomar una decisión importante:

- Concepción
- Elaboración
- Construcción
- Transición
- Mantenimiento

1.6.1. Herramienta y lenguaje aplicado en la metodología.

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. (10) Es un galardonado producto que facilita a las organizaciones el diseño visual y el diagrama, integrar y desplegar sus aplicaciones empresariales de misión crítica y sus bases de datos subyacentes. La herramienta de desarrollo de software ayuda a su equipo a sobresalir todo el modelo de construcción-desplegar software de proceso de desarrollo, y aumentar al máximo la aceleración de ambos equipos y de los individuos.

Se escoge este software de modelado UML porque 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.

Soporta un conjunto de lenguas, tanto en la generación de código y de ingeniería inversa sobre Java, C + +, CORBA IDL, PHP, XML Schema, Ada y Python. Además, apoya la generación de código C#, VB.Net, Object Definición Idioma (EAD), Flash ActionScript, Delphi, Perl, Objective-C, y Ruby. Ingeniería Inversa también apoya clase Java, .NET dll y exe, JDBC, y de los archivos de mapeo Hibernate.

1.7. Herramientas empleadas para la programación y diseño web.

Para el desarrollo de la aplicación se empleo:

- Dreamweaver 8: Herramienta de desarrollo Web que permite a sus usuarios diseñar, desarrollar y mantener de forma eficaz sitios y aplicaciones Web basadas en normas.
- Zend Studio 5.5.0: Editor de texto para paginas PHP que proporciona un buen número de ayudas, desde la creación y gestión de proyectos hasta la depuración del código.

1.8. Conclusiones:

En este capítulo se expusieron los conceptos mas importantes relacionados con el desarrollo de la aplicación. Se exponen y justifican todas las herramientas, lenguajes y metodologías que se escogen

para el desarrollo del software con el fin de lograr un producto de calidad, de fácil manejo y entendimiento para el usuario.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción.

Este capítulo se crea con el objetivo de comprender la estructura del local en el que se va a implementar el sistema e informatizar el flujo de trabajo actual.

En el presente capítulo expone el modelo del negocio, describiendo así los procesos existentes u observados con el objetivo de comprenderlos. Toda la información que se muestra será expresada mediante actores, casos de uso y trabajadores. Se realiza una descripción general de la propuesta del sistema especificando como debe funcionar. Se muestran, además la especificación de requerimientos que el sistema debe poseer definiéndose los casos de uso del sistema y la relación con los actores lo que constituye el diagrama de casos de uso del sistema.

2.2. Objeto de estudio.

2.2.1. Problema y situación problemática.

La gestión de equipos, piezas y partes de computadoras en la UCI corresponde al local de tránsito perteneciente al Grupo de Asistencia Técnica (GAT).

Este proceso se divide en dos actividades fundamentales: la de control de entradas y la de control de salidas. Ambas consisten en el almacenamiento de los datos de cada artículo. De cada pieza, parte o equipo que existe en el local se va a registrar su salida, ya sea por respuesta a una solicitud de préstamo que hace un cliente o a un documento que solicite un artículo determinado. De forma similar sucede cuando hay un abastecimiento de artículos al local o existe una devolución de un préstamo.

Estas funciones son desempeñadas manualmente, lo cual no proporciona una actualización inmediata de los cambios que se realizan, limita la interactividad de los datos, propicia demoras de entrega de información y en un sentido más general viola los principios de seguridad de confidencialidad, usabilidad y disponibilidad concurrente de la gran cantidad de información que constantemente se requiere manejar.

2.2.2. Objeto de automatización.

Tomando en cuenta la situación problemática se hace necesario desarrollar un sistema capaz de realizar todos los procesos que se ejecutan de forma manual. Entiéndase por esto una aplicación que controle, archive y muestre las acciones de entradas y salidas que tienen lugar en el local de tránsito, que registre el documento que justifica estas operaciones y capaz de gestionar los préstamos, devoluciones y de calcular el saldo resultante después de cada operación.

2.2.3. Información que se maneja.

Se controla toda la información referente a las piezas, partes y equipos que existen y que periódicamente están entrando y saliendo del local de tránsito.

Se manipulan documentos que justifican la entrada al local mediante acciones de compra o abastecimiento, así como órdenes de servicios, reportes y otros para respaldar una salida de una pieza, parte o equipo del local hacia cualquier lugar de la universidad. Además de tener un control riguroso sobre los préstamos de herramientas y las devoluciones de estas.

2.2.4. Propuesta del sistema.

Para lograr una mayor eficiencia en el trabajo y tener mejor control y acceso a la información que se maneja en el local de tránsito se va a desarrollar un sistema que solucione las problemáticas antes mencionadas. El sistema que se desarrollará brindará opciones como:

- Autenticarse: Permitirá al usuario acceder al sistema y trabajar con las funcionalidades que este ofrecerá para facilitar el trabajo y control de artículos en el local de tránsito.
- Chequear disponibilidad de un artículo: Donde la trabajadora del local puede ver si el artículo que está solicitando el cliente está disponible o no, o sea si se encuentra prestado o su disponibilidad es nula.
- Gestionar artículo: Permite a la trabajadora del local insertar, o mostrar una lista de artículos, modificar los datos de un artículo que ella desee o eliminarlo del sistema cuando este salga del local de tránsito.
- Gestionar tarjeta de estiba: Con esta opción se controlará todo el movimiento de entrada y salida que tengan los artículos.
- Gestionar préstamo: Esta opción brindará la posibilidad de poder insertar un préstamo en el sistema, modificarlo, eliminarlo del sistema o mostrar todos los préstamos que se han realizado.
- Gestionar devolución: En esta sección se registrarán las devoluciones que haga un cliente, brindando las funcionalidades necesarias para modificarla, mostrar todas las devoluciones o simplemente eliminarlas del sistema cuando el actor lo desee.

El desarrollo de este sistema constituye un beneficio para la trabajadora del local de tránsito, pues se informatizarán todas las actividades que hoy día se realizan de forma manual.

2.3. Modelo del negocio actual.

Actualmente en el local de tránsito las solicitudes de partes y/o equipos se hacen mediante ordenes de servicio emitidos por el técnico del área. Estas órdenes son almacenadas y al artículo que se le da salida se rebaja su existencia de una tarjeta de estiba donde se controla la cantidad real. De forma similar sucede con los préstamos y las devoluciones que realiza un cliente, solo que estas se registran en un libro.

Cuando se produce una entrada al local estas tarjetas se deben incrementar o crear en dependencia del artículo, si este está roto entonces pasará al taller de reparaciones. Cuando lo que se produce es una salida se rebaja la cantidad de artículos de la tarjeta, permitiendo controlar la cantidad real.

2.3.1. Reglas del negocio

Las reglas de negocio describen políticas que deben cumplirse o condiciones que se deben satisfacer, por lo que regulan algún aspecto del negocio.

- Entrada al local
 1. La entrada al local de piezas, partes y equipos deberá ser registrada indicando la cantidad de artículos, nombre y una breve descripción.
 2. La trabajadora del local es la persona autorizada para manejar toda la información referente a los artículos que arriban al local.
 3. Solo se recibirán artículos en el local acompañados de un documento que respalde la acción.
- Salida del local
 1. La salida de artículos del local de tránsito solo se efectuarán mediante la emisión de una orden de servicio u otro documento que avale tal acción.
 2. La trabajadora del local de tránsito es la única persona autorizada para dar salida a un artículo de dicho local.
 3. La salida del local de piezas, partes y equipos deberá ser registrada indicando la cantidad de artículos, nombre y una breve descripción.
- Préstamos.
 1. Los préstamos de piezas solo serán registrados y efectuados por la trabajadora del local.
 2. Un préstamo efectuado deberá incluir los datos del cliente que lo solicita, una descripción de la pieza solicitada y la fecha y plazo del préstamo.
 3. El préstamo puede incluir cualquier cantidad de piezas que el cliente necesite siempre que se registren los datos de las mismas.

- Devoluciones.
 1. Las devoluciones solo serán recibidas y registradas por la trabajadora del local.
 2. En una devolución se debe verificar que el artículo que el cliente va a entregar corresponda con el que se le prestó.

2.3.2 Determinación y justificación de actores y trabajadores del negocio.

Actor del negocio	Descripción
Cliente	Solicita préstamos de piezas. Realiza la devolución de préstamos. Solicita una salida del local de partes o equipos mediante ordenes de servicio, reportes de rotura u otro documento.

Tabla. 1: Definición del actor 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. Representa un rol.

Trabajador del negocio	Descripción
Contadora del local.	Gestiona entrada y salida de artículos mediante documentos, préstamos y devoluciones.

Tabla. 2: Definición del trabajador del negocio.

2.3.3 Diagrama de casos de uso del negocio.

El diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio.

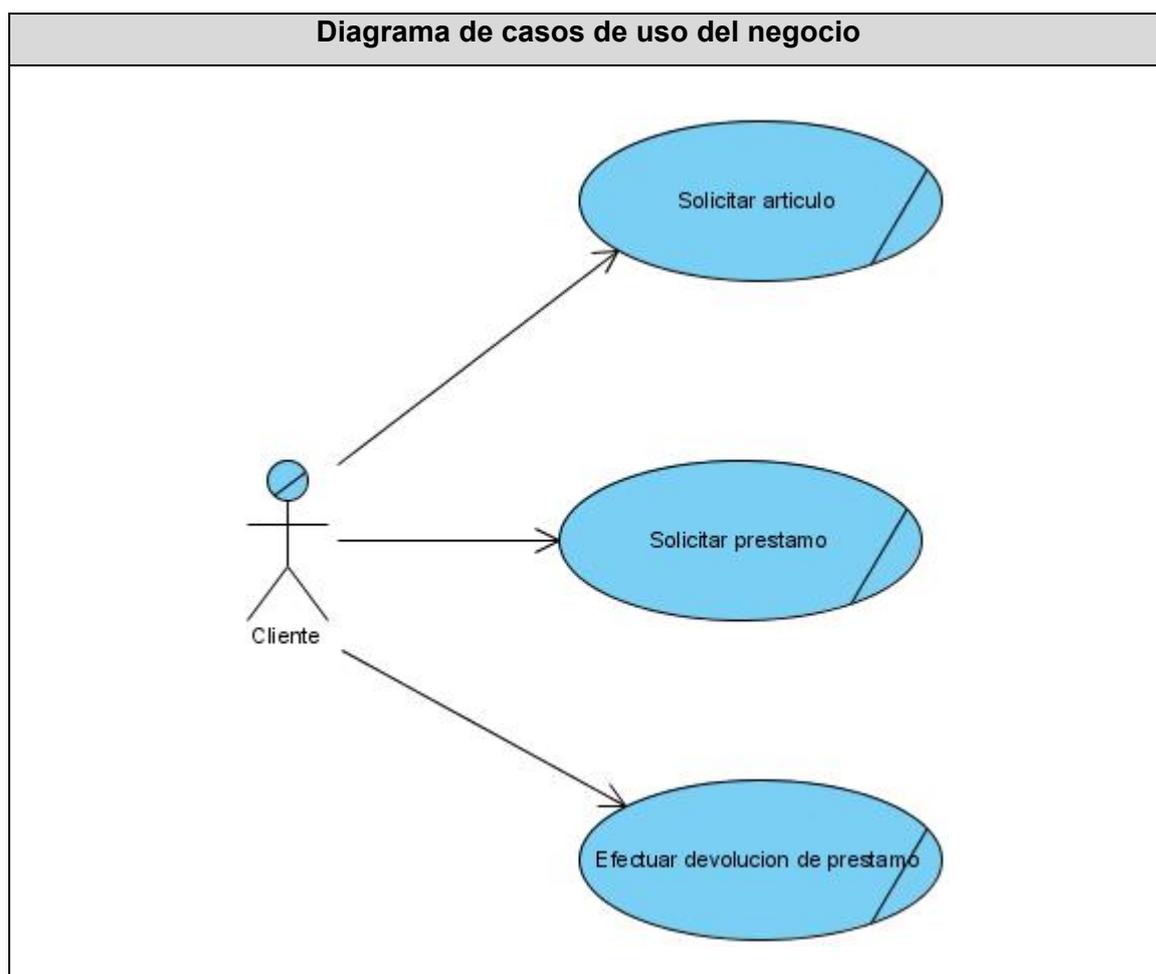


Figura 1: Diagramas de casos de uso del Negocio.

2.3.4 Descripción de los casos de uso del negocio.

La descripción de un caso de usos del negocio muestra cómo colaboran los trabajadores y entidades del negocio para ejecutar el proceso. Son reseñas textuales del caso de uso y explica los procesos o actividades que tienen lugar en el mismo. (Ver anexo 1)

2.3.5 Diagrama de actividades.

El diagrama de actividad es un grafo de actividades que contiene estados en que puede hallarse una actividad. (Ver anexo 2).

2.3.6 Modelo de objetos.

Los modelos de objetos del dominio están asociados a cada uno de los casos de uso del negocio. (Ver anexo 3).

2.4 Captura de requerimientos.

Se define un requerimiento como:

Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.

2.4.1 Requerimientos funcionales:

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

R₁- Autenticar usuario.

R_{1.1}. Verificar usuario y contraseña

R_{1.2}. Otorgar privilegios.

R₂- Gestionar articulo.

R_{2.1}.Insertar artículo.

R_{2.2}.Mostrar datos de artículo.

R_{2.3}.Modificar datos de artículo.

R_{2.4}.Eliminar artículo.

R₃- Chequear disponibilidad de artículo.

R_{3.1}. Buscar artículo en la base de datos del local.

R_{3.2}. Mostrar disponibilidad de un artículo.

R₄- Gestionar préstamo.

R_{4.1}. Insertar préstamo.

R_{4.2}. Mostrar datos del préstamo.

R_{4.3}. Modificar datos del préstamo.

R_{4.4}. Eliminar préstamo.

R₅- Gestionar tarjeta de estiba.

R_{5.1}. Agregar una acción a la tarjeta.

R_{5.2}. Mostrar datos de una acción.

R_{5.3}. Modificar datos de acción.

R_{5.4}. Eliminar acción.

R₆- Registrar documento.

2.4.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

- **Usabilidad:** El futuro usuario del sistema debe poseer conocimientos básicos de computación. El producto debe estar concebido para que el usuario piense que desea hacer y no como hacerlo.
- **Seguridad:** Toda la información guardada referente a los artículos será confidencial, evitando de esta forma su divulgación y que sea consultada por personal no autorizado.
- **Confiabilidad:** El sistema debe funcionar las 24 horas del día y proporcionar salidas precisas.
- **Rendimiento:** Como aplicación de tiempo real, debe tener alto grado de velocidad de procesamiento o cálculo, tiempo de respuesta y de recuperación, y disponibilidad.
- **Diseño e implementación:** Se implementará en Lenguaje PHP, se regirá por la filosofía de Programación Orientada a Objetos. El análisis y el diseño del sistema llevarán la metodología RUP, lenguaje de modelación UML y la herramienta Visual Paradigm. Gestor de base de datos se utilizara PostgreSQL y como servidor web Apache.
- **Portabilidad:** El producto deberá ser multiplataforma.
- **Software:** Para una correcta funcionalidad de la aplicación se necesita de máquina con los siguientes programas instalados:
 - Apache (servidor web).
 - PostgreSQL (gestor de base de datos).
- **Hardware:** La aplicación creada funcionará sobre una computadora con los siguientes requerimientos, como mínimo debe poseer:
 - 256 de RAM.
 - 3 Gb de espacio libre para la BD.

- **Interfaz:** Las funciones de trabajo con la aplicación serán sencillas. La interfaz debe ser amigable y de fácil comprensión visual.

2.5. Modelo de casos de uso del sistema.

2.5.1.

Actor del sistema	Justificación
Contadora.	Responsable de controlar todo el proceso de entrada y salida de piezas, partes y equipos al local de tránsito. Persona autorizada para gestionar los artículos existentes en el local. Persona que gestiona los préstamos y las devoluciones que se realizan.

Tabla. 3: Definición del actor del sistema Definición del actor del sistema.

2.5.2 Diagrama de casos de uso del sistema

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores.

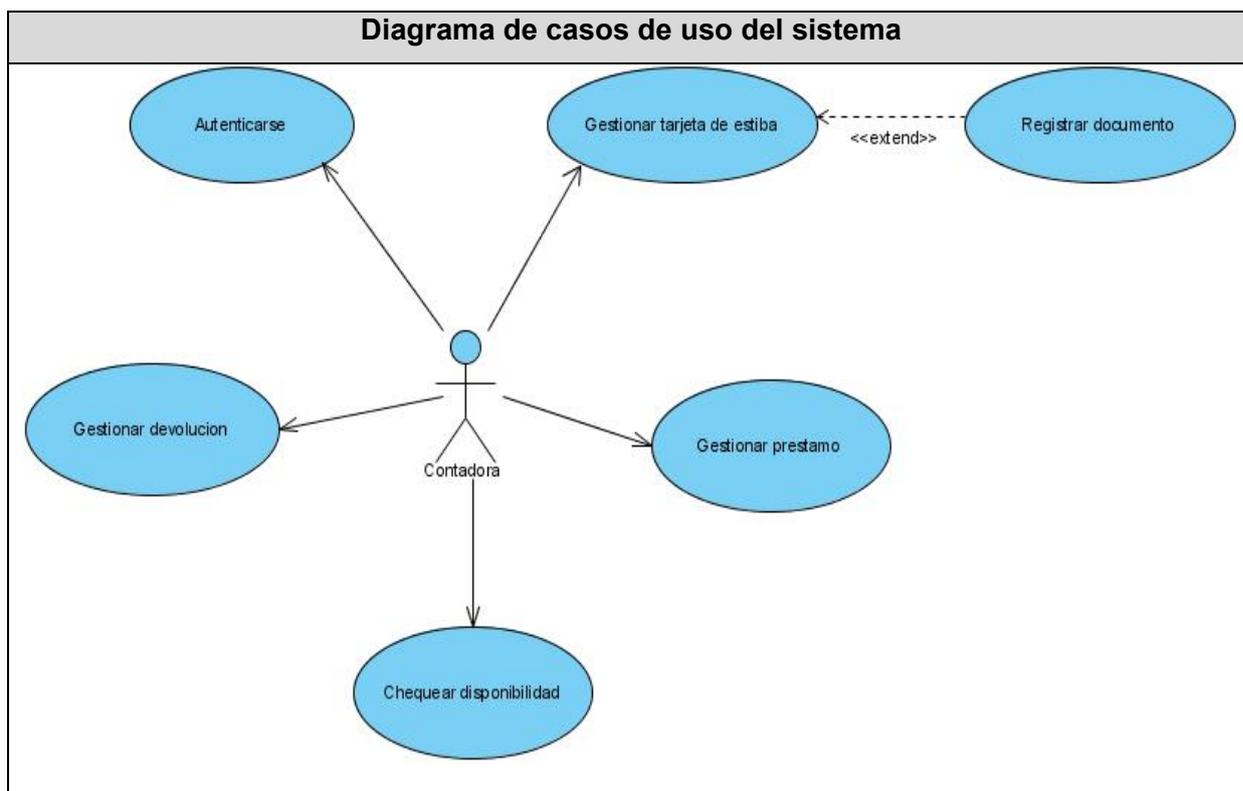


Figura 2: Diagrama de casos de uso del Sistema.

2.5.2 Descripción de los casos de uso del sistema.

Los casos de uso son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa, un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia. (Ver anexo 4).

2.6. Conclusiones

En este capítulo se identificaron, enunciaron y clasificaron los requerimientos que el sistema debe cumplir. Se hace una descripción del negocio a automatizar con el objetivo de brindar un enfoque claro de la problemática existente. Se obtuvieron las propiedades o cualidades que el producto debe tener y un listado de las capacidades o funciones que el sistema debe cumplir.

El resultado del desarrollo del capítulo ha sido la obtención de una vista externa del sistema, que describe lo que se espera de él a través del diagrama de caso de uso del sistema.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1. *Introducción.*

El objetivo de este capítulo es traducir los requisitos a una especificación que describe cómo implementar el sistema. Consiste en obtener una visión del sistema que se preocupa de ver QUÉ hace. Con este capítulo debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

3.2. *Análisis*

3.2.1. Definición del modelo de análisis.

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas. En general se siguen directrices muy parecidas a las que usamos en la construcción del modelo conceptual.

Los propósitos a conseguir con el Análisis son los siguientes:

- Conseguir una comprensión más precisa de los requisitos, refinarlos y estructurarlos.
- Utilizar el lenguaje de los desarrolladores para analizar con profundidad los requisitos funcionales.
- Proporcionar una visión general del sistema.

3.2.2. Diagramas de clases del análisis.

El diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema.

