

Universidad de las Ciencias Informáticas
Facultad 7



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

Sistema para la Cooperación Médica: Módulo de Trámites

Autores:

Indira Pérez García
Edgar Luis Quesada Pérez

Tutores:

Ing. Diana Rosa Alfonso Espinosa
Ing. Alfredo Rodríguez Ruiz

La Habana, junio del 2011
“Año 53 de la Revolución”



"... no hay satisfacción, ni premio más grande que cumplir con el deber."

Fidel Castro Ruz

Datos de Contacto

Ing. Diana Rosa Alfonso Espinosa (dralfonso@uci.cu): Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2008. Profesora de la Universidad de las Ciencias Informáticas, pertenece al Departamento de Práctica Profesional de la Facultad 7, donde imparte actualmente la asignatura Aplicaciones Informáticas en el Sector de la Salud. Es miembro además del Departamento de Sistemas de Apoyo a la Salud del Centro de Informática Médica (CESIM) el cual desarrolla soluciones informáticas en la rama de la salud. Actualmente se desempeña como Analista principal del proyecto de Colaboración Médica.

Ing. Alfredo Rodríguez Ruiz (arruiz@uci.cu): Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2008. Profesor de la Universidad de las Ciencias Informáticas. Es miembro además del Departamento de Sistemas de Apoyo a la Salud del Centro de Informática Médica (CESIM) el cual desarrolla soluciones informáticas en la rama de la salud. Actualmente se desempeña como Líder del proyecto de Colaboración Médica.

Agradecimientos

Indira:

A mi madre por su ejemplo y su infinito amor, por sus sacrificios, su enseñanza, porque todo lo que soy se lo debo a ella.

A mi querido hermano Ido por todo el amor y el cariño que siempre me ha dado.

A mi padrastro Obdulio que ha sido un padre para mí y me ha sabido guiar siempre por el buen camino con su ejemplo, dedicación, sabiduría y amor en cada una de las cosas que hace.

A mi tía Martha por ser tan atenta conmigo, por comportarse como una madre para mí, por su cariño y apoyo.

A mi tía Zoila y mi prima Ángela que a pesar de la distancia siempre han estado pendientes de mí, ayudándome en lo que necesite.

A mi tía Zaida que me ha dado su apoyo, su cariño y afecto.

A mi novio Javier por su amor y comprensión, por estar a mi lado en momentos difíciles, por confiar en mí y decirme que sí se puede.

A mi sobrina Alicia por su cariño y apoyo.

A mis tías por haberme apoyado siempre, a mi abuela, mis primos y mi familia en general, por estar pendientes de mí y darme su cariño en todo momento.

A Yuliet y Manuel por ser mis amigos del alma, por apoyarme siempre y estar pendientes de mí, por su infinito cariño.

A mis suegros por el cariño y el apoyo que me brindaron, por quererme como una hija.

A mi compañero de tesis por el buen equipo que hicimos juntos, por el trabajo que realizó, por su dedicación, su entrega, por ser mi brujito como nos decimos de cariño y por ser más que amigo un hermano para mí.

A mis compañeras de cuarto con las que he compartido momentos de tristezas y alegrías: a Nuria, Dainiris, Yurlenis, Amailis, Magde, Adriana, Leidanis, Daneisy.

A mis amigos y compañeros de grupo por su afecto y cariño durante estos cinco años, a Cubilla, Carlillo, Ruber, Yuniel, Ines, Ismaray, Alianny, Luismel, Raysel, Renan, Mojena, los jimaguas, Jorgito y todos en general.

A los profesores que han contribuido a mi formación como profesional y se han ganado mi cariño, a Danny, Pedro, Angel, Annia, Beatriz, Yisel, Urco, Vel, Ariskjen, la profe Pura por su ayuda con el documento.

A mi querido profesor y amigo Danny por todos sus consejos, su cariño y su ayuda incondicional.

A mi oponente Ranniel por su ayuda y sus sugerencias.

A los tutores por sus exigencias, que me han enseñado a ser más independiente y que es mejor hacer las cosas por uno mismo sin depender de nadie más.

A los profesores del tribunal por todos sus consejos y sugerencias.

A todos los amigos, que me han ayudado a lo largo de la carrera.

A todos los que de una forma u otra han colaborado con mi formación profesional.

A la dirección de la revolución por darme la oportunidad de estudiar en una universidad como esta.

Edgar:

Agradezco especialmente a mi madre que lo ha sido todo y me lo ha dado todo en esta vida, ha sido mi guía, mi ejemplo, espero no defraudarla nunca y que siempre viva orgullosa del hijo que ha forjado con su sacrificio y dedicación.

Le agradezco a mi familia en general por la confianza que depositaron en mi desde siempre, a mi papá, a mi abuela Olguita, a mi abuelo el Quesa, a mi tío Jorge, a mi tía Ángela Alicia a mi tío Luis Machado, a mi primo Jorge Luis, a mi tía Gloria, a Ernesto, a Paquito y a primo Alfredo.

A tío Juanqui quien le ha aportado muchísimo junto con tío Carlos, a mi concepto de familia.

A mi tío Carlos que me ha enseñado muchas cosas, entre ellas a ser una persona más responsable y con mayores valores humanos muy por encima de cualquier defecto que puedan tener los que nos rodean. Siempre dándome buenos consejos, el cariño y el afecto de un padre. A mis hermanos, que aunque no nos comuniquemos mucho, ellos saben que los llevo en mi corazón.

Le agradezco a mi primo Mario Eduardo por mantener en constante evolución mi paradigma tecnológico e incentivar la creatividad en mi.

A mi familia como base de mi educación y formación académico - profesional.

Un agradecimiento especial a Dora que ha sido como mi madre durante estos 5 años, siempre halándome las orejas y guiándome por el camino correcto, aconsejándome. Madre que depositó una gran confianza en mi aún sin conocerme y a quién le estaré eternamente agradecido por su apoyo en esta etapa de mi vida.

Le agradezco a Antonio por ser amigo y apoyarme en momentos difíciles aconsejándome como lo habría hecho con su propio hijo.

Les agradezco a todas aquellas personas que han contribuido con mi formación profesional.

A mis hermanos de la Universidad: David, MaikeL y René quienes han estado siempre en las buenas y en las malas.