Para la elaboración de los diagramas de clases del análisis se han utilizado tres tipos de clases:

- Entidad: Para modelar información que posee larga vida y que es a menudo persistente.
- Interfaz: Para modelar la interacción entre el sistema y sus actores.
- Control: Para coordinar la realización de los casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

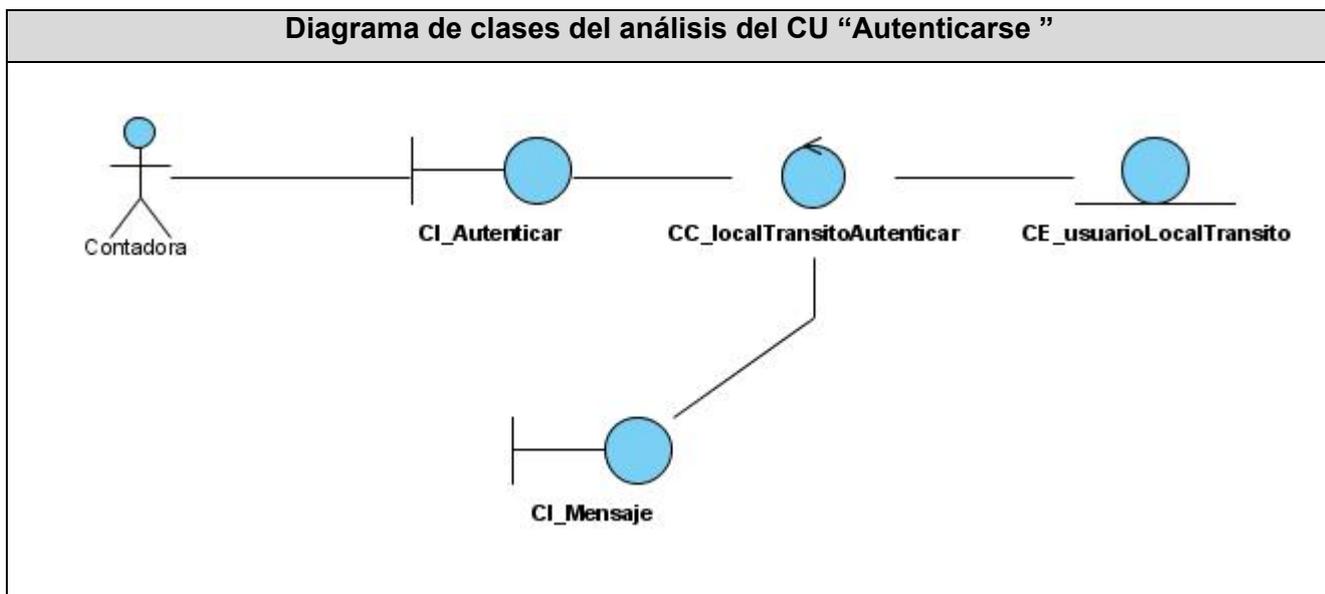


Figura 3: Diagrama de clases del análisis del caso de uso "Autenticarse".

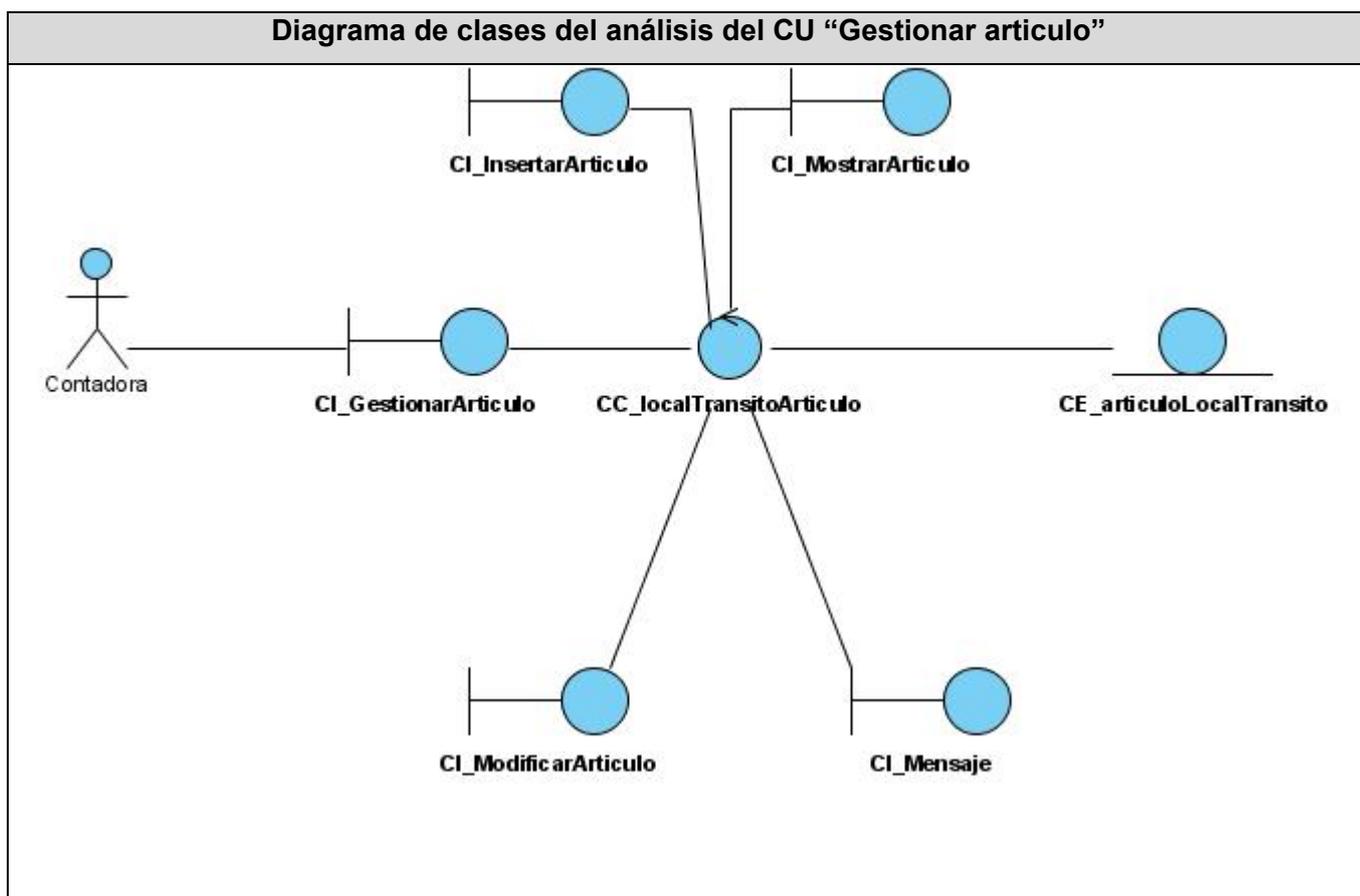


Figura 4: Diagrama de clases del análisis del caso de uso "Gestionar articulo"

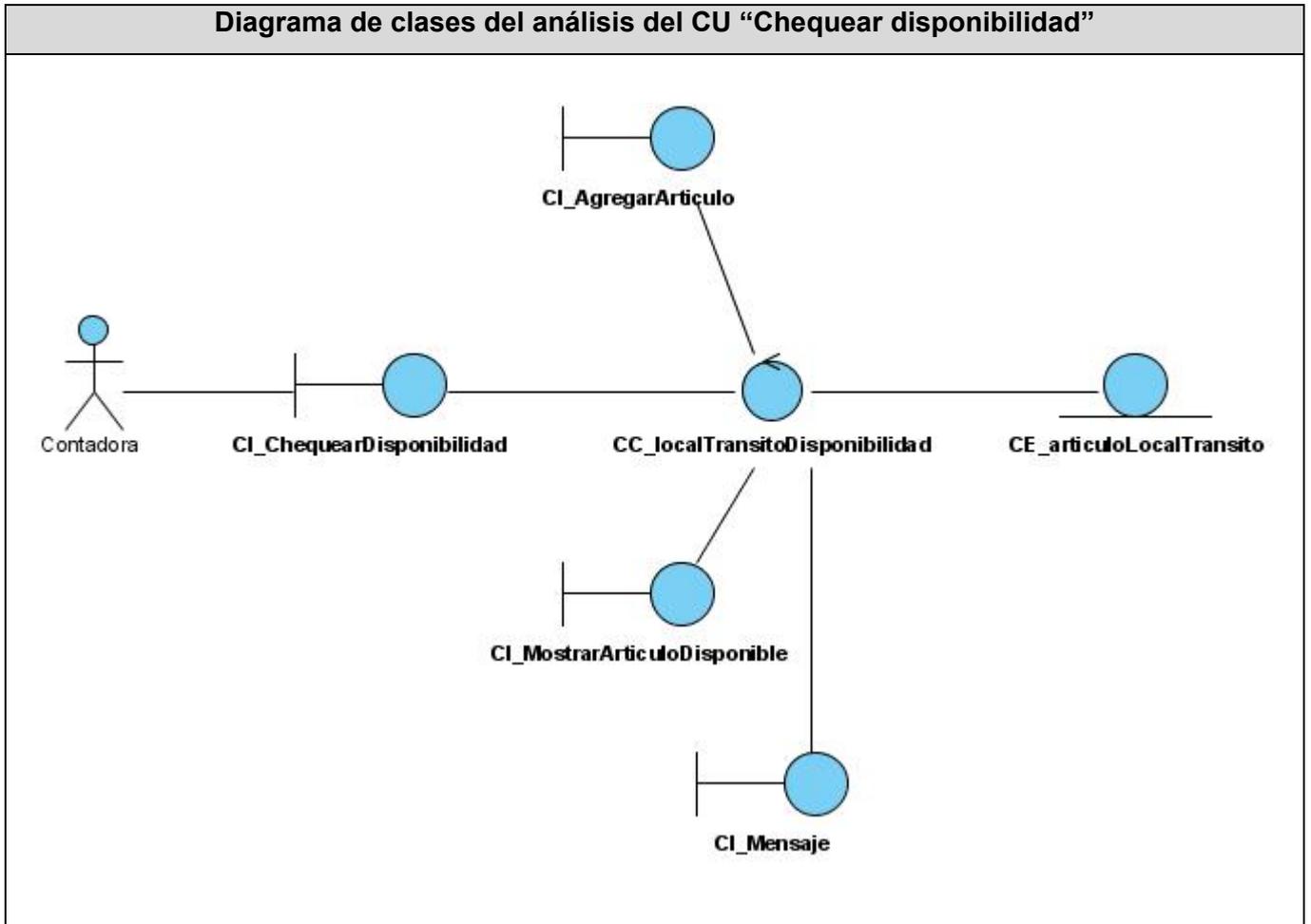


Figura 5: Diagrama de clases del análisis del caso de uso "Chequear disponibilidad".

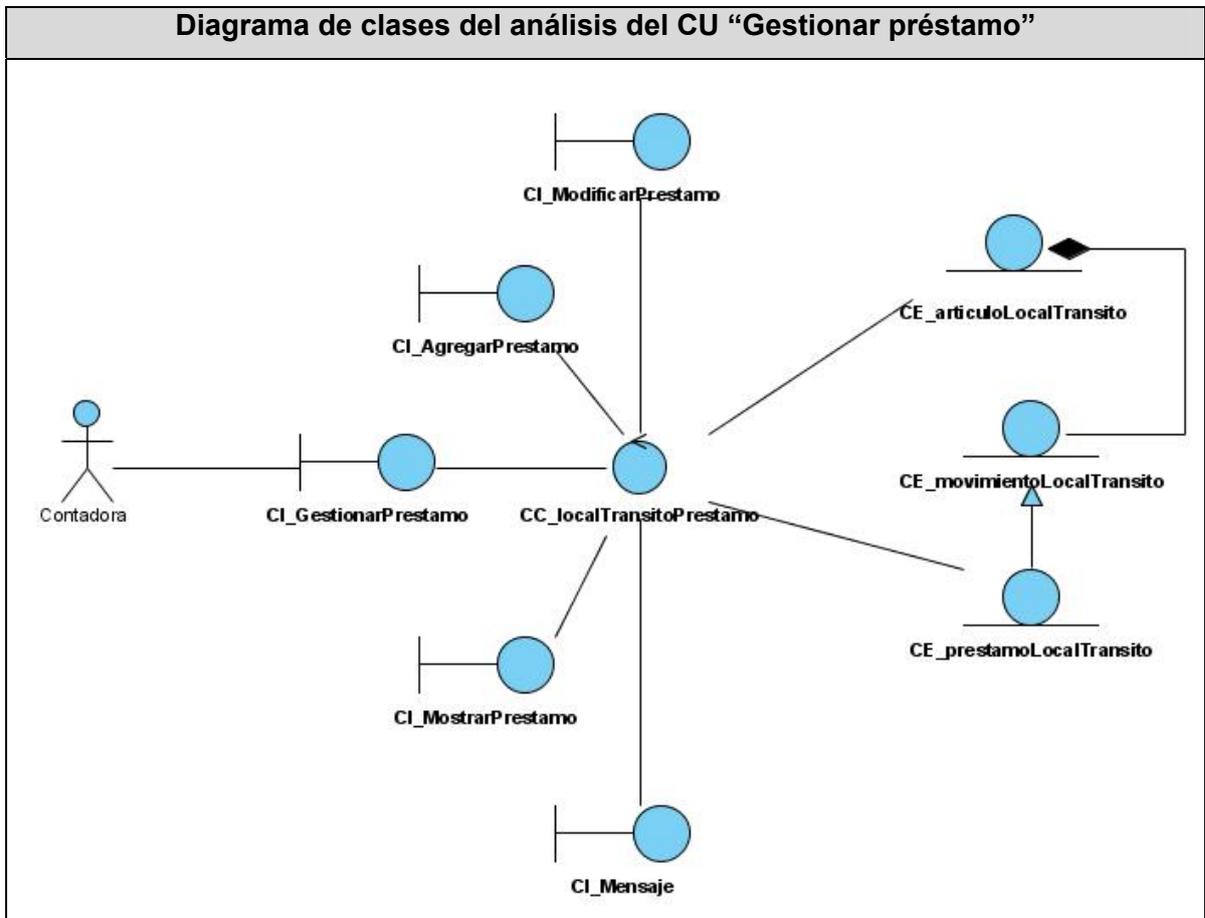


Figura 6: Diagrama de clases del análisis del caso de uso "Gestionar préstamo"

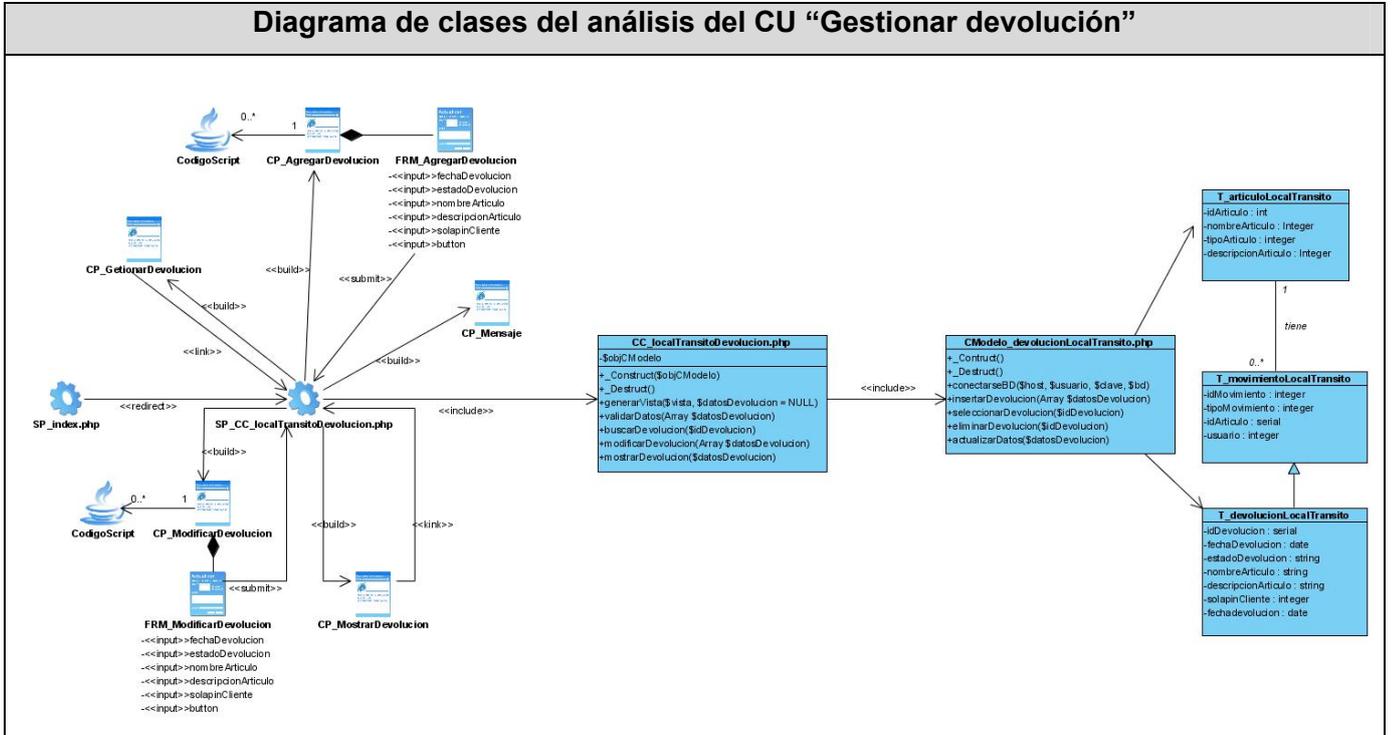


Figura 7: Diagrama de clases del análisis del caso de uso "Gestionar devolución".

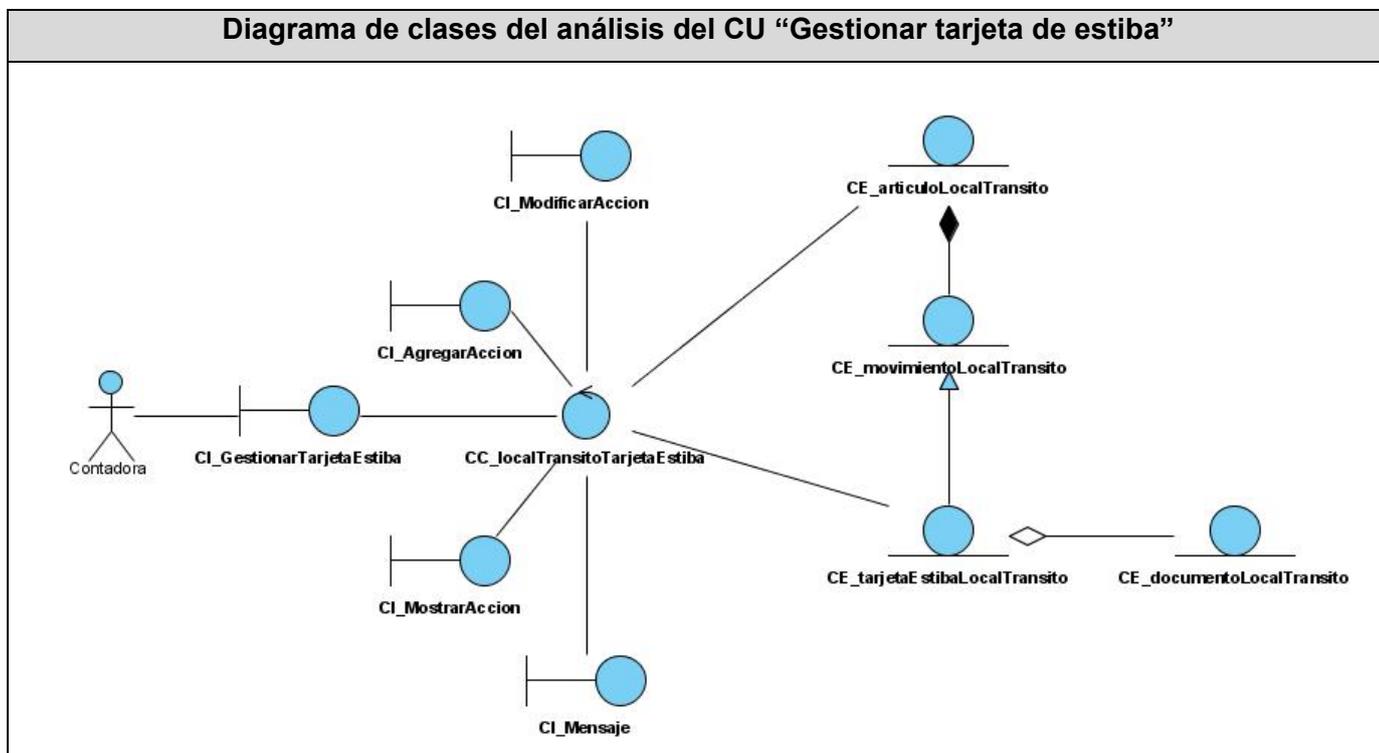


Figura 8: Diagrama de clases del análisis del caso de uso "Gestionar tarjeta de estiba".

3.3. Diseño.

En el diseño se produce un modelo o representación técnica del software que se va a desarrollar, en esta etapa se define una subdivisión del sistema por funciones y la forma de comunicación para su interacción. Durante esta fase se utiliza la arquitectura cliente-servidor.

Ventajas de la arquitectura cliente-servidor:

- ✓ El servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes.
- ✓ Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre para otra conexión.

La decisión de soportar esta arquitectura surgió por la necesidad de una mayor organización del código, una funcionalidad más proporcionada además de contar con una mayor fortaleza y menos riesgo de que intrusos corrompan el sistema. Proporciona también la construcción de un sistema distribuido.

3.3.1. Diagramas de interacción.

Un Diagrama de Interacción muestra una interacción, que consta de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

3.3.1.2. Diagramas de secuencia.

Un diagrama *de* secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes. Gráficamente, un diagrama de secuencia es una tabla que representa objetos, dispuestos a lo largo del eje X, y mensajes, ordenados según se suceden en el tiempo, a lo largo del eje Y. (Ver anexo 5).

3.3.2. Diagramas de Clases del Diseño por cada caso de uso.

Un Diagrama de Clases de Diseño muestra la especificación para las clases software de una aplicación. Muestra definiciones de entidades software más que conceptos del mundo real.

Cuando se trata de una aplicación web UML ofrece una serie de estereotipos para representar las partes que contiene dicha aplicación. Por ejemplo en esta aplicación:

- Las páginas de código HTML, son las páginas clientes, que están compuestas por formularios y contienen código Java Script.
- Las páginas de código PHP, son las páginas servidoras.
- Las clases entidades son las tablas de la base de datos.

En los distintos diagramas de clases del diseño se representa el patrón de arquitectura Modelo Vista Controlador (MVC).

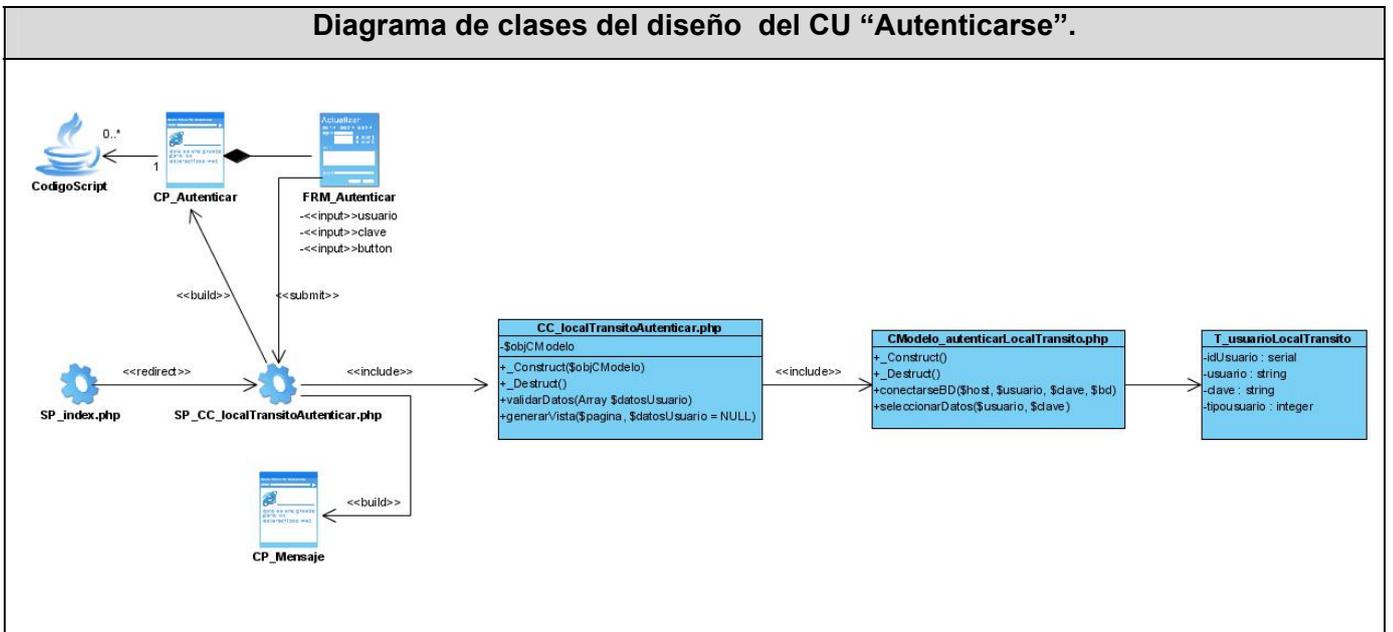


Figura 9: Diagrama de clases del diseño del caso de uso "Autenticarse".

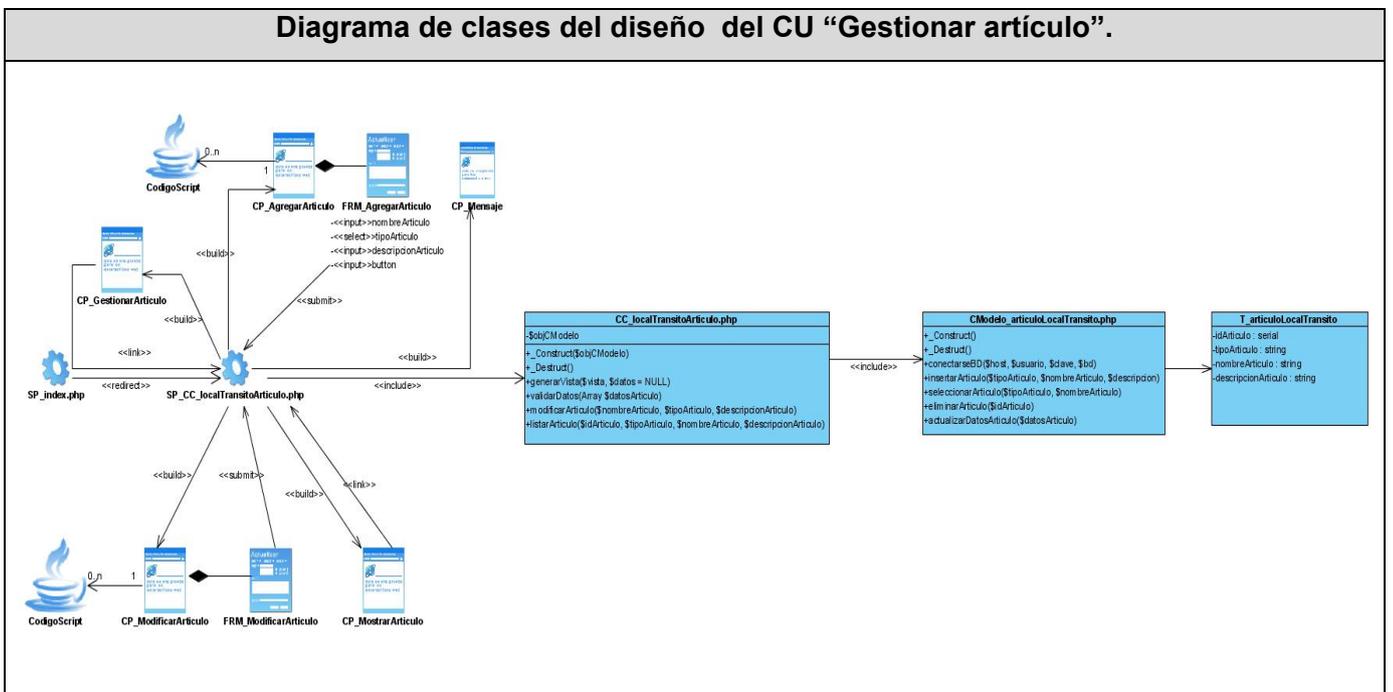


Figura 10: Diagrama de clases del diseño del caso de uso "Gestionar artículo".

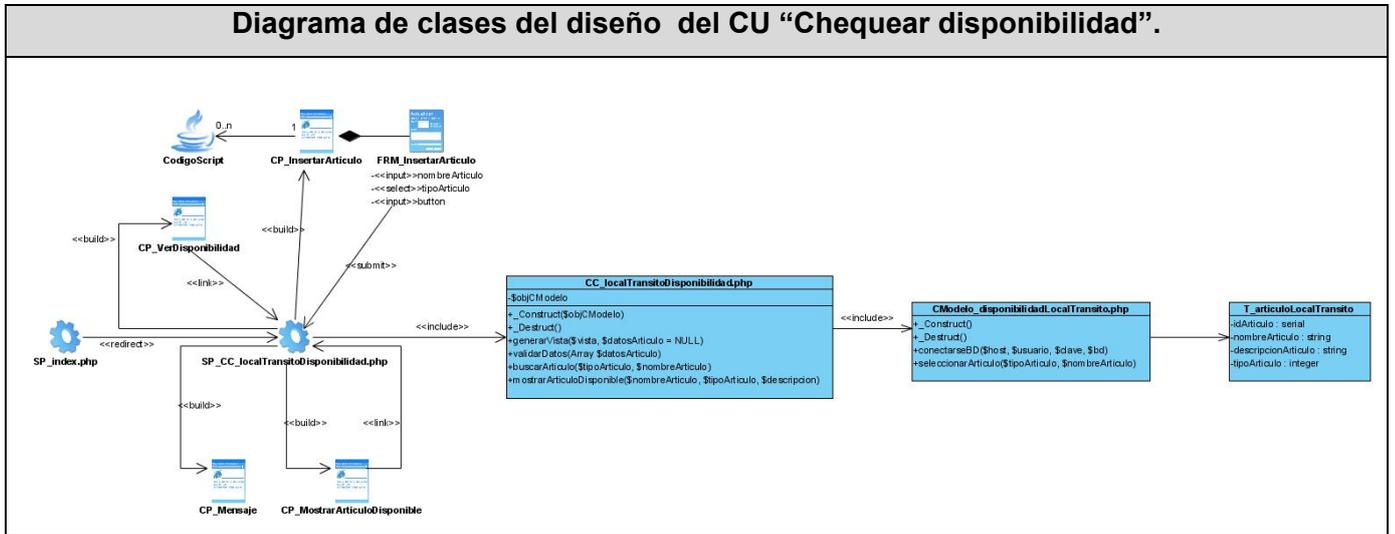


Figura 11: Diagrama de clases del diseño del caso de uso "Chequear disponibilidad".

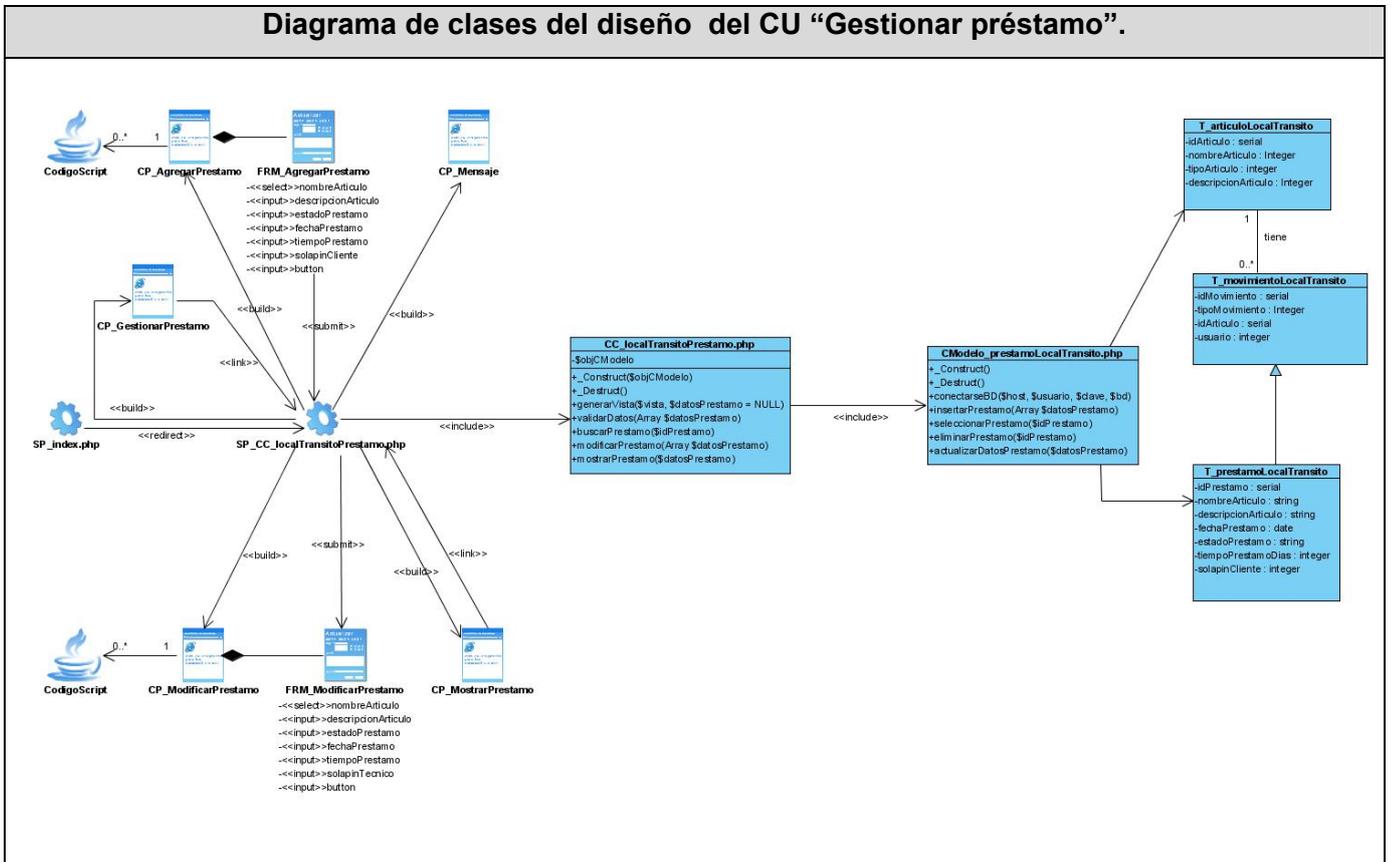


Figura 12: Diagrama de clases del diseño del caso de uso "Gestionar préstamo".

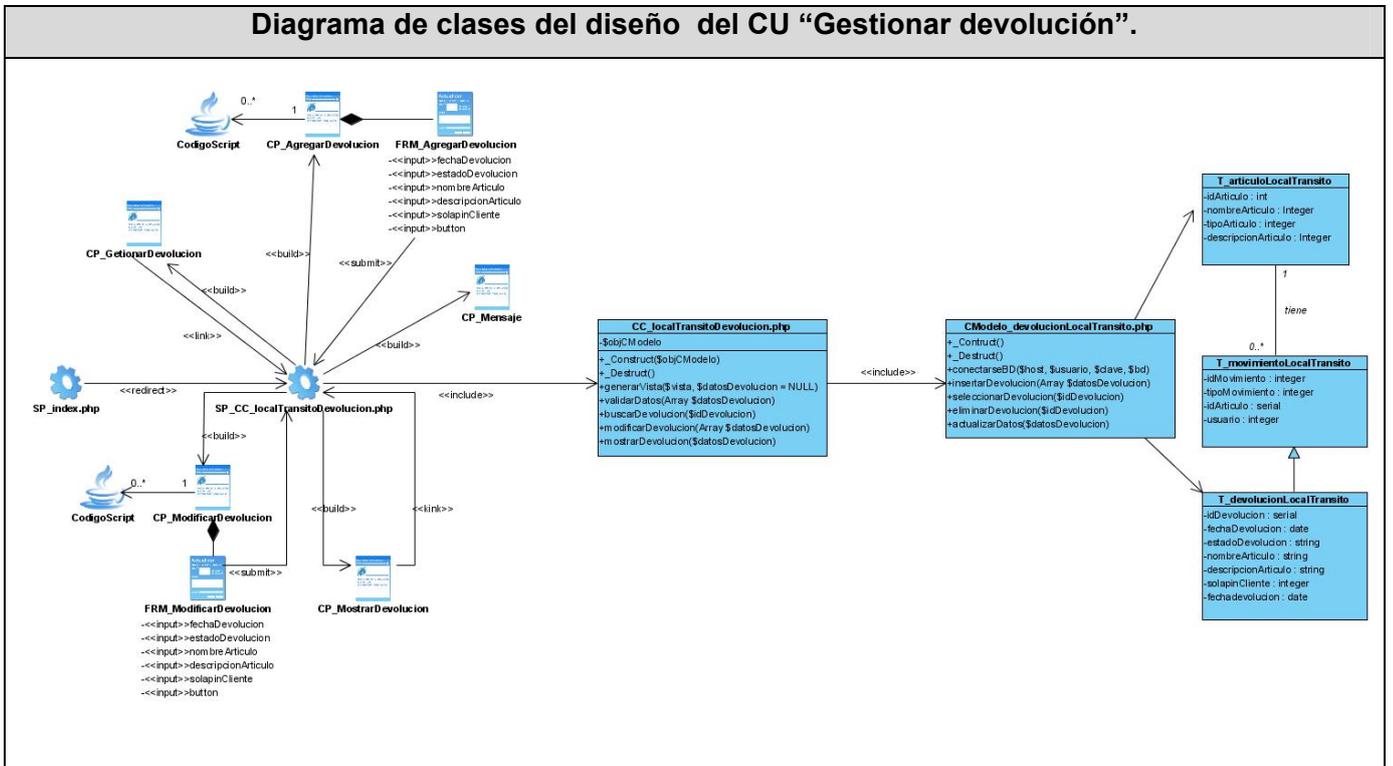


Figura 13: Diagrama de clases del diseño del caso de uso "Gestionar devolución"

Diagrama de clases del diseño del CU "Gestionar tarjeta de estiba".

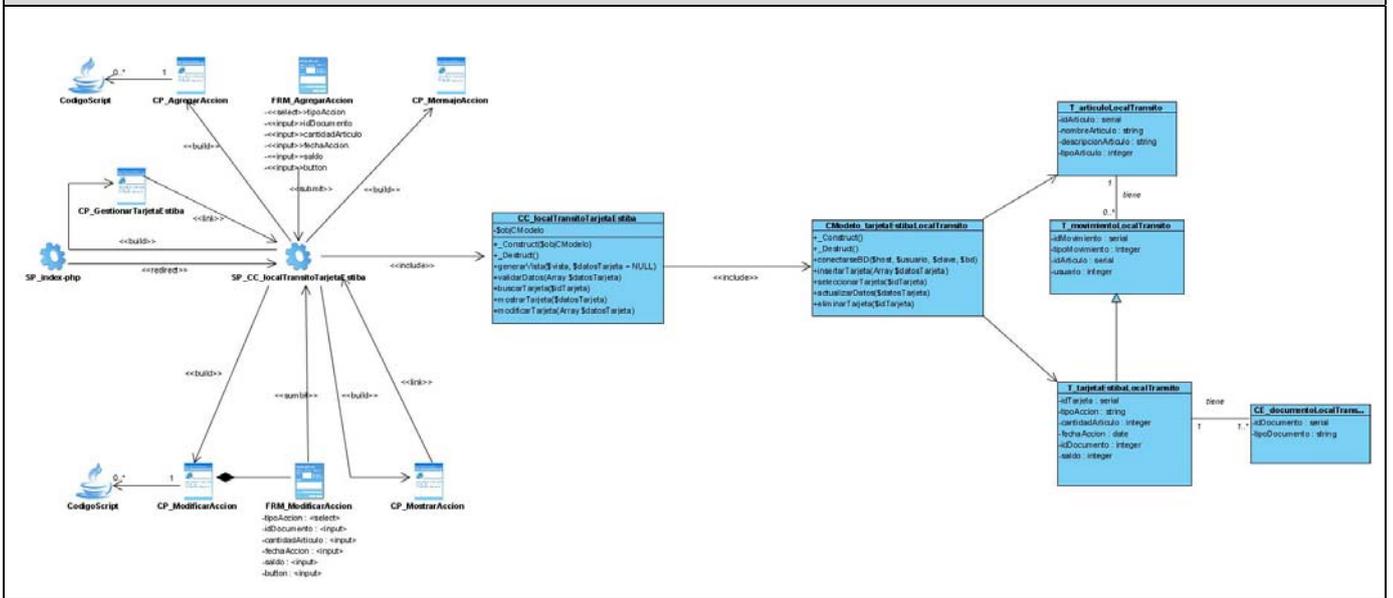


Figura 14: Diagrama de clases del diseño del caso de uso "Gestionar tarjeta de estiba".