A mis compañeros de apartamento, Carlito, El yuyu, Brayan, y Bradshaw. A las personas que formaron parte de mi grupo durante la carrera: Nurita, Dane, Amailis, Dainiris, Luismel, Raysel, El guille, Arrebato, Jose Carlos, Meme, Franklin, Cucu, el moje, Idayana, Los Gimán y muchos mas que aunque no los mencione a todos, saben que pertenecen a esta lista.

A mi oponente, que ha sido un tutor más que a pesar de ser severo en sus revisiones, nunca faltaron recomendaciones ni ideas encaminadas a mejorar la tesis.

A Ariskjén que a pesar de ser prácticamente nuevo en el proyecto ha sido amigo y ejemplo, muchas veces dejando su trabajo a un lado para ayudarme con algún error o darme alguna vía mejor de solución derivada de su experiencia, sin escatimar en tiempo ni métodos de enseñanza.

A Pompa por brindarme sus conocimientos. A Annia por sacrificar su tiempo libre en aras de ayudarnos.

A mi puro Manolito!!! Que ha sido el me ha dado la formación y preparación para la vida como hombre de trabajo, transmitiéndome su conocimiento y experiencia a lo largo de 15 años de vecinos. Ha sido un padre más, y un ejemplo de persona laboriosa y perfeccionista. Con su frase característica: "Todo es cuestión de método", "Si lo vas a hacer, hazlo bien, sino, no lo hagas!"

Le quiero agradecer este trabajo y no por último menos importante, a mi compañera de tesis mi brujita como le digo de cariño, por su dedicación, responsabilidad, laboriosidad y amistad. Juntos hicimos un dúo encaminado siempre en una misma dirección, alejados de la discordia, porque más que una compañera de tesis y amiga ha sido una hermana.

Dedicatoria

Indira:

Todo el sacrificio y el esfuerzo de este trabajo lo dedico a mi madre por ser el mayor tesoro que me dio la vida, por ser mi fuente de inspiración, mi guía y mi razón de ser. Porque aunque no este presente estará en mi corazón y será siempre mi ejemplo a seguir, a ella dedico todos mis logros.

Edgar:

A mami, que ni con todas las tesis del mundo le pago la educación y el amor que me ha dado.

Resumen

El Sistema para la Cooperación Médica (Colaben) surge a partir de la necesidad que tiene el estado cubano de gestionar la información real de las misiones médicas y los colaboradores en el exterior, concebida por la Unidad Central de Cooperación Médica (UCCM), que no cuenta con un registro único y actualizado de estos datos. El objetivo de la presente investigación es el desarrollo de un módulo para el producto Colaben que informaticice la gestión de trámites de pasaportes de los colaboradores médicos que viajan al exterior.

El módulo se desarrolló a través del uso del lenguaje de programación PHP v5.3. Para una mejor estructuración del código fuente se utilizó el Framework Symfony v1.4. Como Sistema Gestor de Base de Datos se utilizó PostgreSQL v8.3, se eligió Apache v2.2 como el servidor Web sobre el que correrá la aplicación. Se trabajó con la herramienta Enterprise Architect v7.1 para realizar el modelado haciendo uso de la notación UML v2.1 (Lenguaje Unificado de Modelado). Se desarrolló el módulo en plataformas de software libre, con el uso del Sistema Operativo Linux Distribución Ubuntu v10.04.

El módulo desarrollado constituye una gran ayuda para el Departamento de Trámites de la UCCM, pues permite agilizar la gestión de los documentos asociados a los colaboradores médicos a través de un medio informático que brinda además reportes acerca de los mismos. Con la automatización de la tramitación de estos documentos se obtiene una mejor organización y rápido acceso a la información que se almacena.

Palabras claves: *Colaboradores, Cooperación, Pasaportes.*

Índice

Introducción.....	1
Capítulo 1. Fundamentación Teórica	6
1.1 Conceptos Fundamentales.....	6
1.2 Estudio del Arte.....	7
1.2.1 Nivel Internacional	7
1.2.2 Nivel Nacional	8
1.2.3 Universidad de las Ciencias Informáticas (UCI).....	9
1.3 Tecnologías, Metodologías y Lenguajes utilizados.....	10
1.4 Herramientas.....	16
1.5 Arquitectura del Sistema.....	17
Capítulo 2. Características del Sistema	19
2.1 Descripción de los Procesos del Negocio	19
2.1.1 Proceso Confeccionar Solicitud de Trámite de Pasaporte	19
2.1.2 Proceso Tramitar Pasaporte.....	20
2.1.3 Proceso Emitir Libro de Entrada.....	20
2.1.4 Proceso Entrada de Pasaporte	20
2.1.5 Proceso Salida de Pasaporte	21
2.1.6 Proceso Incinerar Pasaporte.....	21
2.2 Modelo del Negocio	21
2.2.1 Descripción de Procesos del Negocio.....	22
2.3 Especificación de Requisitos de Software.....	24
2.3.1 Requerimientos Funcionales	24
2.3.2 Diagrama de Paquetes de Requisitos Funcionales	26
2.3.3 Requerimientos No Funcionales	26
2.4 Definición de los Casos de Uso del Sistema.....	30
2.4.1 Descripción de los Actores del Sistema.....	30
2.4.2 Vista Global de los Actores del Sistema	31
2.4.3 Lista de Casos de Uso.....	31
2.4.4 Diagrama de Casos de Uso del Sistema	32
2.4.5 Descripción Textual de los Casos de Uso del Sistema	32
Capítulo 3. Análisis y Diseño del Sistema.....	36
3.1 Descripción de la Arquitectura	36

3.2	Modelo de Análisis	39
3.2.1	Clases del Análisis	39
3.2.2	Diagrama de Clases del Análisis	39
3.2.3	Diagramas de Comunicación.....	41
3.3	Modelo de Diseño	44
3.3.1	Descripción de Elementos del Diseño.....	44
3.3.2	Diagramas de Clases del Diseño	45
3.3.3	Descripción de las Clases del Diseño.....	50
3.3.3.1	Descripción de Páginas Clientes	50
3.3.3.2	Descripción de Páginas Servidoras	50
3.4	Diagrama de Despliegue	50
Capítulo 4.	Implementación	52
4.1	Modelo de Datos	52
4.2	Modelo de Implementación.....	53
4.2.1	Diagrama de Componentes.....	54
4.3	Tratamiento de Errores	56
4.4	Seguridad	56
	Conclusiones.....	58
	Recomendaciones	59
	Referencias Bibliográficas	60
	Bibliografía	63
	Anexos	67
	Glosario de Términos.....	71

Introducción

En el pasado siglo el desarrollo informático impulsó el conocimiento, al romper barreras antes inalcanzables y logrando avances en la ciencia y la tecnología. Además este desarrollo es considerado el pilar del avance tecnológico en la actualidad, trazando nuevas pautas hacia un control de la información abierto, dinámico y orientado a procesos.

La informática, se ha convertido en el eslabón fundamental del desarrollo económico, político y social de Cuba. Hoy día, con el creciente auge de la informatización en todas las esferas de la sociedad y el acelerado desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC), la mayoría de las empresas e instituciones requieren automatizar los procesos de trabajo que realizan.

En Cuba, antes del triunfo de la revolución no se brindaba ningún tipo de colaboración médica, pero en 1960 se ofrece ayuda médica internacional, enviando una brigada de colaboradores a Chile, después del terremoto que azotó a ese país. A finales de 1962, en el acto inaugural de la Facultad de Ciencias Básicas y Pre-clínicas "Victoria de Girón", el compañero Fidel Castro anunció al pueblo de Cuba la decisión del gobierno revolucionario de brindar ayuda médica internacional en el campo de la salud. Esta expresión se pone de manifiesto al año siguiente al comenzar la colaboración médica internacional, con el envío de la primera brigada médica cubana a Argelia.

Entre las funciones llevadas a cabo por la colaboración médica cubana se destaca la asistencia clínico-quirúrgica a la población de los países en los lugares más apartados, además del desarrollo conjunto de las campañas de educación sanitaria y de vacunación masiva de la población, el control higiénico-epidemiológico y la prestación de servicios en brigadas médicas a damnificados por huracanes, sismos y otras contingencias en diferentes países. La mayor expresión de solidaridad e internacionalismo de la colaboración médica de Cuba, se evidenció a finales de 1998 tras el azote del huracán Mitch a varios países de Centro América, principalmente a las naciones de Honduras y Guatemala. (1)

Durante los 45 años de colaboración médica más de 37 800 trabajadores de la salud han cumplido misiones en 83 países. De ese impresionante número de trabajadores los médicos sobrepasan los 25 700, quienes en su mayoría han brindado sus conocimientos en zonas intrincadas y de difíciles condiciones de vida, al ser portadores de la calidad científica y el ejemplo de los profesionales formados por la revolución. (2)

Debido a la gran cantidad de cooperantes cubanos, decididos a brindar ayuda solidaria a países hermanos, surge la necesidad de crear la Unidad Central de Cooperación Médica (UCCM), con el objetivo de gestionar toda la información posible de las personas que se encuentran cumpliendo misión dentro y fuera del territorio nacional. La UCCM está constituida básicamente por varios departamentos, entre ellos se encuentra el Departamento de Trámites, que es el encargado de llevar a cabo todo lo relacionado con el proceso de gestión de trámites de pasaportes de los colaboradores que van a salir de misión.

En la actualidad, los procesos de confección, rehabilitación y prórrogas de pasaportes a los colaboradores médicos se realizan de forma manual, haciéndose difícil la gestión de información de los mismos. Como resultado de estos procesos se genera, por cada colaborador, un conjunto de modelos y solicitudes cuya gestión se vuelve engorrosa, debido al alto número de colaboradores que diariamente salen a cumplir misión internacionalista.

La inexistencia de un mecanismo en tiempo real de actualización y consulta dinámica de los datos relacionados a estos procesos, imposibilita llevar a cabo un análisis estadístico de la información manejada en el Departamento de Trámites, prolongando el tiempo en la toma de decisiones.

Estos problemas afectan directamente a los colaboradores de la salud, a la UCCM, al Ministerio de Salud Pública (MINSAP), al Ministerio de Relaciones Exteriores (MINREX) y por consiguiente al Consejo de Estado y al país.

Debido a lo planteado anteriormente, el **problema a resolver** de esta investigación se enfoca en: ¿Cómo agilizar el proceso de gestión de la información de trámites de pasaportes en la Unidad Central de Cooperación Médica?

El **objeto de estudio** del presente trabajo de investigación se centra en el proceso de gestión de la información en la Unidad Central de Cooperación Médica.

Enmarcándose en el **campo de acción** el proceso de gestión de la información en el Departamento de Trámites de la Unidad Central de Cooperación Médica.

Por lo que se propone como **objetivo general** desarrollar el Módulo de Trámites del Sistema para la Cooperación Médica.

Para dar solución al objetivo trazado se presentan las siguientes **tareas de la investigación**:

- Realizar un estudio del arte de los sistemas existentes a nivel nacional e internacional relacionados con la gestión de trámites de pasaportes, que sirvan de punto de partida para la propuesta del sistema.
- Asimilar los procesos realizados dentro del Departamento de Trámites de la Unidad Central de Cooperación Médica, obteniéndose una visión general del negocio.
- Aplicar las herramientas y tecnologías propuestas por el Departamento Sistemas de Apoyo a la Salud (SAS), para el desarrollo de la solución.
- Generar los artefactos que propone el Proceso Unificado de Desarrollo (RUP) en los flujos de trabajo: Modelado de Negocio, Requerimientos, Análisis y Diseño, Implementación y Prueba, sirviendo de base a los desarrolladores del sistema.
- Desarrollar el módulo aplicando las pautas de diseño definidas por el Departamento SAS, siguiendo lo establecido en la Especificación de Requisitos de Software.

En el presente trabajo se utilizaron los métodos de investigación, estos permiten estudiar las características del objeto que no son observables directamente. Como parte de los mismos se utilizaron los métodos teóricos que fueron empleados de la siguiente manera:

- Método de Análisis Histórico-Lógico:

Se utilizó durante todo el desarrollo de la investigación, incluye el estudio de la evolución de los sistemas nacionales e internacionales sobre la gestión de trámites de pasaporte, el análisis de los procesos que dieron lugar a la creación de los mismos, además la exploración de las posibles herramientas que se emplearán en la solución.