3.4. Diseño de la base de datos.

3.4.1. Modelo lógico de datos (Diagrama de clases persistentes)

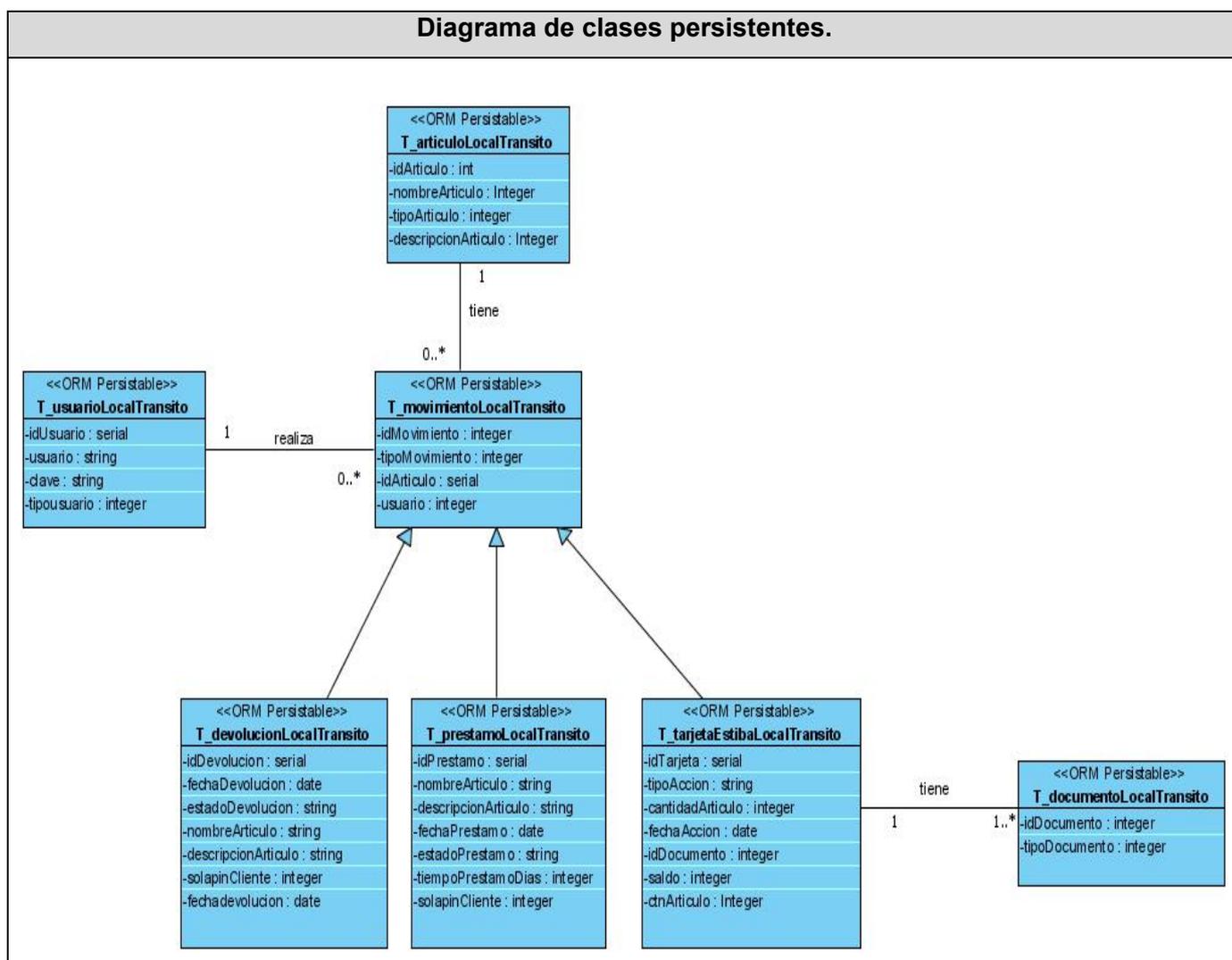


Figura 15: Diagrama de clases persistentes.

3.4.2. Modelo físico de datos (Modelo de datos).

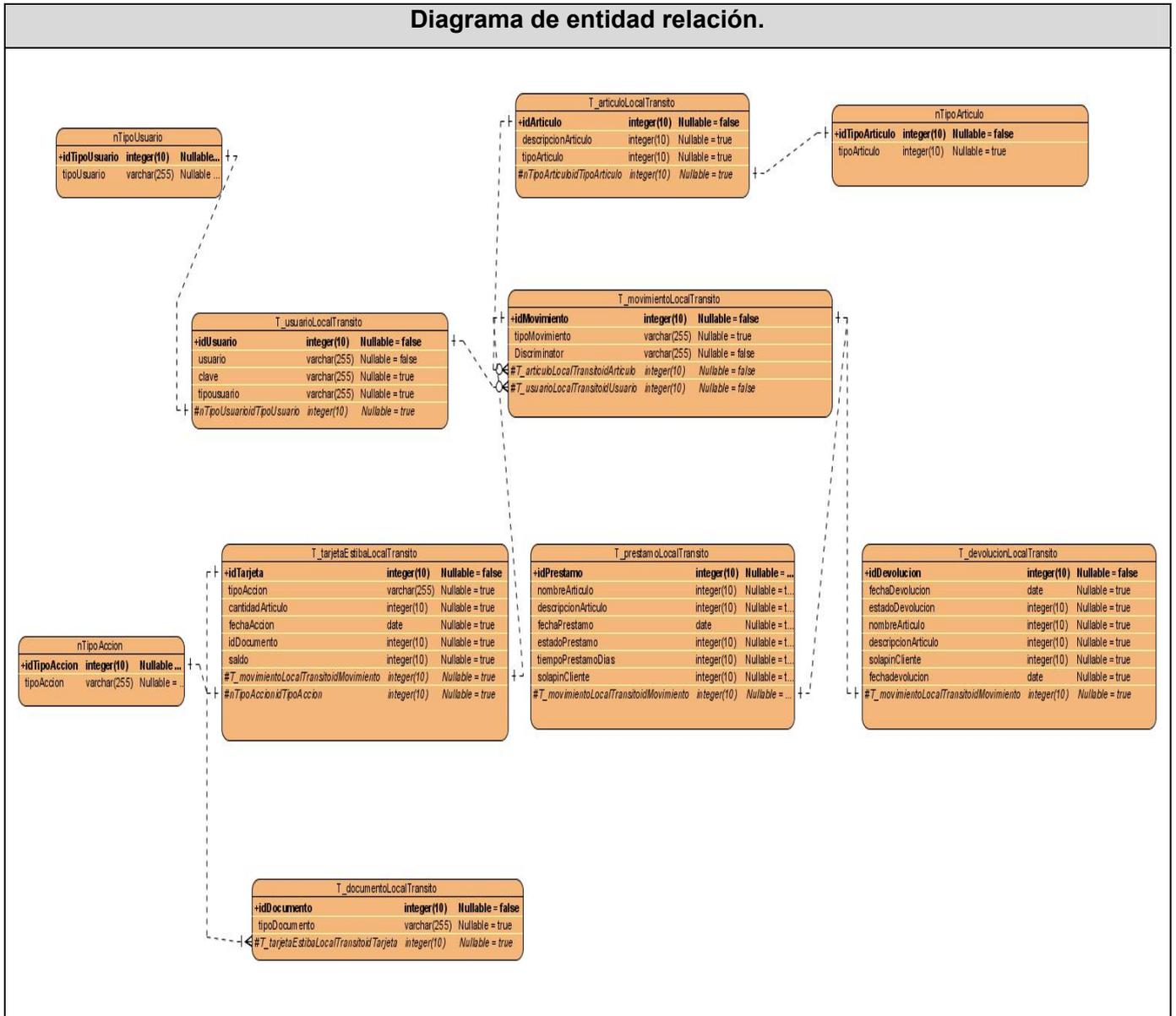


Figura 16: Diagrama entidad-relación.

3.4.3. Diagrama de despliegue.

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).

En general un nodo será una unidad de computación de algún tipo, desde un sensor a un mainframe. Las instancias de componentes software pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (ya que un componente puede tener más de una interfaz).

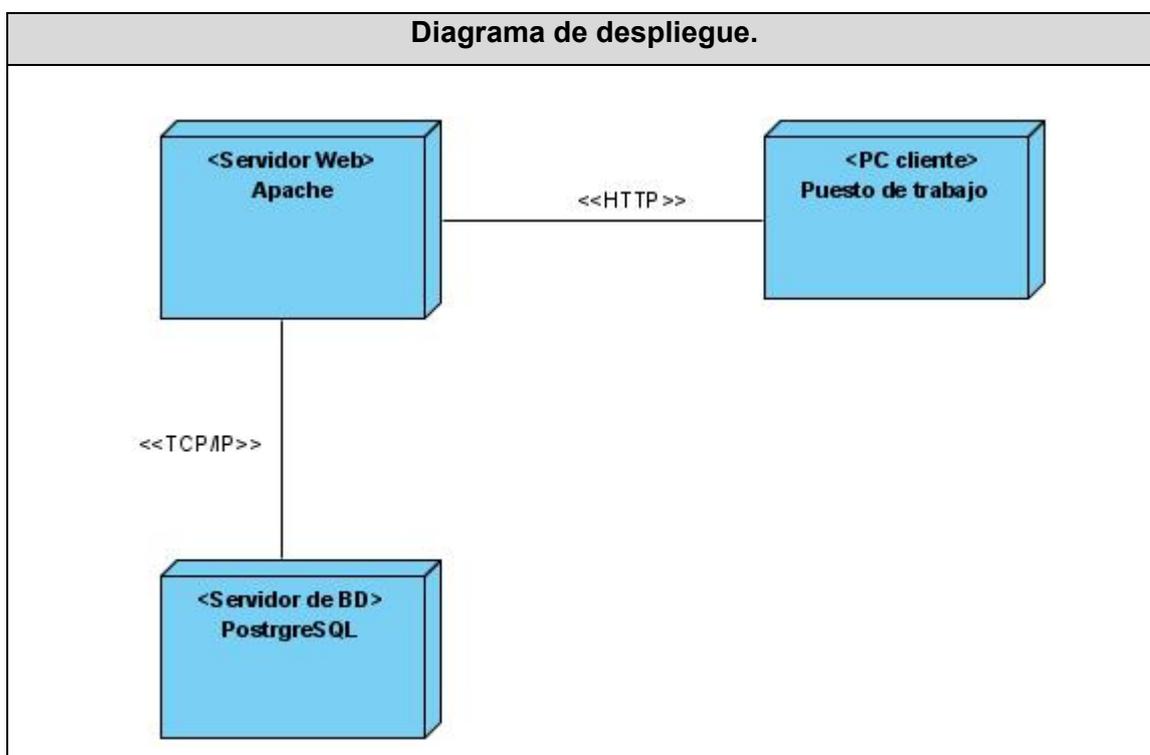


Figura 17: Diagrama de despliegue.

3.5. Conclusiones

Existen varios modelos o métodos para la realización del análisis y diseño de un sistema, lo primero del trabajo fue revisar que es el Análisis y el diseño y posteriormente presentar varios modelos que podemos utilizar para la realización y elaboración de un proceso y trabajo exhaustivo y dar solución o respuesta al problema que se ha generado.

En este capítulo se muestran los resultados obtenidos en el flujo de trabajo Análisis y Diseño, así como una breve explicación de los artefactos que se construyeron durante el desarrollo en la búsqueda de una solución final óptima y que satisfaga las necesidades del usuario.

CAPÍTULO 4: IMPLEMENTACIÓN

4.1. Introducción

En este capítulo se mostrará como los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, ficheros de código fuente, ejecutables etc. A través de los diagramas de despliegue y de componentes, artefactos que se construyen en este flujo de trabajo.

4.2. Diagrama de componentes.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. Se representa como un grafo de componentes software unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten.

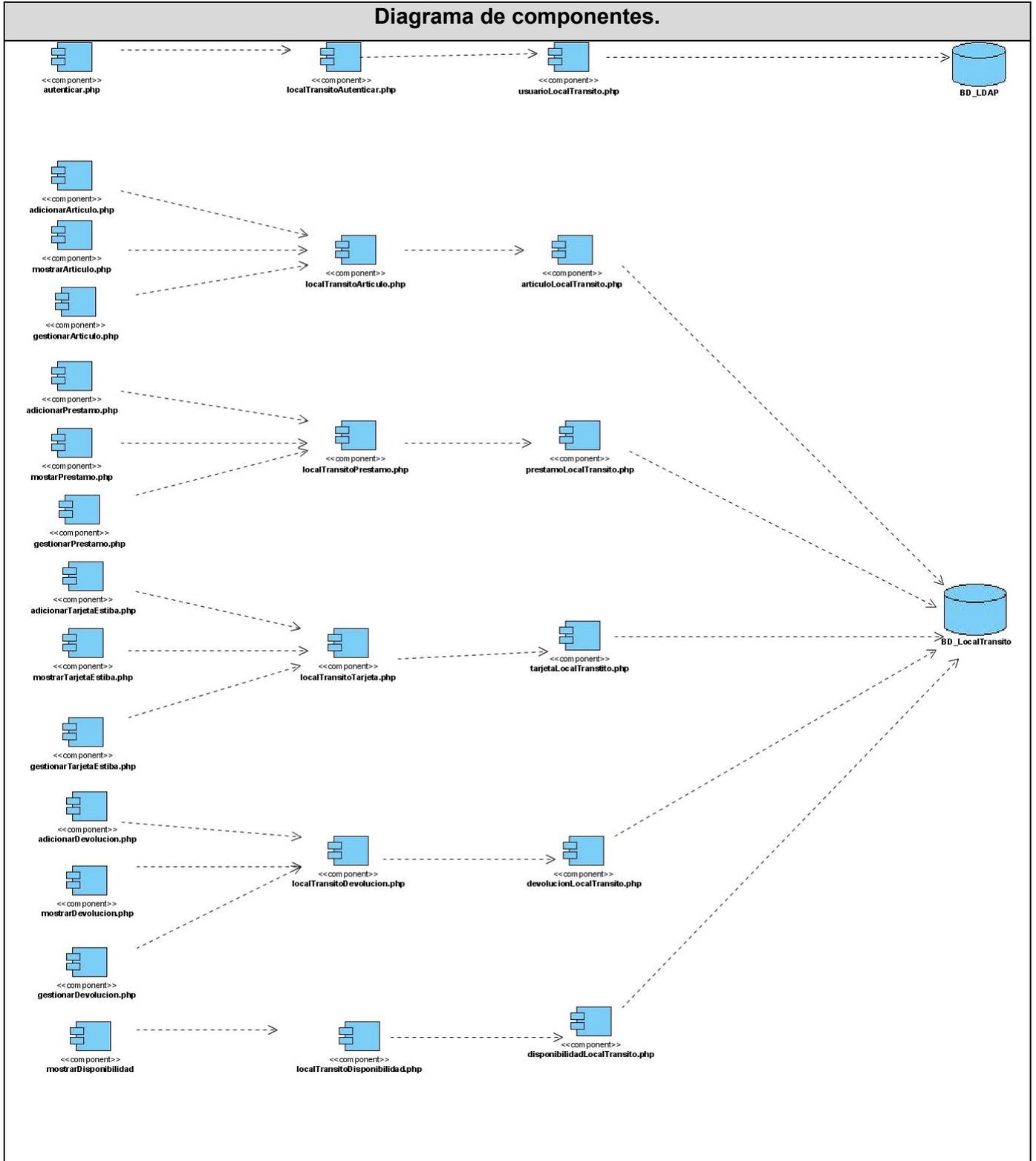


Figura 18: Diagrama de componentes.

4.3 Conclusiones

En este capítulo se ha podido contemplar la dependencia entre los diferentes componentes con la realización del diagrama de componentes.

CAPÍTULO 5: ESTUDIO DE LA FACTIBILIDAD.

5.1. Introducción.

Una de las principales tareas para asumir la realización de un proyecto es el análisis de factibilidad y viabilidad del proyecto, debido a la importancia de visualizar los beneficios que reportará. Este estudio previo permite estimar el tiempo de desarrollo en meses, el costo y la cantidad de personas que se necesitan para desarrollar el proyecto.

En este capítulo se describe la estimación de costos del sistema propuesto y sus beneficios.

5.2. Planificación basada en puntos de Casos de Uso.

La estimación mediante el análisis de Puntos de Casos de Uso se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

A continuación, se detallan los pasos a seguir para la aplicación de éste método.

Paso 1: Calculo de puntos de Casos de Uso sin ajustar.

Consiste en el cálculo de los Puntos de Casos de Uso sin ajustar. Este valor, se calcula a partir de la siguiente ecuación:

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

donde,

- **UUCP:** Puntos de Casos de Uso sin ajustar.
- **UAW:** Factor de Peso de los Actores sin ajustar.
- **UUCW:** Factor de Peso de los Casos de Uso sin ajustar.

Calculando el factor de peso de los actores sin ajustar (UAW).

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos.

Tipo de actor	Descripción	Factor de peso	Actor	Total
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación	1	0	0
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto	2	0	0
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica	3	1	3

Tabla. 4: Factor de peso de los actores sin ajustar.

$$UAW = \sum \text{Factor de peso} * \text{Cantidad de actores}$$

$$UAW = 3.$$

Calculando factor de Peso de los Casos de Uso sin ajustar (UUCW).

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia. Los criterios se muestran en la siguiente tabla:

Tipo de CU	Descripción	Factor de peso	Cant de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	3	15
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	4	40
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0

Tabla. 5: Factor de peso de los casos de uso sin ajustar.

$$UUCW = \sum \text{cantCU} * \text{peso}$$

$$UUCW = 55$$

$$UUCP = UAW + UUCW$$

$$UUCP = 3 + 55$$

$$UUCP = 58$$

Paso 2: Cálculo de Puntos de Casos de Uso ajustados.

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

$$UCP = UUCP * TCF * EF$$

donde,

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica.

EF: Factor de ambiente

Calculando Factor de complejidad técnica (TCF).

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores:

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	1	1
T3	Eficiencia del usuario final	1	1	1
T4	Funcionamiento Interno complejo	1	1	1
T5	El código debe ser reutilizable	1	3	3
T6	Facilidad de instalación	0.5	2	1
T7	Facilidad de uso	0.5	4	2
T8	Portabilidad	2	2	4
T9	Facilidad de cambio	1	2	2
T10	Concurrencia	1	4	4
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	5	5
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	1

Tabla. 6: Factor de complejidad técnica.

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

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

$$TCF = 0.6 + 0.01 * 28$$

$$TCF = 0.88$$

Calculando el Factor Ambiente (EF).

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente.

- Para los factores E1 al E4, un valor asignado de 0 significa sin experiencia, 3 experiencia media y 5 amplia experiencia (experto).
- Para el factor E5, 0 significa sin motivación para el proyecto, 3 motivación media y 5 alta motivación.
- Para el factor E6, 0 significa requerimientos extremadamente inestables, 3 estabilidad media y 5 requerimientos estables sin posibilidad de cambios.
- Para el factor E7, 0 significa que no hay personal part-time (es decir todos son full-time), 3 significa mitad y mitad, y 5 significa que todo el personal es part-time (nadie es full-time).

- Para el factor E8, 0 significa que el lenguaje de programación es fácil de usar, 3 medio y 5 que el lenguaje es extremadamente difícil.

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	5	7.5
E2	Experiencia en la aplicación	0.5	0	0
E3	Experiencia en la orientación a objetos.	1	4	4
E4	Capacidad del analista líder.	0.5	5	2.5
E5	Motivación.	1	5	5
E6	Estabilidad de requerimientos	2	3	6
E7	Personal Part-Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	3	-3

Tabla. 7: Factor ambiente.

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i * \text{Valor asignado}_i)$$

$$EF = 1.4 - 0.03 * 22$$

$$EF = 0.74$$

Finalmente, los Puntos de Casos de Uso ajustados resultan:

$$UCP = 58 * 0.88 * 0.74 = 37.77$$

Paso 3: Estimación del esfuerzo

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CF$$

donde,

E: esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: factor de conversión

$$E = 37.77 * 20$$

$$E = 755.4 \text{ Horas-Hombre.}$$

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	140
Diseño	20	280
Implementación	40	560
Pruebas	15	210
Sobrecarga (otras actividades)	15	210
Total	100	1400

Tabla. 8: Estimación del esfuerzo.

Esfuerzo Total horas-hombre (E_{THH}) \rightarrow 1400

Esfuerzo Total mes-hombre (E_{TMH}) \rightarrow Este valor se obtiene de la división del E_{THH} entre el total de horas que se trabajan al mes. Como se trabajan un total de 8 horas diarias al mes esto representa un valor de 240 horas, entonces:

$$\begin{aligned}
 E_{TMH} &= E_{THH} / 240 \\
 &= 1400 / 240 \\
 &= 5,83
 \end{aligned}$$

Salario Promedio (S_M) \rightarrow 87,5

Cantidad de Hombres (C) \rightarrow 2

Costo Hombre-Mes (C_{HM}) \rightarrow Este valor es el resultado de la multiplicación del S_M por C :

$$C_{HM} = 175$$

Costo Total (T) \rightarrow El costo total se obtiene multiplicando E_{TMH} por el C_{HM} , entonces:

$$\begin{aligned}
 T &= E_{TMH} * C_{HM} \\
 &= 5,83 * 175 \\
 &= 1020,25
 \end{aligned}$$

5.3. Beneficios tangibles e intangibles.

El sistema para la automatización de la entrada y salida de piezas partes y equipos en de tránsito, no se ha realizado con fines comerciales puesto que su objetivo fundamental es:

- Tener un control preciso de la entrada y salida de las piezas, partes y equipos en el local de tránsito.

Beneficios Intangibles.

- Disminución del tiempo y esfuerzo que se invierte en las tareas que se realizan, de forma manual.
- Mejor calidad del servicio ofertado a los trabajadores.
- Fácil obtención de la información, dinámica de reportes en el momento y lugar que desee el trabajador.

5.4. Análisis de costos y beneficios.

La publicación de la fachada traerá consigo considerables beneficios a la hora brindar a los trabajadores un servicio de información referente a los recursos humanos, pues reduce considerablemente el tiempo que se emplea realizando de forma tradicional, además de que permitirá generar reportes con una mayor rapidez y efectividad.

5.5. Conclusiones.

En este capítulo se calcularon y analizaron los beneficios que puede traer la realización de la aplicación, así como el costo que puede implicar el desarrollo de la misma.

CONCLUSIONES

Una vez realizado el sistema para la automatización del movimiento de equipos, partes y piezas en el local de tránsito se ha cumplido con el objetivo general y los objetivos específicos que se trazaron antes de concebirse la implementación, flujo de trabajo que contribuyó a erradicar el engorroso trabajo manual y a la agilización de los procesos que se desarrollan en el local, materializándose así todas las metas propuestas para la entrega del software.

RECOMENDACIONES

Se recomienda seguir de cerca el desarrollo del local de tránsito para que en caso de que sus funcionalidades se incrementen la aplicación mejore y cumpla con los requerimientos necesarios garantizando una correcta funcionalidad de la misma.

BIBLIOGRAFÍA

1. Wikipedia, la enciclopedia libre. [En línea] 7 de julio de 2007. [Citado el: 10 de febrero de 2008.] http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web.
2. desarrolloweb.com. [En línea] [Citado el: 16 de febrero de 2008.] <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.
3. Universidad Nacional de Colombia. [En línea] <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060029/lecciones/cap11-2.html>. 3.
4. LinuxCentro.net. [En línea] <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
5. Wikipedia, la enciclopedia libre. [En línea] http://es.wikipedia.org/w/index.php?title=Servidor_HTTP_Apache&oldid=5792245.
6. [En línea] www.alegsa.com.ar/Dic/javascript.php.
7. Wikipedia, la enciclopedia libre. [En línea] <http://es.wikipedia.org/wiki/PostgreSQL>.
8. Facultad de Ciencias, UNAM. [En línea] <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>.
9. [En línea] http://www.ulpgc.es/otros/tutoriales/java/Apendice/arq_mvc.html.
10. Tutorial de Java. [En línea] http://www.cica.es/formacion/JavaTut/Apendice/arq_mvc.html.
11. Wikipedia, la enciclopedia libre. [En línea] <http://es.wikipedia.org/wiki/RUP>.
12. Free Download Manager. *Sitio de descargas de software*. [En línea] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Pi_ataforma_de_Java_14715_p/.

ANEXOS

Anexo I: Descripción de los casos de uso del negocio en formato expandido.

Caso de uso del negocio: Solicitar préstamo	
Actores del negocio: Cliente (inicia)	
Trabajador del negocio: Contadora del local.	
Propósito: Solicitar al personal autorizado el préstamo de un artículo.	
Resumen: El caso de uso comienza cuando el cliente llega al local de tránsito solicitando préstamo de un artículo. La contadora el local chequea la disponibilidad del artículo que se solicita y si este se encuentra disponible se recogen los datos del solicitante y entrega la pieza, en caso contrario el cliente se retira.	
Curso normal de eventos:	
Acción del actor	Respuesta del proceso de negocio
1- Solicitar préstamo de un artículo determinado.	2- Chequear disponibilidad de artículo.
	3- Solicitar al cliente su solapín.
4- Entregar solapín.	5- Registrar datos del técnico y del artículo en el libro de préstamos.
	6- Entregar artículo.
7- El cliente se retira.	
Flujo alternativo de los eventos:	
Acción del actor:	Respuesta del proceso de negocio:
	2.1 Informar al cliente que el artículo no está disponible.
2.2- El cliente se retira.	

Tabla 9: Caso de uso del negocio “Solicitar préstamo”.

Caso de uso del negocio: Efectuar devolución de préstamo.	
Actores del negocio: Cliente (inicia)	
Trabajador del negocio: Contadora del local.	
Propósito: Devolver al personal autorizado un artículo que ha sido prestado con anterioridad.	
Resumen: El caso de uso comienza cuando el cliente llega a local de tránsito a realizar una devolución de préstamo. La contadora el local chequea que el artículo que trae el cliente sea el que está registrado en el libro y lo recoge, en caso contrario informa al cliente que ese no es el artículo que se le prestó y el cliente se retira.	
Curso normal de eventos:	
Acción del actor	Respuesta del proceso de negocio
1- Realizar devolución de préstamo.	2- Verificar datos de artículo.
	3- Asentar devolución de artículo.
4- El cliente se retira.	
Flujo alternativo de los eventos:	
Acción del actor:	Respuesta del proceso de negocio:
	2.1 Informar al cliente que el artículo que va a devolver no coincide con el que esta registrado.
2.2- El cliente se retira.	

Tabla 10: Caso de uso del negocio:"Efectuar devolución de préstamo".

Caso de uso del negocio: Solicitar artículo	
Actores del negocio: Cliente (inicia)	
Trabajador del negocio: Contadora del local.	
Propósito: Solicitar al personal autorizado un artículo.	
Resumen: El cliente llega a local de tránsito con una orden de servicio de un artículo descompuesto y dicho artículo para reemplazarlo por uno en buen estado, en el caso de que existan artículos disponibles en buen estado, la contadora solicita al cliente la orden de servicio, el artículo roto y entrega el que se encuentra en buen estado.	
Curso normal de eventos:	
Acción del actor	Respuesta del proceso de negocio
1- Solicitar un artículo determinado.	2- Chequear disponibilidad de artículo.
	3- Solicitar orden de servicio y artículo descompuesto.
4- Entregar orden de servicio y artículo descompuesto.	5- Entrega artículo en buen estado.
6- El cliente se retira.	
Flujo alterno de los eventos:	
Acción del actor:	Respuesta del proceso de negocio:
	2.1- Informar al cliente que el artículo no está disponible.
2.2- El cliente se retira.	

Tabla 11: Caso de uso del negocio "Solicitar artículo".

Anexo II: Diagramas de actividad.

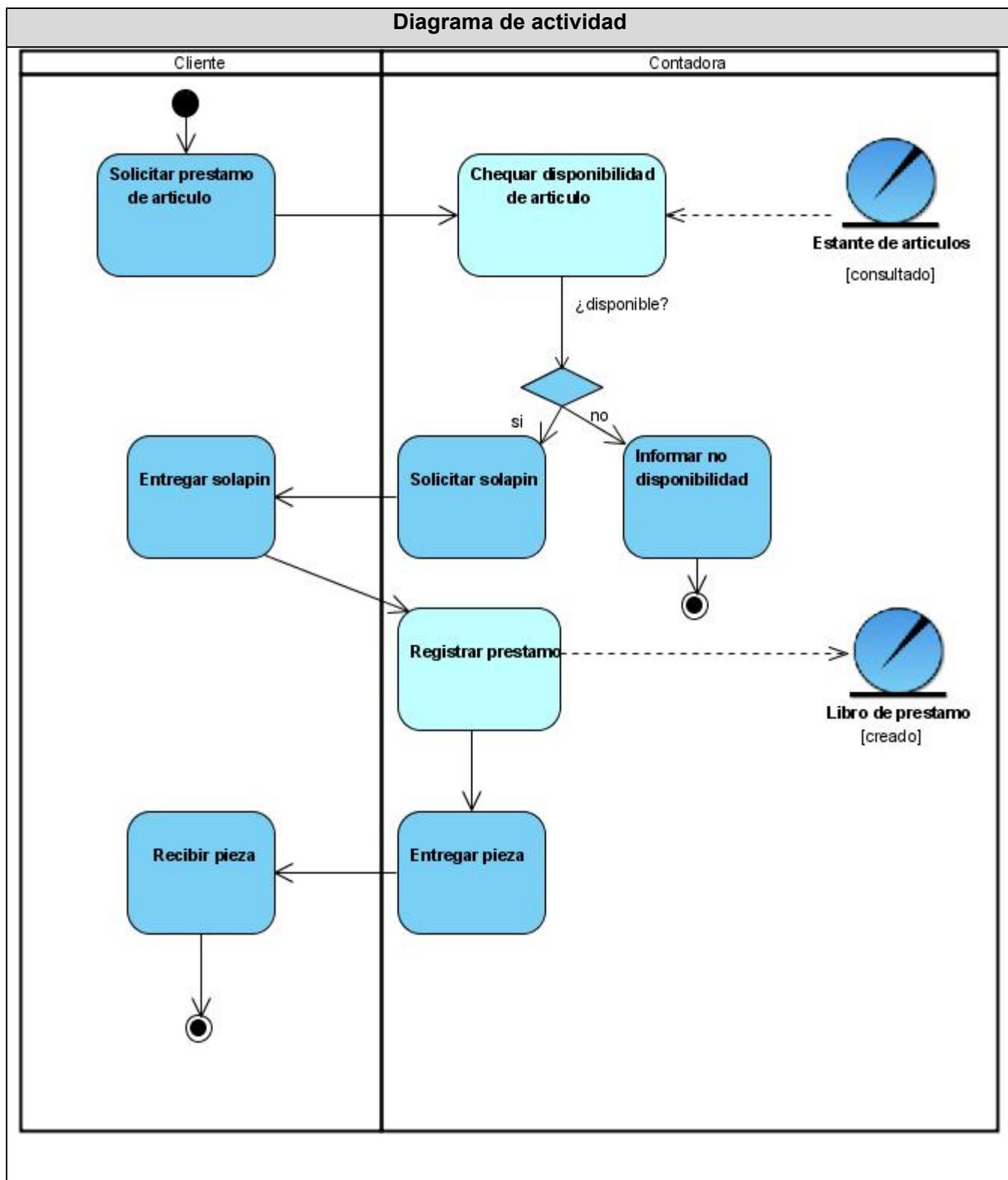


Figura 19: Caso de uso del negocio "Solicitar préstamo".

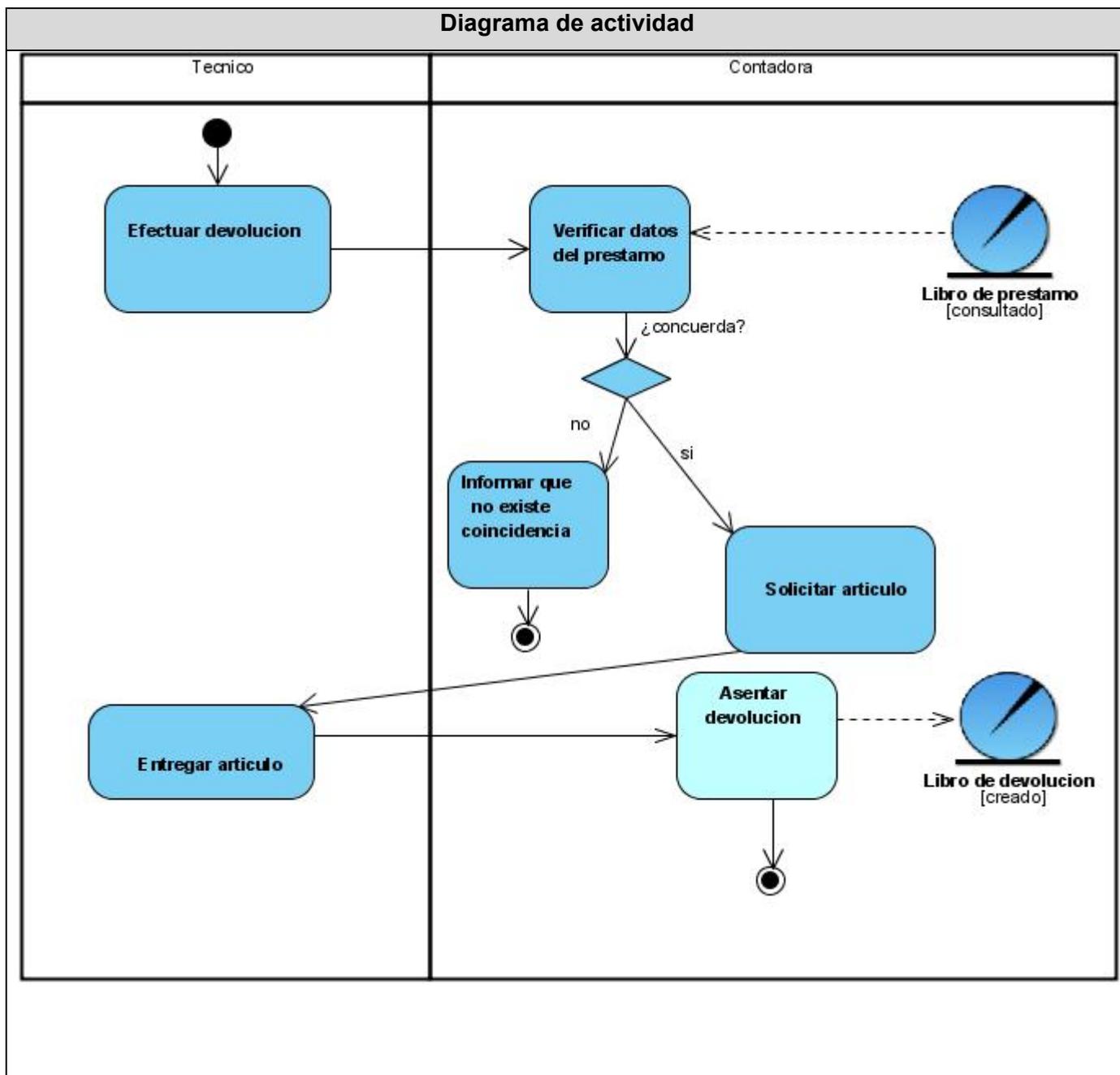


Figura 20: Caso de uso del negocio "Efectuar devolución".

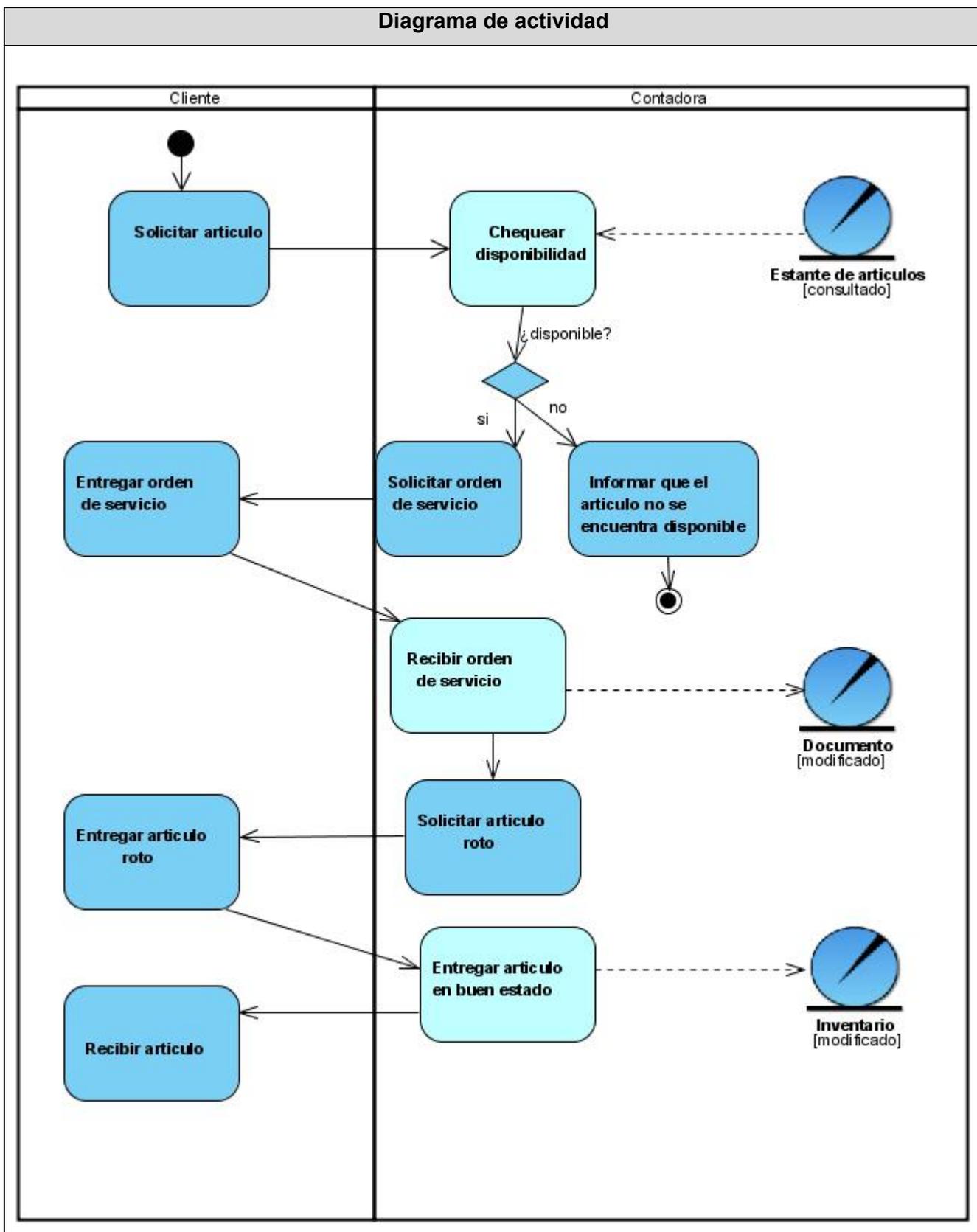


Figura 21: Caso de uso del negocio "Solicitar artículo".
Anexo III: Modelo de objetos

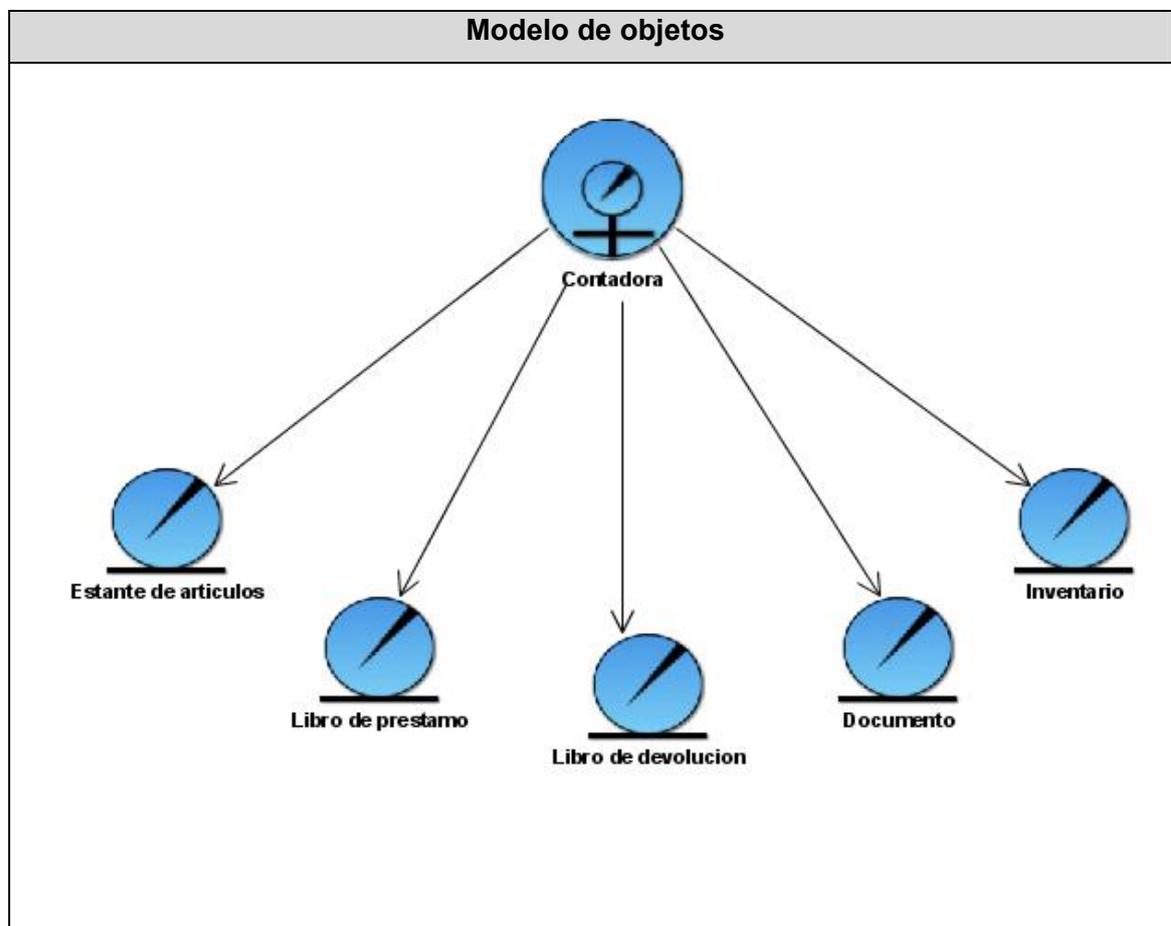


Figura 22: Modelo de objetos.