- Método Modelación:

Como método se convierte en un instrumento de la investigación de carácter material teórico, creado en aras de reproducir simplificada y subjetivamente la parte de la realidad objetiva que se estudia como objeto. En la presente investigación se utiliza este método científico en la representación de los modelos de procesos de negocio, de casos de uso del sistema, de requerimientos y de despliegue que se encuentran dentro de los modelos que se desarrollan durante la investigación propuesta.

De los métodos empíricos se emplearon:

➤ Método Entrevista:

Es una conversación planificada para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado, puede influir en determinados aspectos de la conducta humana por lo que es importante una buena comunicación. Este método fue aplicado en la UCCM durante la etapa de familiarización con el flujo de los procesos que allí se desarrollan y que serán informatizados posteriormente, proporcionando los datos necesarios para las siguientes etapas de la investigación. (3)

La solución propuesta ofrece al Sistema para la Cooperación Médica un módulo que brinda los siguientes beneficios:

- Permite agilizar la gestión de los documentos asociados a los pasaportes de los colaboradores médicos, eslabón fundamental para su salida a cumplir misión.
- Posibilita desarrollar análisis estadísticos de la información manejada en el Departamento de Trámites, de manera ágil y organizada.
- Garantiza la disponibilidad de la información para una correcta toma de decisiones.

El módulo desarrollado tiene gran impacto social pues:

- Proporciona la obtención de un registro actualizado y único de los pasaportes de los colaboradores médicos en el exterior.
- Provee reportes con información detallada y consolidada de los trámites realizados a cada colaborador.
- Permite el cumplimiento en tiempo de los convenios de cooperación internacional del país.
- Ayuda a disminuir los gastos económicos considerablemente ya que disminuye el margen de errores en las confecciones de pasaportes, las pérdidas y extravío de los mismos.
- Garantiza un mejor control interno en la custodia y conservación de los pasaportes en el momento en que el colaborador se encuentra en Cuba.

Para dar cumplimiento de manera satisfactoria a las tareas propuestas en la investigación, se ha estructurado el presente documento en cuatro capítulos, donde se explica detalladamente todo lo relacionado con el Módulo de Trámites del Sistema para la Cooperación Médica.

Capítulo 1: Fundamentación Teórica, se exponen los diferentes aspectos teóricos relacionados con el desarrollo de este trabajo. Se abordan también conceptos fundamentales relacionados directa o indirectamente con el proceso de trámites de pasaporte de los colaboradores para un mejor entendimiento del mismo, el estado del arte, los antecedentes del sistema, un análisis de las tecnologías en las cuales se sustentará el desarrollo del sistema, así como las herramientas que se utilizarán para el desarrollo del mismo.

Capítulo 2: Características del Sistema, se detallan las características generales y específicas del sistema. Se describen también los procesos de negocio, así como los diagramas de actividades y los requisitos tanto funcionales como no funcionales.

Capítulo 3: Análisis y Diseño, se describe el sistema a través de diagramas de clases del análisis y diseño. Se realizan los diagramas de interacción por cada realización de caso de uso que apoya luego la fase de implementación. Se obtienen las clases del diseño que representan a la aplicación, además de una explicación sobre los principios de diseño que se tuvieron en cuenta para la realización del sistema, así como la descripción de la interfaz, la descripción de la seguridad y la concepción de la ayuda para un mejor manejo de la aplicación.

Capítulo 4: Implementación, lleva el resultado del diseño a componentes que son ficheros de código fuente y scripts; estos son los archivos que finalmente integran la aplicación de forma física. En el capítulo se indica cómo los elementos del diseño se implementarán y se organizarán de acuerdo con el diagrama de despliegue, que permite la configuración de nodos de procesamiento y los componentes que residen en ellos. Además, se realizará el diagrama de componente general que organiza las dependencias entre un conjunto de componentes; todo dando lugar al buen control de la aplicación.

Capítulo 1. Fundamentación Teórica

El presente capítulo tiene como objetivo fundamental abordar distintos temas que sirven de soporte teórico a la investigación, así como la realización de un estudio detallado de las diferentes tecnologías existentes en la actualidad, plasmándose los aspectos fundamentales que influyeron en su elección.

Se abordan detalladamente los conceptos fundamentales para un mejor entendimiento del sistema que se desea desarrollar, así como un análisis previo de sistemas que a nivel nacional e internacional tienen similitud con la gestión de la información de trámites de pasaportes.

Se realiza un análisis de la metodología a utilizar, el servidor de bases de datos, el servidor Web, y los lenguajes que serán usados para el desarrollo de la aplicación. Finalmente, se hace referencia a las herramientas que son necesarias en la ingeniería y desarrollo de la aplicación.

1.1 Conceptos Fundamentales

Cooperación

La cooperación es el aporte de origen externo que se solicita a países u organismos internacionales, con el fin de apoyar el desarrollo nacional mediante acciones, proyectos y programas específicos. Además, se refiere abstractamente a todo proceso en donde se involucre el trabajo de varias personas en conjunto. (4)

Pasaporte

Documento personal que acredita la identidad y la nacionalidad de una persona y que es necesario para viajar a cualquier país.

Trámite

Estado de un proceso administrativo por el que tiene que pasar un asunto para ser solucionado.

Gestión de la Información

Proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma. (5)

1.2 Estudio del Arte

1.2.1 Nivel Internacional

Teniendo en cuenta las características que presenta el proceso de realización de trámite de pasaportes para colaboradores en misiones médicas, es de gran importancia destacar que en el mundo existen diferentes sistemas de software enfocados a la gestión de trámite de pasaportes que son utilizados por varias empresas. Entre estos sistemas se encuentran:

Servicio Administrativo de Identificación, Migración y Extranjería (SAIME):

El Servicio Administrativo de Identificación, Migración y Extranjería, es un organismo adscrito al Ministerio de Relaciones Interiores de Venezuela, cuyo objetivo principal es planificar y ejecutar las actividades correspondientes a la identificación de las personas naturales que habitan en el territorio nacional de acuerdo con lo establecido en la Ley Orgánica de Identificación, así como la coordinación, supervisión, dirección y control de las distintas actividades para el otorgamiento de las solicitudes de los ciudadanos extranjeros, siempre que cumplan con las disposiciones de la ley. (6)

Realizándose una comparación entre los procesos que implementa este sistema, y aquellos observados en el Departamento de Trámites de la UCCM es imposible encontrar puntos en común entre ambos negocios, debido a que el primero gestiona los pasaportes de personas naturales en las oficinas destinadas a tal fin en la República Bolivariana de Venezuela mientras que el Módulo de Trámites se encarga del estado de los pasaportes de los colaboradores a nivel ministerial. Además, el sistema SAIME rige sus procesos por leyes propias del estado venezolano, por otro lado es un sistema desarrollado para cumplir con dichas leyes y no contempla modificaciones o actualizaciones que permitan adaptarlo a otros procesos de negocio.