Anexo IV: Descripción de los casos de uso del sistema.

Caso de uso	Autenticarse	
Propósito	Verificar que el usuario registrado tiene los permisos necesarios para acceder al sistema.	
Actores	Contadora.	
Resumen	El caso de uso se inicia cuando la contadora del local desea acceder a la aplicación, pero para ello debe autenticarse previamente. El sistema debe ser capaz de validar los datos e introducidos y permitir o denegar al usuario el acceso al sistema, mostrando en ambos casos un mensaje de fallo o éxito en la acción.	
Referencias	R ₁ .	
Precondiciones	Insertar usuario y contraseña.	
Poscondiciones	Se habilitan privilegios para el usuario.	
Curso normal de eventos		
Acción del actor	Respuesta del sistema	
1- Usuario decide entrar al sistema.	2- Sistema muestra la interfaz de autenticación.	
3- Usuario introduce su nombre de usuario y contraseña en las casillas correspondientes y elige la opción "Autenticar".	4- Valida los datos introducidos	
	5- El sistema muestra un mensaje indicando el éxito de la acción y se habilitan los privilegios.	
Curso alternativo de los eventos		
Línea 5a: Campos vacíos.		
5a.1- El usuario deja campos vacíos.	5a.2- El sistema muestra un mensaje confirmando la existencia de campos vacíos.	
5a.3- Retornar al paso 3.		
Línea 5b: Datos incorrectos		
5b.1- El usuario entra datos incorrectos.	5b.2- El sistema muestra un mensaje informando que hay errores en la autenticación en el dominio.	
Prioridad: Crítico.		

Tabla 12: Caso de uso del sistema "Autenticarse".

Caso de uso	Gestionar artículo	
Propósito	Insertar, mostrar, modificar datos de artículo y/o eliminar.	
Actores	Contadora.	
Resumen	El caso de uso se inicia cuando la contadora del local accede a la aplicación para insertar, modificar, mostrar o eliminar datos de un artículo.	
Referencias	R ₂ .	
Precondiciones	Se habilitaron los privilegios para el usuario autenticado.	
Poscondiciones	Se actualiza la base de datos del sistema.	
Curso normal de eventos		
Acción del actor	Respuesta del sistema	
1- La contadora del local selecciona la opción "Artículo" en el menú.	2- Sistema muestra la interfaz con una lista de todos los artículos que existen en el sistema, dándole la posibilidad al usuario de realizar diferentes opciones:	
	2.1- Si desea insertar un artículo en el sistema, ir a la sección "Agregar artículo" 2.2- Si va a eliminar un artículo ir a la sección "Eliminar artículo". 2.3- Si va a modificar un artículo del sistema, ir a la sección "Modificar artículo"	
Sección "Agregar artículo"		
3- La contadora del local elige la opción "Agregar artículo".	4- El sistema muestra la interfaz para agregar el artículo.	
5- La contadora introduce los datos del artículo y selecciona la opción "Enviar".	6- El sistema valida los datos introducidos.	
	7- El sistema almacena los datos del artículo en la base de datos y muestra un mensaje indicando el éxito de la operación.	
8- La contadora elige la opción "Continuar".	9- Retorna al paso 2 y finaliza el caso de uso.	

Curso alternativo de los eventos	
Línea 5a: Operación cancelada.	
5a.1- La contadora elige la opción "Cancelar".	5a.2- El sistema cancela la operación y retorna al paso 2.
Línea 5b: Campos vacíos.	
5b.1- La contadora deja campos vacíos.	5b.2- El sistema muestra un mensaje debajo de cada campo vacío.
5b.3- Retornar al paso 5.	
Línea 5c: Datos incorrectos.	
5c.1- La contadora introduce datos incorrectos.	5c.2- El sistema no almacena la información y muestra un mensaje confirmando que hay datos incorrectos y le indica cuáles son.
5c.3- La contadora elige la opción "Continuar"	5c.4- Retornar al paso 4
Línea 5c.3. Ignorar opción "Continuar".	
5c.3.1. La contadora ignora la opción "Continuar".	5c.3.2. Retornar al paso 4.
Línea 8a. Ignorar opción "Continuar".	
8a.1- La contadora ignora la opción "Continuar".	8a.2- Retornar al paso 2.
Sección "Eliminar artículo"	
3. La contadora elige la opción "Eliminar".	4. El sistema muestra una confirmación para eliminar artículo.
5. La contadora elige la opción "Aceptar".	6. El sistema elimina el artículo de su base de datos y muestra un mensaje confirmando el éxito de la operación.
7. La contadora selecciona la opción "Continuar".	8- Retornar al paso 2, culminando así el caso de uso.
Curso alternativo de los eventos	
Línea 5a. Operación cancelada.	
5a.1. La contadora elige la opción "Cancelar".	5a.2. El sistema cancela la operación.
Línea 7a. Ignorar opción "Continuar".	
7a.1. La contadora ignora la opción "Continuar".	7a.2. Retornar al paso 2.
Sección "Modificar artículo"	
5- La contadora del local selecciona el artículo que desea modificar.	6- El sistema muestra en la interfaz los datos del artículo que se pueden modificar.

7- La contadora del local modifica los datos y elige la opción "Aceptar".	8- El sistema muestra una interfaz que ofrece la posibilidad de modificación.
	9- El sistema verifica que los datos introducidos sean correctos y que no existan campos vacios.
	10- Se muestra un mensaje indicando el éxito de la operación y finaliza el caso de uso.
Curso alternativo de los eventos	
Línea 7a:Datos incorrectos	
7a.1- La contadora introduce datos incorrectos.	7a.2- El sistema no almacena la información en su base de datos y muestra un mensaje confirmando que hay datos que no son correctos y le indica cuales son.
7a.3- La contadora elige la opción "Continuar".	7a.4- Retornar al paso 8.
Línea 7b:Datos incorrectos	
7b.1- La contadora deja campos vacios.	7b.2- El sistema muestra un mensaje debajo de cada campo vacio.
7b.3- Retornar al paso 7	
Prioridad: Critico	

Tabla 13: Caso de uso del sistema "Gestionar artículo".

Caso de uso	Gestionar tarjeta de estiba	
Propósito	Registrar acciones de entrada y salida de artículos	
Actores	Contadora del local.	
Resumen	El caso de uso se inicia cuando entran o salen artículos del local y se requiere registrar estas acciones y conocer el saldo existente después de cada entrada o salida	
Referencias	R ₅ , R ₆ .	
Precondiciones	Se habilitaron los privilegios para el usuario autenticado.	
Poscondiciones	Se actualiza la base de datos.	
Curso normal de eventos		
Acción del actor	Respuesta del sistema	
1-La contadora del local selecciona la opción "Tarjeta de estiba" del menú.	2- El sistema muestra una interfaz con la lista de todas las tarjetas de estiba y diferentes opciones que le permiten a la contadora:	
	2.1- Agregar acción, ver sección "Agregar tarjeta". 2.2- Modificar tarjeta, ver sección "Modificar tarjeta". 2.3- Eliminar tarjeta, ver sección "Eliminar tarjeta".	
Curso alternativo de los eventos		
Línea 2a. No hay tarjetas registradas.		
2a.1. No ha registrado a ninguna tarjeta.	2a.2. Retornar al paso 2, pero en vez de mostrar la lista de tarjetas, el sistema muestra un mensaje confirmando que no hay tarjetas registradas.	
Sección "Agregar tarjeta".		
3- La contadora elige la opción "Agregar tarjeta".	4- El sistema muestra la interfaz para agregar la nueva acción y conformar la tarjeta de estiba.	
5- La contadora introduce los datos y presiona la opción "Aceptar"	6- El sistema valida los datos introducidos.	
	7-El sistema almacena la información en su base de datos y muestra un mensaje	

	confirmando el éxito de la operación.
Curso alternativo de los eventos	
Línea 5a: Operación cancelada.	
5a.1- La contadora elige la opción "Cancelar".	5a.2. El sistema cancela la operación. Retornar al paso 2.
Línea 5b: Campos vacíos	
5b.1- La contadora deja campos en blanco.	5b.2- El sistema muestra un mensaje indicando que existen campos vacíos.
5b.3- Retornar al paso 3.	
Línea 5c: Datos incorrectos	
5c.1- La contadora introduce datos incorrectos.	5c.2- El sistema muestra un mensaje indicando que existen datos incorrectos.
Sección "Eliminar tarjeta"	
3- La contadora elige la opción "Eliminar".	4- El sistema muestra una confirmación para eliminar tarjeta.
5- La contadora elige la opción "Aceptar".	6- El sistema elimina los datos de la tarjeta de su base de datos y muestra un mensaje confirmando el éxito de la operación.
7- La contadora selecciona la opción "Continuar".	8- Retornar al paso 2, culminando así el caso de uso.
Curso alternativo de los eventos	
Línea 5a. Operación cancelada.	
5a.1. La contadora elige la opción "Cancelar".	5a.2. El sistema cancela la operación.
Línea 7a. Ignorar opción "Continuar".	
7a.1. La contadora ignora la opción "Continuar".	7a.2. Retornar al paso 2.
Sección "Modificar tarjeta"	
5- La contadora del local selecciona la tarjeta de estiba que desea modificar.	6- El sistema muestra en la interfaz los datos de la tarjeta que se pueden modificar.
7- La contadora del local modifica los datos y elige la opción "Aceptar".	8- El sistema muestra una interfaz que ofrece la posibilidad de modificación.
	9- El sistema verifica que los datos introducidos sean correctos y que no existan campos vacíos.

	10- Se muestra un mensaje indicando el éxito de la operación y finaliza el caso de uso.
Curso alternativo de los eventos	
Línea 7a:Datos incorrectos	
7a.1- La contadora introduce datos incorrectos.	7a.2- El sistema no almacena la información en su base de datos y muestra un mensaje confirmando que hay datos que no son correctos y le indica cuales son.
7a.3- La contadora elige la opción "Continuar".	7a.4- Retornar al paso 8.
Línea 7b:Datos incorrectos	
7b.1- La contadora deja campos vacios.	7b.2- El sistema muestra un mensaje debajo de cada campo vacio.
7b.3- Retornar al paso 7	
Prioridad: Critico	

Tabla 14: Caso de uso del sistema "Gestionar tarjeta de estiba".

Caso de uso	Registrar documento	
Propósito	Insertar un documento.	
Actores	Gestionar tarjeta de estiba.	
Resumen	El caso de uso se inicia cuando la contadora del local accede a la aplicación para insertar una acción (entrada/salida de articulo). Cada acción que se realice en el sistema debe estar respaldada por un documento.	
Referencias	R ₅ , R ₆ .	
Precondiciones	Se habilitaron los privilegios para el usuario autenticado.	
Poscondiciones	Se actualiza la base de datos.	
Curso normal de eventos		
Acción del actor	Respuesta del sistema	
1-La contadora del local selecciona la opción "Tarjeta de estiba" del menú.	2- El sistema muestra una interfaz con la lista de todas las tarjetas de estiba y diferentes opciones que le permiten al administrador:	
	2.1- Agregar acción, ver sección "Agregar tarjeta".	
3- La contadora selecciona cualquiera de la opción "Agregar tarjeta" y realiza la operación correspondiente.	4- El sistema verifica que los datos introducidos sean correctos y que el campo donde se enuncia el articulo que respalda la acción de entrada/salida no este vacio o contenga errores.	
	5- El sistema almacena los datos introducidos y actualiza la tabla de la base de datos donde se guardan los documentos finalizando el caso de uso.	
Curso alternativo de los eventos		
Línea 2a. No hay tarjetas registradas.		
2a.1. No ha registrado a ninguna tarjeta.	2a.2. Retornar al paso 2, pero en vez de	

	mostrar la lista de tarjetas, el sistema muestra un mensaje confirmando que no hay tarjetas registradas.
Curso alternativo de los eventos	
Línea 5a:Datos incorrectos	
5a1- La contadora introduce los datos incorrectos del documento.	5a2- Muestra un mensaje indicando que hay errores en los datos, retornando al paso 3.

Tabla 15: Caso de uso “Registrar documento”

Caso de uso	Gestionar Préstamo	
Propósito	Insertar, eliminar y mostrar un préstamo de una pieza dada una fecha determinada o nombre de la misma.	
Actores	Contadora del local.	
Resumen	El caso de uso se inicia cuando la contadora del local accede a la aplicación para insertar o eliminar solicitudes de préstamo o para mostrar, dada una fecha de préstamo, de devolución o el nombre de una pieza, los préstamos realizados.	
Referencias	R ₄ .	
Precondiciones	Se habilitaron los privilegios para el usuario autenticado.	
Poscondiciones	Se actualiza la base de datos.	
Curso normal de eventos		
Acción del actor	Respuesta del sistema	
1-La contadora del local selecciona la opción "Préstamo" del menú.	2- El sistema muestra una interfaz con la lista de todas las préstamos y diferentes opciones que le permiten al administrador:	
	2.1- Agregar préstamo, ver sección "Agregar préstamo". 2.2- Modificar préstamo, ver sección "Modificar préstamo". 2.3- Eliminar préstamo, ver sección "Eliminar préstamo".	
Curso alternativo de los eventos		
Línea 2a. No hay préstamos registrados.		
2a.1. No ha registrado a ningún préstamo.	2a.2. Retornar al paso 2, pero en vez de mostrar la lista de préstamos, el sistema muestra un mensaje confirmando que no hay préstamos registrados.	
Sección "Agregar préstamo".		

3- La contadora elige la opción "Agregar préstamo".	4- El sistema muestra la interfaz para agregar el préstamo.
5- La contadora introduce los datos y presiona la opción "Aceptar"	6- El sistema valida los datos introducidos.
	7-El sistema almacena la información en su base de datos y muestra un mensaje confirmando el éxito de la operación.
Curso alternativo de los eventos	
Línea 5a: Operación cancelada.	
5a.1- La contadora elige la opción "Cancelar".	5a.2. El sistema cancela la operación. Retornar al paso 2.
Línea 5b: Campos vacíos	
5b.1- La contadora deja campos en blanco.	5b.2- El sistema muestra un mensaje indicando que existen campos vacíos.
5b.3- Retornar al paso 3.	
Línea 5c: Datos incorrectos	
5c.1- La contadora introduce datos incorrectos.	5c.2- El sistema muestra un mensaje indicando que existen datos incorrectos.
Sección "Eliminar préstamo"	
3. La contadora elige la opción "Eliminar".	4. El sistema muestra una confirmación para eliminar el préstamo.
5. La contadora elige la opción "Aceptar".	6. El sistema elimina el préstamo de su base de datos y muestra un mensaje confirmando el éxito de la operación.
7. La contadora selecciona la opción "Continuar".	8- Retornar al paso 2, culminando así el caso de uso.
Curso alternativo de los eventos	
Línea 5a. Operación cancelada.	
5a.1. La contadora elige la opción "Cancelar".	5a.2. El sistema cancela la operación.
Línea 7a. Ignorar opción "Continuar".	
7a.1. La contadora ignora la opción "Continuar".	7a.2. Retornar al paso 2.
Sección "Modificar préstamo."	

5- La contadora del local selecciona el préstamo. que desea modificar.	6- El sistema muestra en la interfaz los datos del préstamo que se pueden modificar.
7- La contadora del local modifica los datos y elige la opción "Aceptar".	8- El sistema muestra una interfaz que ofrece la posibilidad de modificación.
	9- El sistema verifica que los datos introducidos sean correctos y que no existan campos vacios.
	10- Se muestra un mensaje indicando el éxito de la operación y finaliza el caso de uso.
Curso alternativo de los eventos	
Línea 7a:Datos incorrectos	
7a.1- La contadora introduce datos incorrectos.	7a.2- El sistema no almacena la información en su base de datos y muestra un mensaje confirmando que hay datos que no son correctos y le indica cuales son.
7a.3- La contadora elige la opción "Continuar".	7a.4- Retornar al paso 8.
Línea 7b:Datos incorrectos	
7b.1- La contadora deja campos vacios.	7b.2- El sistema muestra un mensaje debajo de cada campo vacio.
7b.3- Retornar al paso 7	
Prioridad: Critico	

Tabla 16: Caso de uso del sistema "Gestionar préstamo".

Caso de uso	Gestionar Devolución	
Propósito	Insertar, eliminar y modificar una devolución de una pieza	
Actores	Contadora del local.	
Resumen	El caso de uso se inicia cuando la contadora del local accede a la aplicación para insertar o eliminar devoluciones.	
Referencias	R ₇ .	
Precondiciones	Se habilitaron los privilegios para el usuario autenticado.	
Poscondiciones	Se actualiza la base de datos.	
Curso normal de eventos		
Acción del actor	Respuesta del sistema	
1-La contadora del local selecciona la opción "Gestionar Devolución" del menú.	2- El sistema muestra una interfaz con la lista de todas las devoluciones y diferentes opciones que le permiten al administrador:	
	2.1- Agregar devolución, ver sección "Agregar devolución". 2.2- Modificar devolución, ver sección "Modificar devolución". 2.3- Eliminar devolución, ver sección "Eliminar devolución".	
Curso alternativo de los eventos		
Línea 2a. No hay préstamos registrados.		
2a.1. No ha registrado a ningún préstamo.	2a.2. Retornar al paso 2, pero en vez de mostrar la lista de devoluciones, el sistema muestra un mensaje confirmando que no hay devoluciones registrados.	
Sección "Agregar devolución".		
3- La contadora elige la opción "Agregar devolución".	4- El sistema muestra la interfaz para agregar la devolución.	
5- La contadora introduce los datos y presiona la opción "Aceptar"	6- El sistema valida los datos introducidos.	
	7-El sistema almacena la información en su	

	base de datos y muestra un mensaje confirmando el éxito de la operación.
Curso alternativo de los eventos	
Línea 5a: Operación cancelada.	
5a.1- La contadora elige la opción "Cancelar".	5a.2. El sistema cancela la operación. Retornar al paso 2.
Línea 5b: Campos vacíos	
5b.1- La contadora deja campos en blanco.	5b.2- El sistema muestra un mensaje indicando que existen campos vacíos.
5b.3- Retornar al paso 3.	
Línea 5c: Datos incorrectos	
5c.1- La contadora introduce datos incorrectos.	5c.2- El sistema muestra un mensaje indicando que existen datos incorrectos.
Sección "Eliminar devolución"	
3. La contadora elige la opción "Eliminar".	4. El sistema muestra una confirmación para eliminar la devolución.
5. La contadora elige la opción "Aceptar".	6. El sistema elimina la devolución de su base de datos y muestra un mensaje confirmando el éxito de la operación.
7. La contadora selecciona la opción "Continuar".	8- Retornar al paso 2, culminando así el caso de uso.
Curso alternativo de los eventos	
Línea 5a. Operación cancelada.	
5a.1. La contadora elige la opción "Cancelar".	5a.2. El sistema cancela la operación.
Línea 7a. Ignorar opción "Continuar".	
7a.1. La contadora ignora la opción "Continuar".	7a.2. Retornar al paso 2.
Sección "Modificar devolución."	
5- La contadora del local selecciona la devolución que desea modificar.	6- El sistema muestra en la interfaz los datos de la devolución que se pueden modificar.
7- La contadora del local modifica los datos y elige la opción "Aceptar".	8- El sistema muestra una interfaz que ofrece la posibilidad de modificación.
	9- El sistema verifica que los datos introducidos sean correctos y que no existan campos vacíos.

	10- Se muestra un mensaje indicando el éxito de la operación y finaliza el caso de uso.
Curso alternativo de los eventos	
Línea 7a:Datos incorrectos	
7a.1- La contadora introduce datos incorrectos.	7a.2- El sistema no almacena la información en su base de datos y muestra un mensaje confirmando que hay datos que no son correctos y le indica cuales son.
7a.3- La contadora elige la opción "Continuar".	7a.4- Retornar al paso 8.
Línea 7b:Datos incorrectos	
7b.1- La contadora deja campos vacios.	7b.2- El sistema muestra un mensaje debajo de cada campo vacio.
7b.3- Retornar al paso 7	
Prioridad: Critico	

Tabla 17: Caso de uso del sistema "Gestionar devolución".

Caso de uso	Chequear disponibilidad	
Propósito	Verificar si un artículo solicitado se encuentra disponible o no.	
Actores	Contadora del local.	
Resumen	El caso de uso se inicia cuando la contadora va a realizar un préstamo o va a responder a una orden de solicitud. Entonces necesita chequear si el artículo solicitado esta disponible o no.	
Referencias	R ₃ .	
Precondiciones	Se habilitaron los privilegios para el usuario autenticado.	
Poscondiciones	Se consulta la base de datos.	
Curso normal de eventos		
Acción del actor	Respuesta del sistema	
1- La contadora del local desea chequear la disponibilidad de un artículo.	2- Sistema muestra la interfaz para que el usuario introduzca datos del artículo que desea chequear su disponibilidad.	
3- La contadora del local introduce el nombre del artículo y presiona la opción "Verificar".	4- Verifica que los datos introducidos sean correctos y que no existan campos vacíos. 5- Busca artículo en el depósito y ejecuta una de las dos acciones: - Si el artículo está disponible, va a la sección "Disponible". - Si el artículo no está disponible va a la sección "No disponible".	
Sección "Disponible"		
	5.1- El sistema muestra el artículo está disponible y finaliza el caso de uso.	
Sección "No disponible"		
	5.1- El sistema muestra un mensaje indicando que el artículo no está disponible y finaliza el caso de uso	
Curso alternativo de los eventos		
	3.1- El sistema muestra un mensaje informando que hay errores en los datos y/o existen campos vacíos.	
Prioridad: Critico		

Tabla 18: Caso de uso del sistema "Chequear disponibilidad".
Anexo V: Diagramas de secuencia.