Sistema de Trámites Online (SITRON):

Es un sistema Argentino que permite hacer trámites de Cambios de domicilio, solicitud de nuevo ejemplar de Documento Nacional de Identidad (DNI), renovación de DNI, certificación y reposición del mismo y/o gestiones relacionadas con pasaportes y certificaciones. Muchos de los servicios que brinda son los trámites que requieren intervención consular (visas, certificados, pasaportes, legalizaciones, poderes, y otros), así como también los trámites del Registro Nacional de las Personas (DNI, Cambios de domicilio, entre otros.). (7)

Este sistema ofrece una visión general de los procesos que realiza, y algunos de estos procesos no se corresponden con los que se desean implementar en el Módulo de Trámites, además se rige por la legislación de su país, razón suficiente para que no pueda ser utilizado.

1.2.2 Nivel Nacional

Cuba es un país socialista que se caracteriza por poseer un alto nivel de colaboración internacional, por lo que se hace necesario gestionar todos los procesos de recogida de datos de los viajeros; con el objetivo de confeccionar sus documentos para que puedan efectuar sus viajes. A continuación se describen algunos sistemas de trámites migratorios existentes en Cuba.

Sistema Atención a Misiones del Ministerio de la Industria Sideromecánica (SIME):

En este Ministerio se utiliza este sistema desde el año 2000, fue elaborado por la Empresa de Servicios Informáticos de Pinar del Río. Se emplearon 4 meses en su elaboración sin la utilización de alguna metodología de desarrollo, solamente se basaron en la interacción programador-cliente y las entrevistas para conocer los requisitos. El sistema es una aplicación de escritorio realizada en Delphi, que presenta grandes beneficios a esta empresa, los servicios que brinda son los siguientes:

- Búsqueda de información de los misioneros que han viajado y de las misiones realizadas.
- Búsqueda sobre el estado de los pasaportes.
- Genera los modelos oficiales AO1, AO3 y AO4 de la Dirección de Inmigración y Extranjería.
- Brinda un sistema de reportes.

El uso del sistema indudablemente les ahorra tiempo y fuerza de trabajo a los responsables de tramitar misioneros del SIME, pero por otro lado, desde hace varios años la Empresa de Servicios Informáticos de Pinar del Río no brinda soporte a este sistema, por lo que en la actualidad las necesidades de la empresa no son cubiertas totalmente por la aplicación. Una de las necesidades de este ministerio es que las empresas y grupos empresariales adjuntos al mismo, utilicen la aplicación con el objetivo de tener toda la información en la misma base de datos, necesidad que pudiera ser satisfecha si el sistema fuera una aplicación web. Además, en la actualidad esta empresa realiza una serie de trámites a extranjeros que la aplicación no cubre por completo, debido a que en el momento que se desarrolló, estas gestiones no se realizaban.

No es conveniente emplear dicho sistema, la primera razón es que es una aplicación de escritorio que tiene como principales desventajas que se debe realizar la instalación de la misma en cada una de las PC a las que se desea acceder, y además de esto no se tiene centralizado quien usa la aplicación y en

cuantas PC esté instalado, la segunda es que no recoge toda la información de los procesos para la gestión de trámites de pasaportes en la UCCM, por ejemplo; a la hora de hacer algún reporte, no permite distintos criterios de búsqueda en la base de datos, además de que la información reflejada en los mismos se muestra de forma desorganizada e incoherente.

Sistema de Tramitación del Ministerio de Educación Superior:

Este sistema opera sobre Web, e incluye una potente herramienta para realizar cualquier tipo de reporte. Tiene el inconveniente que por cada cliente que hace una petición al sistema, se ejecuta una aplicación diferente en el servidor, lo cual no constituye un problema en una empresa o institución pequeña y con recursos limitados, pero es un inconveniente al haber un gran número de usuarios que acceden al sistema en una Institución de mayores recursos.

No es viable adoptar este sistema, porque es insuficiente dadas las necesidades actuales, además la base de datos está diseñada en Microsoft Access, gestor que no ofrece mucha seguridad en la protección de la información. Su estructuración por departamentos no coincide con la organización de la UCCM y contiene términos, clasificaciones y automatiza procesos que no son los mismos con los cuales se trabaja en el Departamento de Trámites de la UCCM.

1.2.3 Universidad de las Ciencias Informáticas (UCI)

Sistema Informatizado de Cooperación Internacional:

Actualmente la UCI cuenta con una Dirección de Cooperación Internacional donde el grupo de trámites es el encargado de tramitar todo lo que esté relacionado a los asuntos de migración y extranjería, ya sea por razones de estudios o salidas por misiones, con el objetivo de facilitar las actividades, tales como la gestión de la documentación, elaboración de los reportes estadísticos y económicos de las personas que viajan y las misiones, además de los partes sobre el estado de estas últimas y del proceso de tramitación de los misioneros. (8)

La Dirección de Cooperación Internacional cuenta con un sistema desarrollado en .NET que automatiza los procesos del grupo de trámites y algunos que han sido definidos como procesos del Grupo de Cooperación Internacional. Este sistema no se ajusta al volumen de información que maneja el Departamento de Trámites de la UCCM, además de que está desarrollado en .NET y utiliza MySQL como gestor de base de datos. Por lo planteado anteriormente es necesario destacar que no es aconsejable adoptar este sistema.

1.3 Tecnologías, Metodologías y Lenguajes utilizados

Proceso Unificado de Desarrollo (RUP)

El Proceso Unificado de Desarrollo, es una metodología de desarrollo de software basada en el Lenguaje Unificado de Modelado (UML). RUP, además de ser un proceso de desarrollo es también un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto.

Este proceso, divide en 4 fases el desarrollo del software: Inicio, Elaboración, Construcción y Transición, cada una de ellas con los objetivos de determinar la visión del proyecto, definir la mejor arquitectura, obtener la capacidad operacional inicial y llegar a obtener el *reléase* del proyecto respectivamente. Estas fases son desarrolladas a través de iteraciones y ciclos de desarrollo, en una misma fase puede que exista más de una iteración, en cambio, un ciclo se extiende desde la fase de inicio hasta la de Transición, al final del mismo se obtiene una versión del producto. Durante una iteración se desarrollan los nueve flujos de trabajo: Modelo de Negocio, Requerimiento, Análisis y Diseño, Implementación, Prueba, Instalación, Ambiente, Administración de Proyecto, Administración de Configuración y Cambios, en unos el esfuerzo requerido es mayor que en otros, en dependencia de la fase en que se encuentre el desarrollo del proyecto. (9)

Las características que definen a RUP son:

- Dirigido por casos: Se utiliza respectivamente para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. Los Casos de Uso son el instrumento para validar la arquitectura del Software y extraer los casos de prueba.
- Centrado en la arquitectura: Incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura es una vista del diseño completo con las características más importantes resaltadas, lo cual ayuda al arquitecto a centrarse en los objetivos adecuados, ya sea la comprensibilidad, la capacidad de adaptación al cambio y la reutilización.
- Iterativo e incremental: Las iteraciones ofrecen como resultado un incremento del producto desarrollado que mejora las funcionalidades del sistema en desarrollo, es decir, las iteraciones hacen referencias a pasos en el flujo de trabajo y los incrementos, al incremento del producto. (10)

En la actualidad constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Propone cómo deben ser ejecutadas las actividades para la obtención de los productos típicos de trabajo y la forma de documentar todas las acciones que se llevan a cabo en las mismas, además soporta el ciclo completo de desarrollo de la aplicación.

Lenguaje Unificado de Modelado (UML)

UML v2.1 es un lenguaje de modelado visual, está compuesto por diversos elementos gráficos que se combinan para conformar diagramas, y puede ser usado en todas las fases de desarrollo del software. Su función principal es especificar, visualizar, construir y documentar artefactos de un sistema de software. Se utiliza para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

UML es un lenguaje más expresivo, claro y uniforme que el resto definidos para el diseño Orientado a Objetos (OO), que no garantiza el éxito de los proyectos pero si mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios, razón por la cual se escoge para el modelado del sistema a desarrollar. (11)

Sistema Gestor de Bases de Datos (SGBD)

Un Sistema Gestor de Base de datos (SGBD) o DBMA (Data Base Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, además de garantizar la seguridad e integridad de los mismos. (12)

PostgreSQL

PostgreSQL v8.3 es un sistema gestor de base de datos relacional orientada a objetos y libre, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales.

Dentro de sus principales ventajas se encuentran:

- Soporta distintos tipos de datos.
- Posee una gran escalabilidad, haciéndolo idóneo para su uso en sitios web que atienden un gran número de solicitudes.
- Puede ser instalado un número ilimitado de veces sin temor de sobrepasar la licencia.
- Posee estabilidad y confiabilidad legendaria.
- Es extensible a través del código fuente disponible sin costos adicionales.
- Soporte nativo para los lenguajes más populares del medio: PHP, C++, Perl, Python, entre otros.
- Extensiones para alta disponibilidad, nuevos tipos de índices, datos especiales, minería de datos, entre otros.

- Es multiplataforma, disponible en Linux, Unix, Mac Os X y Windows, entre otros sistemas operativos. (13)

Se seleccionó PostgreSQL v8.3, pues se tuvo en cuenta la necesidad de utilizar herramientas libres, para el desarrollo del módulo, además de que es un gestor multiplataforma, confiable, estable, con gran escalabilidad, control de concurrencia y funcionalidades que lo destacan como uno de los SGBD más potentes en la actualidad.

Ajax

Ajax (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma, es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. Es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript.

No es un lenguaje de programación sino un conjunto de tecnologías (HTML-JavaScript-CSS-DHTML-PHP/ASP.NET/JSP-XML) que permiten hacer páginas de Internet más interactivas. (14)

Se decide la utilización de Ajax para las validaciones y el envío de datos a través de las páginas.

Framework Symfony

Es un framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Además, automatiza las tareas más comunes, permitiéndole al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas Unix (Unix, Linux, etc.) como en plataformas Windows. Soporta Ajax y plantillas. (15)

Sus características son:

- Escalable: es infinitamente escalable si se disponen de los recursos necesarios.
- Seguro: permite controlar hasta el último acceso a la información e incluye por defecto protección contra ataques XSS y CSRF.
- Calidad: Su código fuente incluye más de 9.000 pruebas unitarias y funcionales.

- Soporte: sigue una política de tipo LTS (*long term support*), por la que las versiones estables se mantienen durante 3 años sin cambios pero con una continua corrección de errores.
- Licencia: se publica bajo licencia MIT, con la que puedes desarrollar aplicaciones webs comerciales, gratuitas y/o de software libre.
- Documentado: es el framework mejor documentado, ya que ha publicado cinco libros gratuitos de calidad y siempre actualizados. Además, toda la documentación está traducida al español.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo. (16)

Se decide utilizar Symfony v1.4 como framework para el desarrollo del módulo ya que su licencia es de software libre, es uno de los framework PHP más populares entre los usuarios y las empresas, permite que los programadores sean mucho más productivos y a la vez se crea código de más calidad y más fácil de mantener y actualizar. Symfony es maduro, estable, profesional y está muy bien documentado.

ExtJS

Se seleccionó EXTJS v3.1 como framework de diseño visual, pues posee código reutilizable, es un framework de código limpio, que utiliza tecnología AJAX, escrito en Java Script con soporte para la mayoría de los navegadores, presenta cientos de funcionalidades que hacen de él una herramienta indispensable, ya que junto con el ahorro de tiempo y de líneas de código hace que el usuario se sienta a gusto pues presenta una interfaz amigable. Contiene estilos CSS predefinidos aunque es totalmente configurable. Posee 2 licencias, una comercial y otra Open Source.