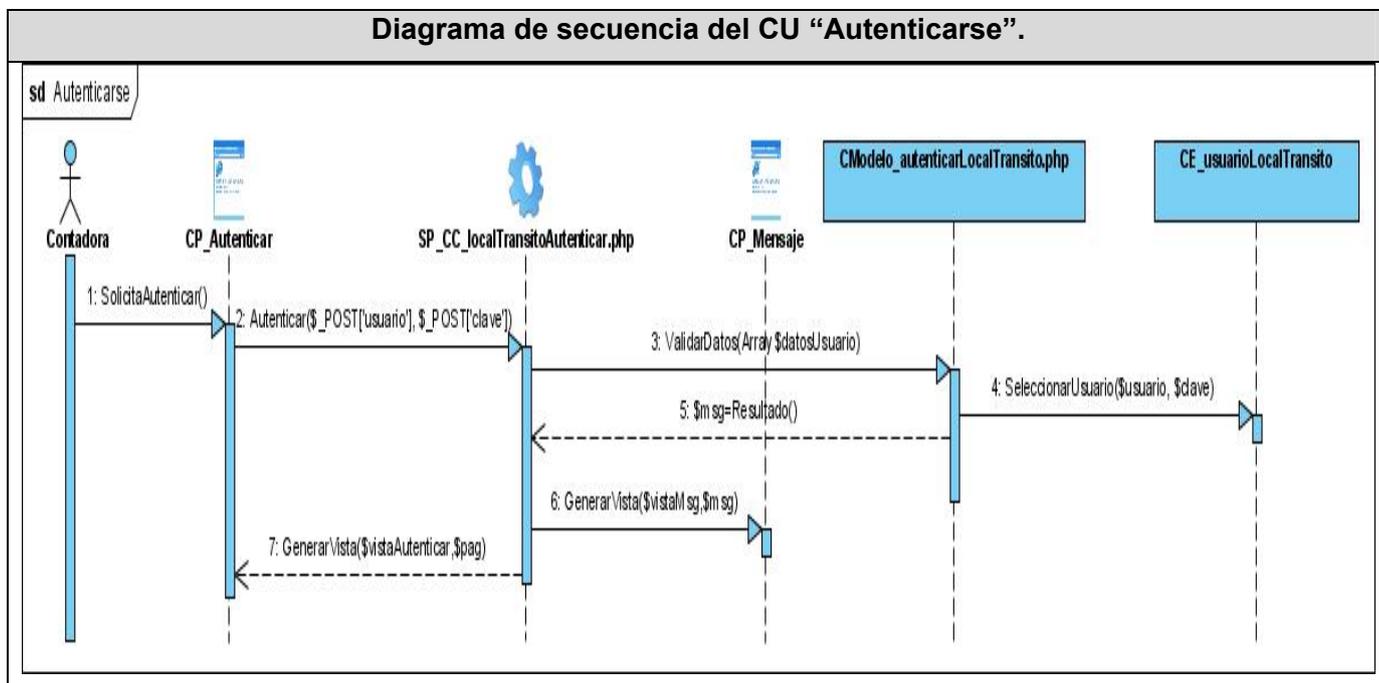


Figura 23: Diagrama de secuencia del caso de uso "Autenticarse".

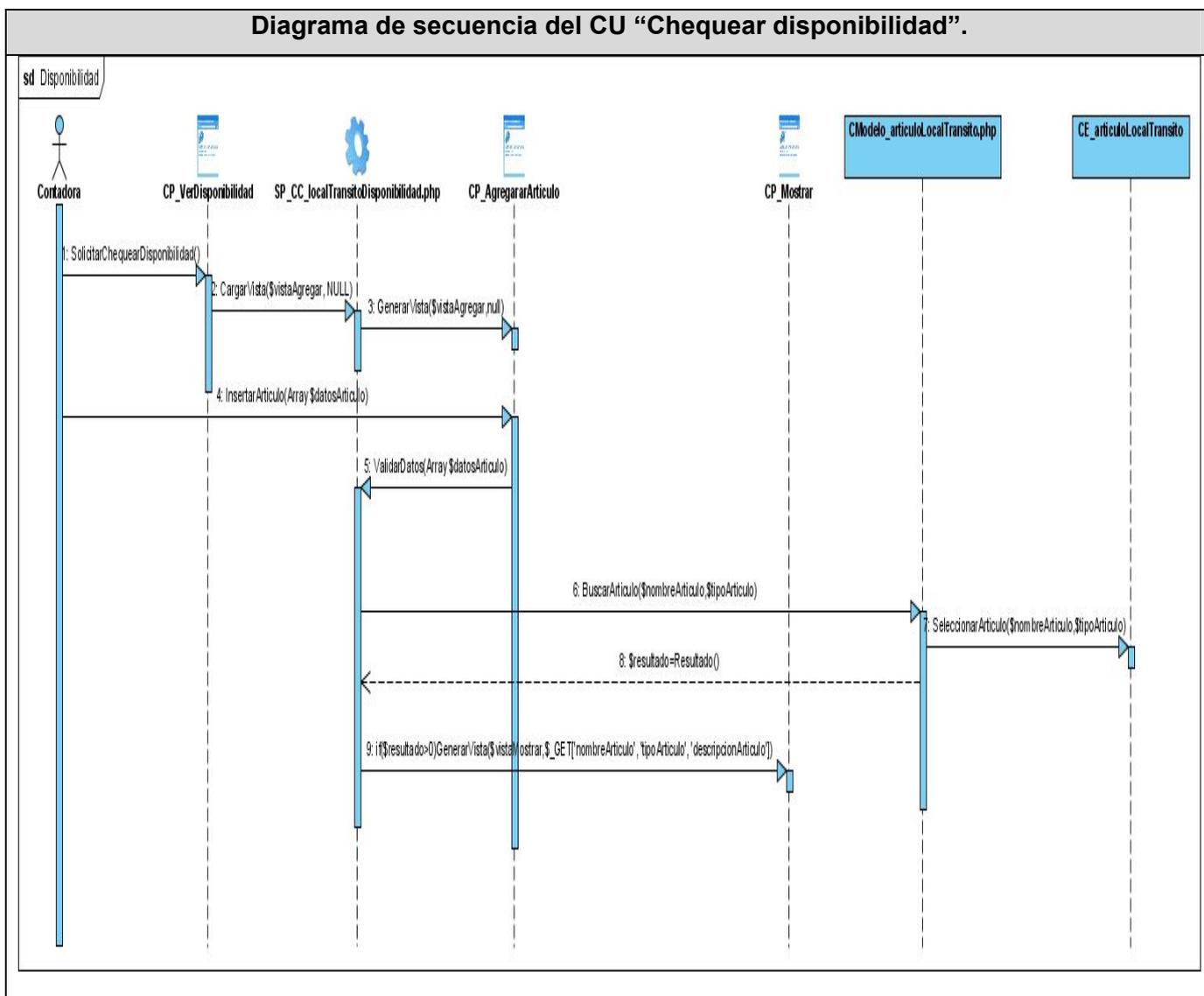
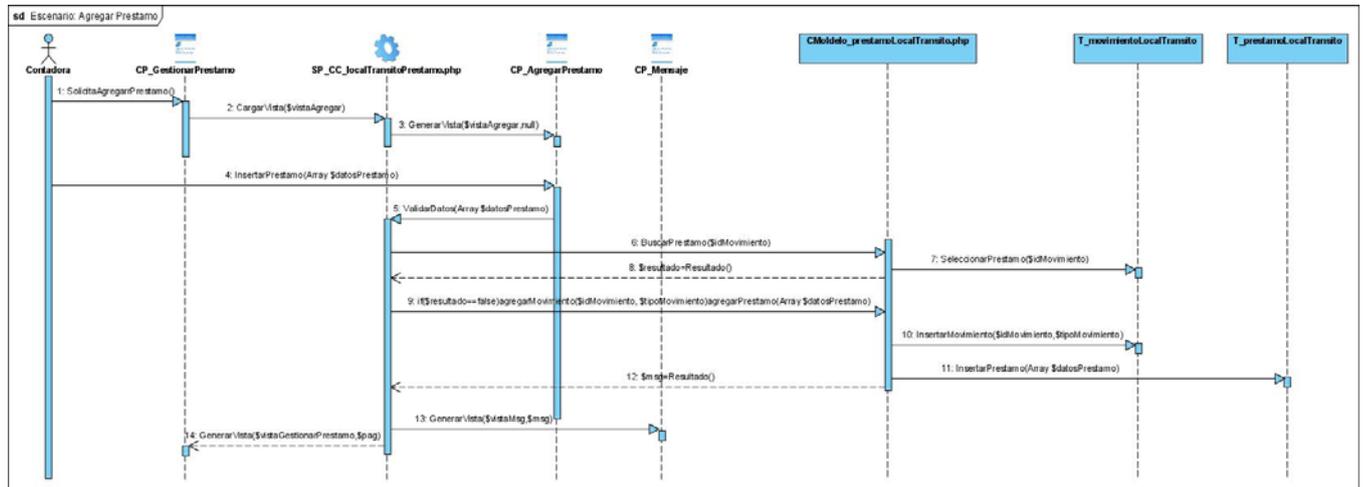


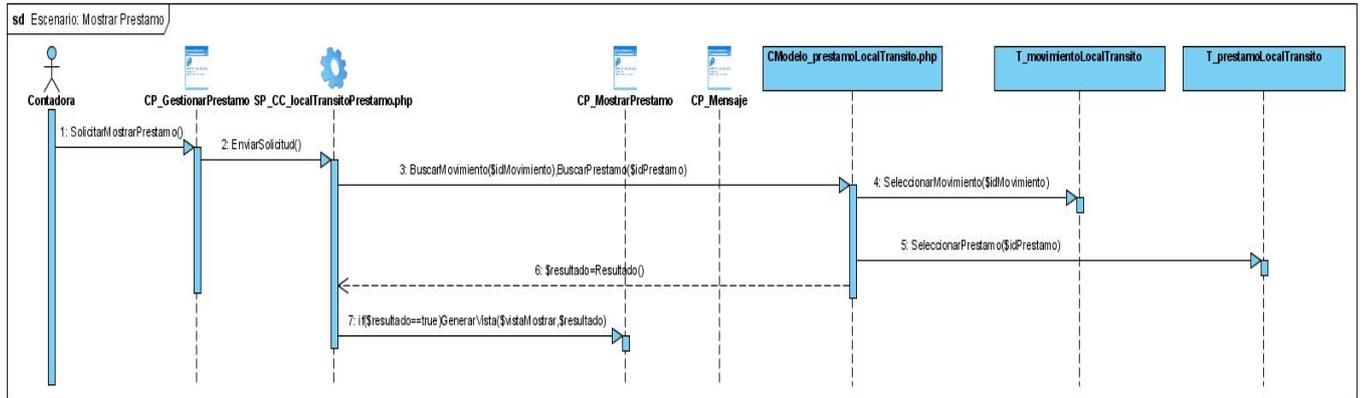
Figura 24: Diagrama de secuencia del caso de uso “Chequear disponibilidad”.

Diagrama de secuencia del CU "Gestionar préstamo".

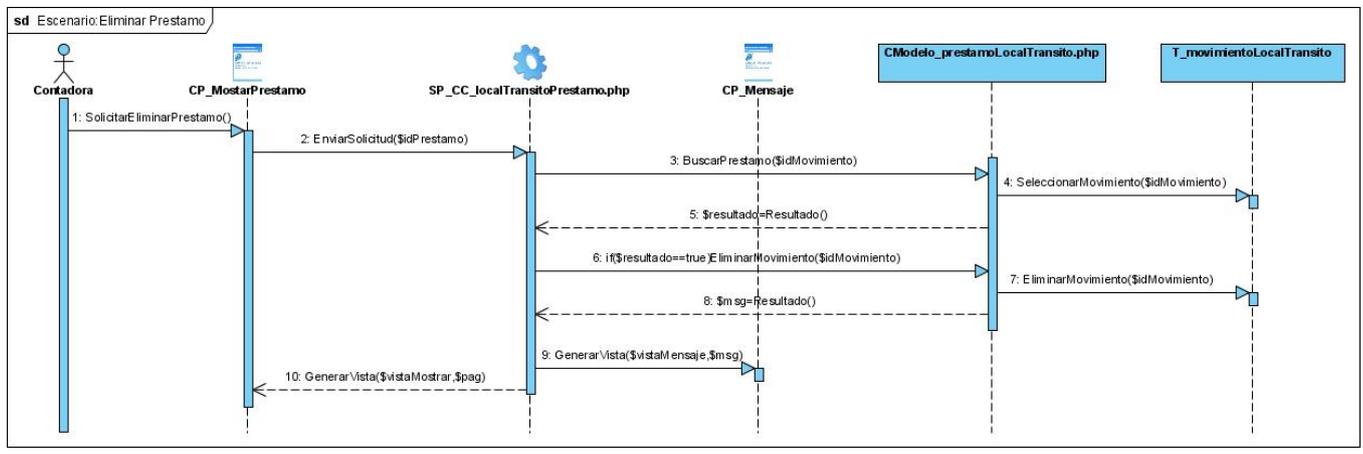
Escenario: Agregar préstamo



Escenario: Mostrar préstamo



Escenario: Modificar préstamo



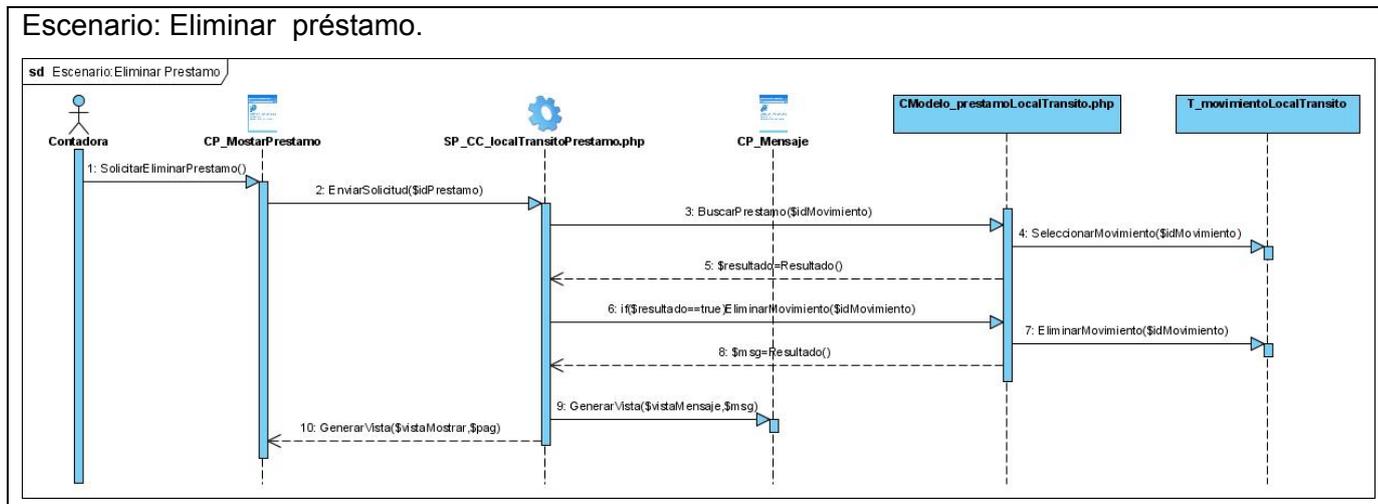
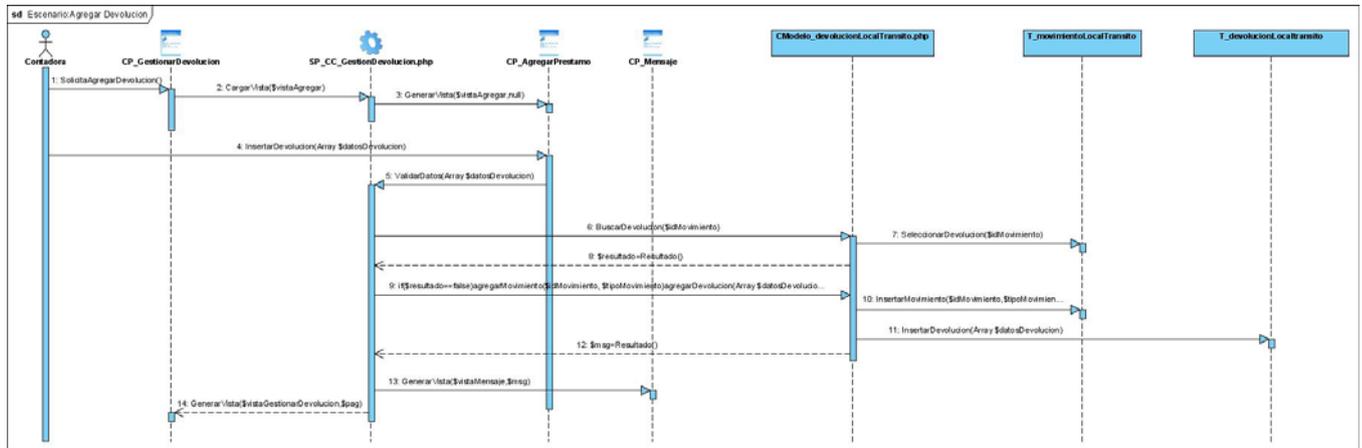


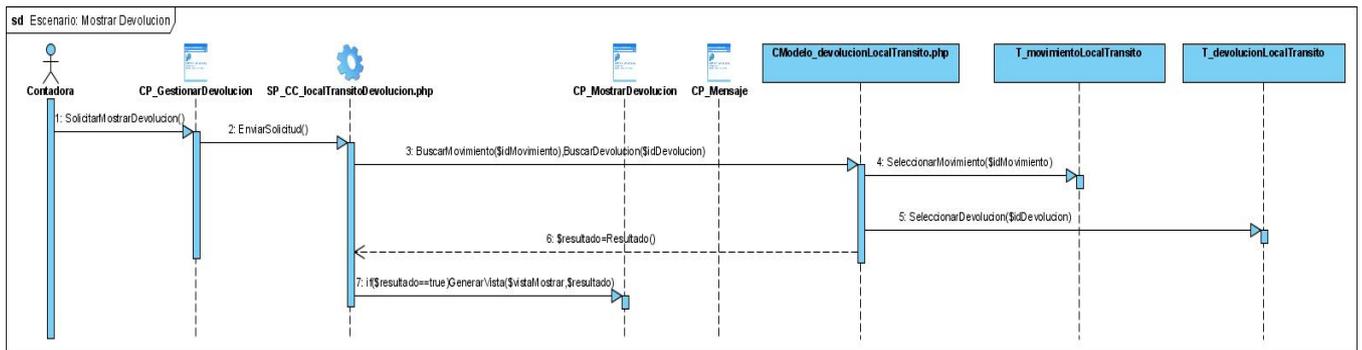
Figura 25: Diagrama de secuencia del caso de uso “Gestionar préstamo”.

Diagrama de secuencia del CU "Gestionar devolución".

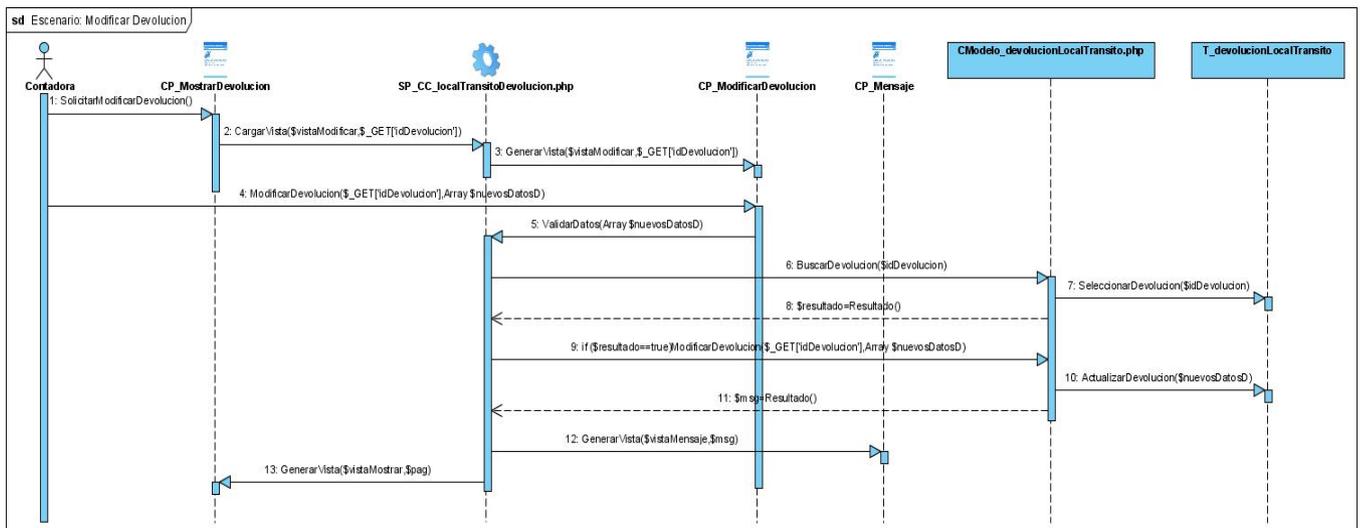
Escenario: Agregar devolución.