Ventajas de Extend:

- Código reutilizable.
-
- Orientada a la programación de interfaces tipo desktop en la Web.
- La Interfaz de programación de aplicaciones (API) es homogeneizado independientemente del adaptador usado. Los controles siempre se verán igual.
- Soporte comercial.
- Una extensa comunidad de usuarios. (17)

Lenguaje de programación Web (PHP)

PHP v5.3 es un lenguaje de programación interpretado, usado normalmente para la creación de páginas web dinámicas. Es un lenguaje de programación independiente de la plataforma que se use, rápido, con una gran librería de funciones, una amplia documentación y sobre todo es un lenguaje del lado del servidor gratuito; es decir, se ejecuta en el servidor Web justo antes de que se envíe la página a través de Internet al cliente. Permite la conexión a diferentes tipos de servidores de bases de datos tales como: MySQL, Postgres, Oracle y Microsoft SQL Server; lo cual permite la creación de aplicaciones Web muy robustas. (18)

PHP tiene muchas ventajas, es un lenguaje multiplataforma, capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, capacidad de expandir su potencial al utilizar una 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. Además, es ampliamente usado en la universidad. 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, además PHP está basado en el lenguaje C++ y la sintaxis usada es muy similar a C/C++ el cual es considerado aún el mejor lenguaje de programación por muchos programadores. (19)

JavaScript

JavaScript v2.2 es un lenguaje que no requiere de compilación ya que funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. JavaScript puede crear diferentes efectos e interactuar con los usuarios. Es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros. Además, pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente.

Se decide la utilización de JavaScript como lenguaje del lado del cliente por las siguientes razones:

- Es interpretado (no compilado) por el cliente, es decir, directamente del programa fuente se pasa a la ejecución de dicho programa, por lo que al contrario de los lenguajes compilados no se genera ni código objeto ni ejecutable.
- Está basado en objetos. No es un lenguaje de programación orientada a objetos (OOP). JavaScript no emplea clases ni herencia, ni otras técnicas típicas de la OOP.

- No es necesario declarar los tipos de variables que van a utilizarse ya que realiza una conversión automática de tipos.
- Las referencias a objetos se comprueban en tiempo de ejecución. Esto es consecuencia de que JavaScript no es un lenguaje compilado.
- No puede escribir automáticamente al disco duro. Por eso se dice que JavaScript es un lenguaje seguro para el entorno de Internet en el que se aplicará. (20)

Servidor Web Apache

Es uno de los servidores Web más populares del mercado, y el más utilizado actualmente, de código abierto y gratuito, disponible para Windows y GNU/Linux, entre otros. Apache 2.2 presenta entre otras características, mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Apache tiene amplia aceptación en la red: desde 1996, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 al ser el servidor empleado en el 70% de los sitios Web en el mundo. (21)

Es considerado el servidor web por excelencia, pues desde su surgimiento ha demostrado que es estable y que tiene mejor rendimiento que sus competidores. Además brinda algunas ventajas que son fundamentales para el desarrollo web como son:

- **Fiabilidad:** Más del 90% de los servidores con más alta disponibilidad funcionan bajo un servidor Apache.
- **Software libre:** El servidor Apache es gratuito y es distribuido bajo la licencia de Apache en la cual se permite realizarle cambios al código fuente.
- **Extensibilidad:** Se pueden añadir módulos para ampliar aún más las capacidades del servidor. (22)

Se decide utilizar como servidor Web Apache v2.2, porque fue creado para proveer un alto grado de calidad y fortaleza para las implementaciones que utilizan el protocolo HTTP. Es un software libre y multiplataforma que permite a clientes o instituciones construir sistemas confiables con fines experimentales o para resolver un problema específico de la organización.

Navegador Mozilla Firefox

Es un navegador de Internet, libre, de código abierto y con gran potencia, con interfaz gráfica de usuario, desarrollado por la Corporación Mozilla y un gran número de voluntarios externos. Cualquier usuario

corriente de PC puede usar el navegador sin tener la necesidad de cambiar su sistema operativo. Firefox, es multiplataforma y está disponible en versiones para Microsoft Windows, Mac OS X y GNU/Linux. (23)

Es recomendable utilizar Mozilla Firefox v3.6 para visualizar el sistema, ya que se encuentra disponible para distintas plataformas y utiliza las mismas características en cada una de ellas.

1.4 Herramientas

Enterprise Architect

Enterprise Architect v7.1 es una herramienta flexible, completa y potente de modelado en UML bajo plataforma Windows. Provee lo más nuevo en desarrollo de sistemas, administración de proyectos y análisis de negocio. Abarca integralmente el ciclo de vida de un proyecto software pues cubre el desarrollo desde el levantamiento de los requerimientos a través de las etapas de análisis, modelos de diseño, pruebas y finalmente el mantenimiento y reutilización.

Fue construida en base al excepcional éxito de las versiones previas con un completo soporte para el estándar UML v2.1. Puede generar código fuente C++, Java, C#, VB.Net, Visual Basic, Delphi, PHP, Python y ActionScript. Permite la ingeniería inversa para muchos sistemas que contengan bases de datos Oracle, SQL Server, MySQL, Access, PostgreSQL y otras. (24)

NetBeans

El Entorno Integrado de Desarrollo (IDE) NetBeans v6.9 es una herramienta pensada para escribir, compilar, depurar y ejecutar programas. Consta de una gran comunidad de usuarios en constante crecimiento, lo que le ha permitido, al igual que muchos otros sistemas libres, el progreso paulatino de sus prestaciones y la eliminación de errores que pudiesen existir. El soporte para PHP ha sido extendido para incluir el framework Symfony y PHP v5.3. (25)

NetBeans es gratuito para desarrolladores de software. Ofrece todas las herramientas necesarias para crear aplicaciones Web y de escritorio con el lenguaje Java, C/C ++ y lenguajes dinámicos como PHP y JavaScript. Es fácil de instalar y se puede ejecutar tanto en Windows como en Linux; también tiene detección de errores de sintaxis en tiempo real. NetBeans permite que las aplicaciones se desarrollen a partir de un conjunto de módulos o componentes de software. Brinda una barra de navegación para el acceso rápido a funciones en una clase muy extensa, detecta errores de sintaxis en tiempo real, además de un completamiento de código fuente eficiente y seguro. (26)

Esta herramienta es fácil de instalar y usar será utilizada para el desarrollo del presente módulo, pues es un producto libre y gratuito sin restricciones de uso. Permite crear aplicaciones Web con PHP v5.3 y además incluye soporte para Symfony y AJAX, lo que la hace aun más popular entre los desarrolladores.