Escenario: Mostrar devolución.



Escenario: Modificar devolución.



Escenario: Eliminar devolución.

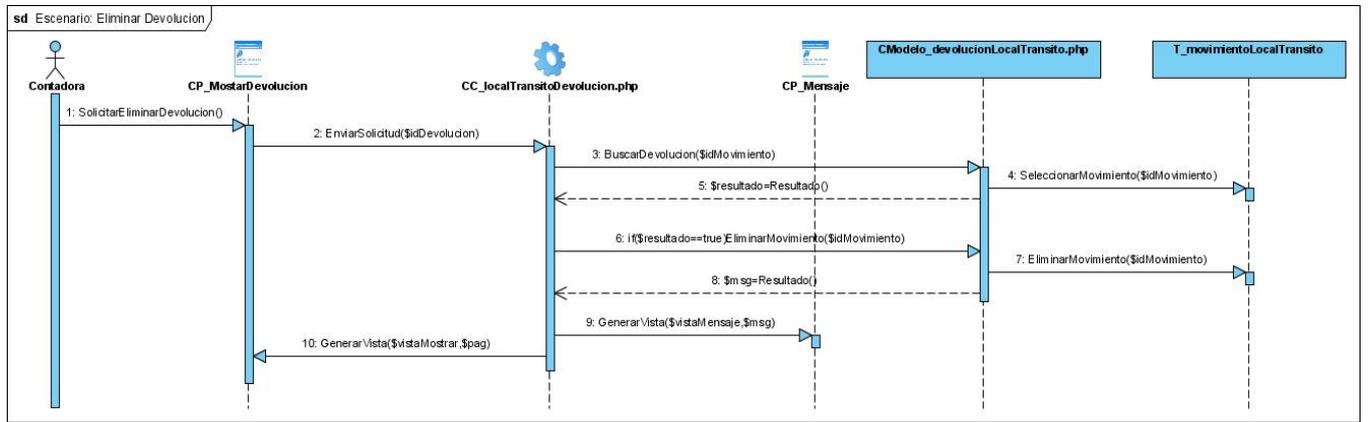
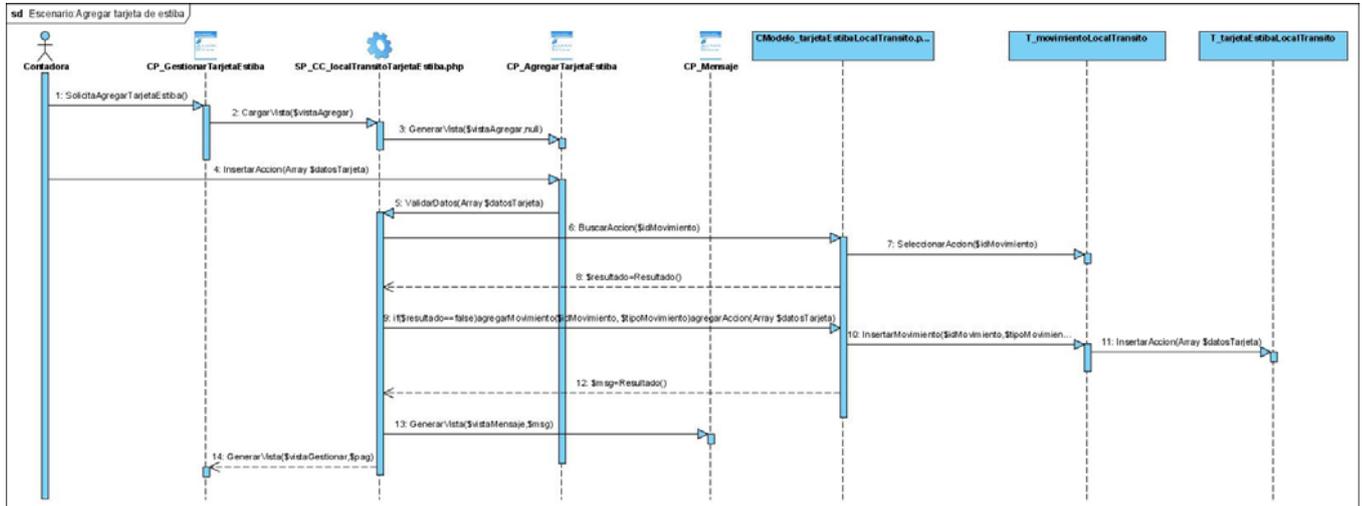


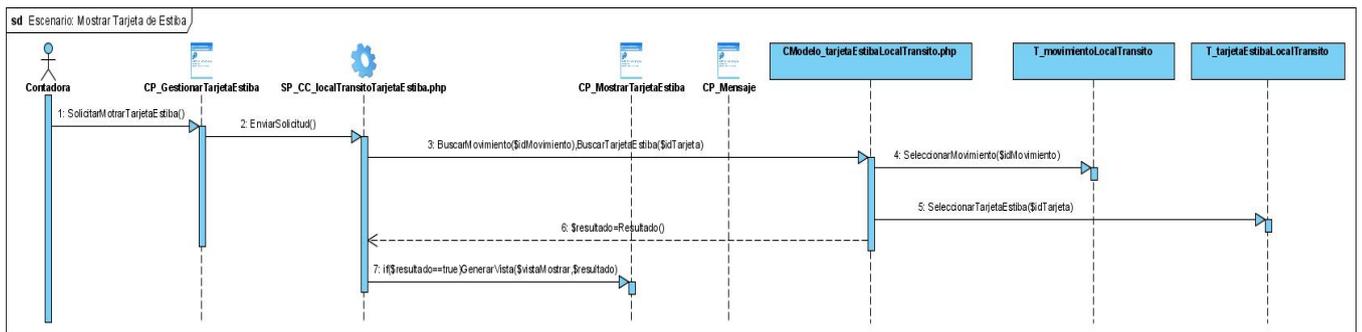
Figura 26: Diagrama de secuencia del caso de uso “Gestionar devolución.”

Diagrama de secuencia del CU "Gestionar tarjeta de estiba".

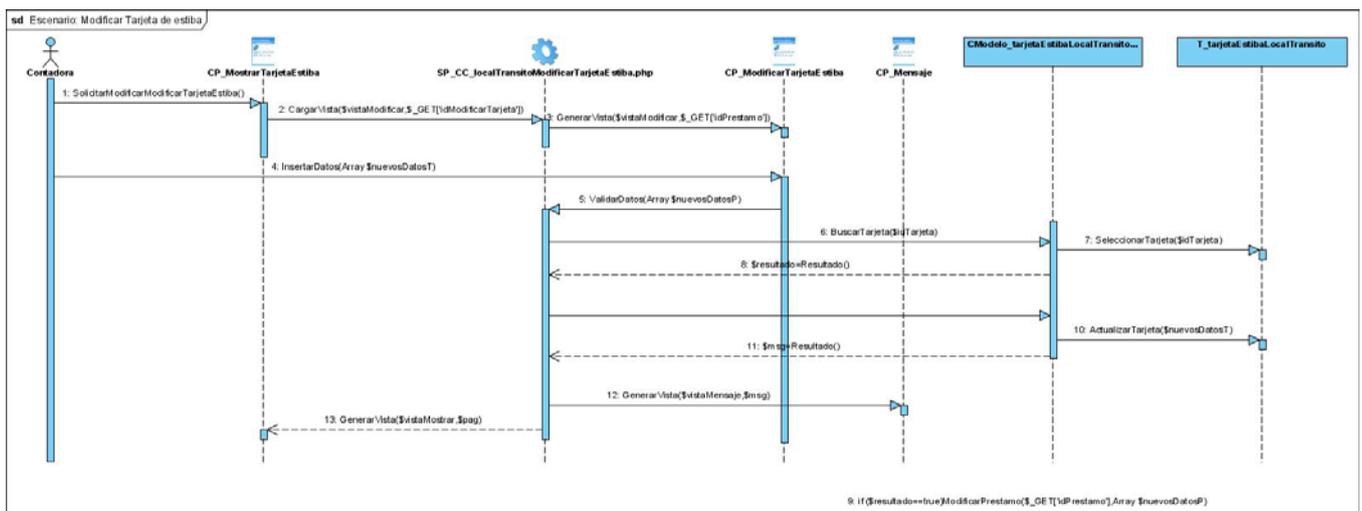
Escenario: Agregar tarjeta de estiba.



Escenario: Mostrar tarjeta de estiba.



Escenario: Modificar tarjeta de estiba.



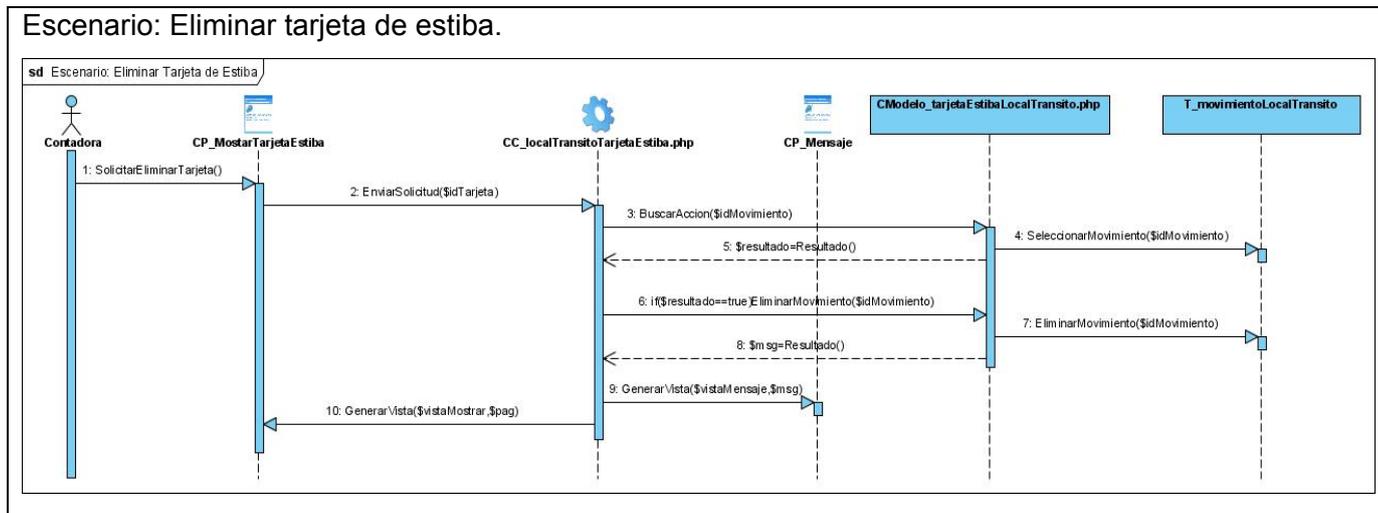
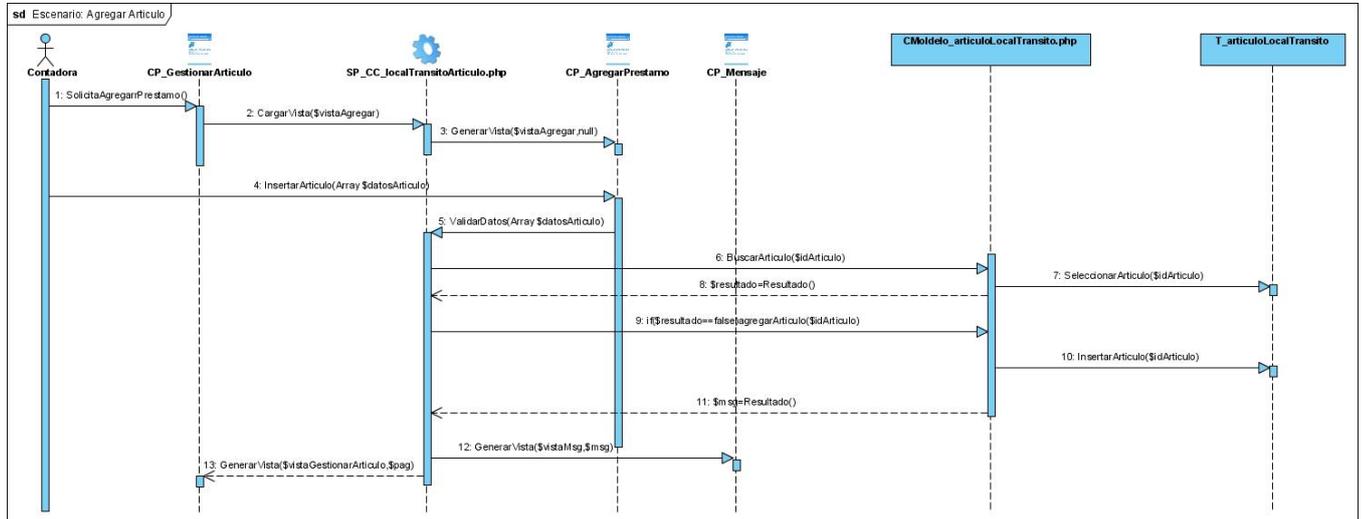


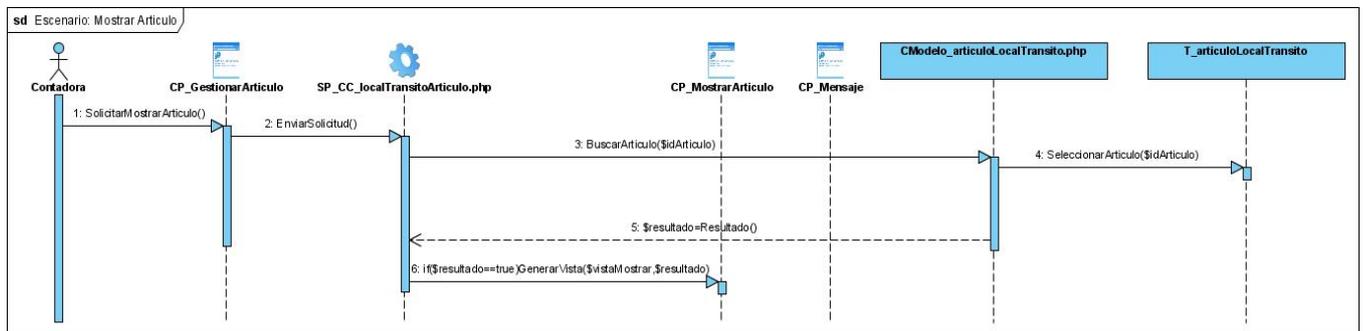
Figura 27: Diagrama de secuencia del caso de uso “Gestionar tarjeta de estiba”

Diagrama de secuencia del CU "Gestionar artículo".

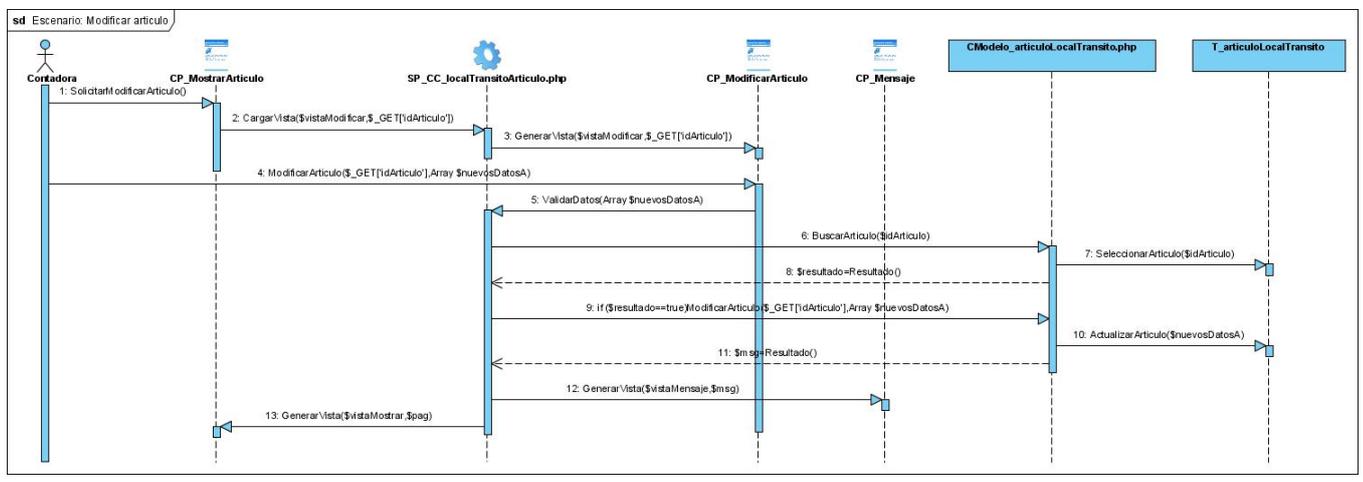
Escenario: Agregar artículo.



Escenario: Mostrar artículo.



Escenario: Modificar artículo.



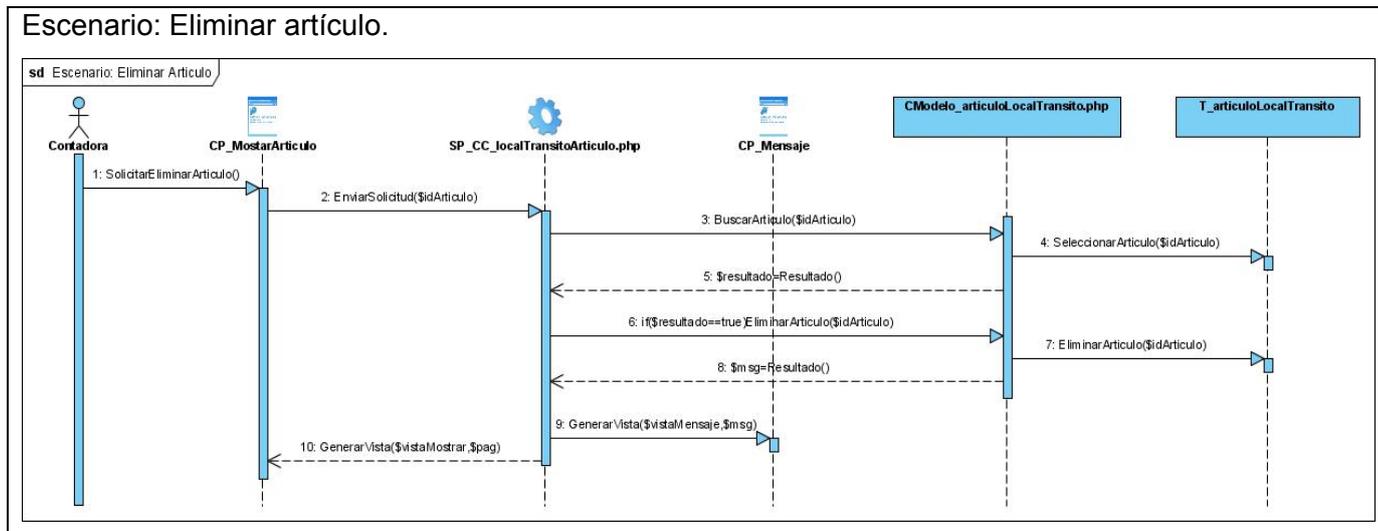


Figura 28: Diagrama de secuencia del caso de uso "Gestionar artículo".

GLOSARIO

- **Acción:** Referente a una entrada o una salida que se registre en el sistema.
- **API:** Interfaz de programación de aplicaciones (*Applications Programming Interface*): una serie de funciones que están disponibles para realizar programas para un cierto entorno.
- **BSD:** *Berkeley Software Distribution*, de la Universidad de California en Berkeley, autora de una versión de UNIX que lleva su mismo nombre.
- **Open source:** Cualidad de algunos softwares de incluir el código fuente en la distribución del programa. En general se usa para referirse al software libre.
- **HTML:** *HyperText Markup Language*, el lenguaje de descripción de páginas habitual en Internet.
- **SGBD:** Sistema de gestión de bases de datos (en inglés *DataBase Management System - DBMS*).
- **LDAP:** es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.
- **TLS:** Protocolo que garantiza la privacidad y la integridad de los datos entre aplicaciones cliente/servidor que se comunican a través de Internet.
- **WML:** *Wireless Markup Language*. Una versión reducida de HTML, utilizada para crear páginas destinadas a las pantallas de terminales móviles.