1.5 Arquitectura del Sistema

Modelo Vista Controlador (MVC)

El MVC es un patrón de arquitectura de software muy utilizado en la Web, describe la forma de organizar el código de una aplicación al separar los datos, la interfaz de usuario, y la lógica de control en tres componentes distintos:

- **Modelo:** Representa la lógica de negocio, es decir, encapsula los datos y las funcionalidades, es independiente de cualquier representación de salida y comportamiento de entrada.
- **Vista:** Es el objeto que muestra la información al usuario. Puede ser una página web o una parte de ésta. Cada vista tiene asociado un componente controlador. Transforma el modelo en una página web que permite al usuario interactuar con ella e interpretar el código que provee de datos dinámicos a la página.
- **Controlador:** Se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. (27)

Se decide utilizar el patrón MVC por la facilidad que brinda para la creación de sistemas Web, además de la organización que provee en el código fuente por lo que se consigue un mantenimiento más sencillo de las aplicaciones.

Arquitectura Cliente-Servidor

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Como ventajas principales de esta arquitectura se tiene el aumento de la productividad, la disminución de los costes de operación y la mejora en el rendimiento de la red.

Entre las principales características de la arquitectura cliente/servidor, se pueden destacar que:

- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente. (28)

El estudio realizado en el presente capítulo generó una visión de los conceptos fundamentales que rodean a los procesos de gestión de trámites de pasaportes. A partir de este análisis, se inició la búsqueda de distintas aplicaciones que automatizan estos procesos, sin embargo, ninguno de los sistemas estudiados arrojó resultados satisfactorios debido a sus características específicas. Por lo antes planteado surge la necesidad de estudiar algunas tecnologías existentes que propicien el desarrollo de la solución a la problemática planteada. Estos aspectos proveen la base de conocimientos necesaria para llevar a cabo la elaboración del módulo.

Capítulo 2. Características del Sistema

En este capítulo se describe cómo se desarrollan los procesos actuales del negocio, con el objetivo de entender la estructura y la dinámica de la organización en la cual se va a desarrollar el sistema. Se hace una propuesta del sistema para darle solución a la problemática del Departamento de Trámites de la UCCM, para así, identificar los requisitos funcionales y no funcionales de la aplicación, los actores del sistema y mostrar el diagrama de casos de uso del sistema, además de una breve descripción de los mismos.

2.1 Descripción de los Procesos del Negocio

El Departamento de Trámites de la UCCM es el encargado de gestionar todos los procesos relacionados con el trámite de pasaporte de los colaboradores que van a salir a cumplir misión. De manera general se realizan seis procesos: Confeccionar Solicitud de Trámite de Pasaporte, Tramitar Pasaporte, Emitir Libro de Entrada, Entrada de Pasaporte, Salida de Pasaporte e Incinerar Pasaporte, en este capítulo se describen los seis procesos por poseer gran importancia.

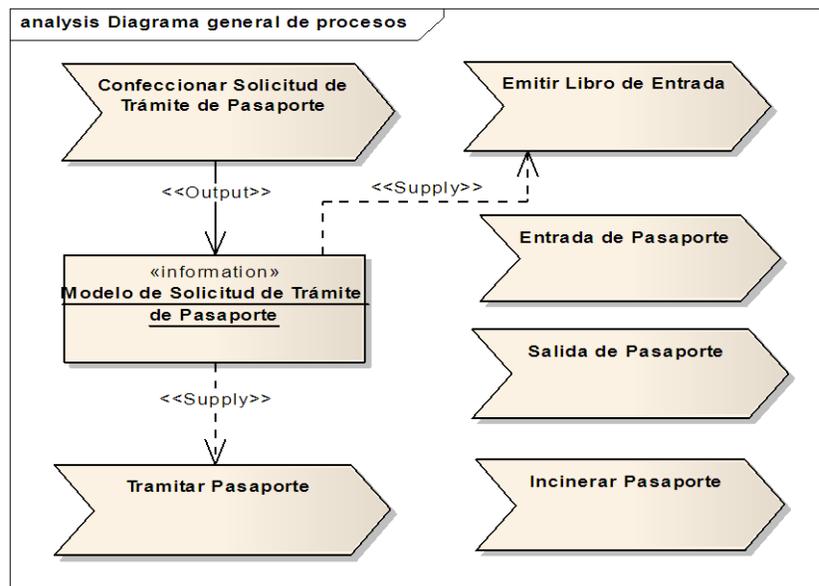


Figura 1. Diagrama General de Procesos del Negocio.

2.1.1 Proceso Confeccionar Solicitud de Trámite de Pasaporte

En el Departamento de Trámites este proceso se genera a partir de la llegada de un expediente del aspirante a colaborador, las diferentes cartas y autorizo del centro de trabajo y la solicitud de pasaporte realizada por este. Una vez que la especialista obtiene esta documentación revisa la calidad y completitud

de la información contenida en la misma y si esta no presenta problemas entonces, inserta los datos precisos en un Excel elaborado por ella y confecciona la Solicitud de Trámite de Pasaporte en la cual se registran los colaboradores, el país de misión, la fecha de salida, el tipo de pasaporte, el tipo de solicitud, entre otros datos.

2.1.2 Proceso Tramitar Pasaporte

El proceso se ejecuta con el objetivo de realizar los trámites necesarios para la confección, rehabilitación ó prórroga de pasaporte. Inicia al recibir la tramitadora 3 copias del modelo de solicitud de trámite de pasaporte y la solicitud de pasaporte con la fotocopia del CI por parte de la especialista. Una vez que la tramitadora recibe esta documentación, entonces pasa a realizar los trámites necesarios para la confección o rehabilitación del pasaporte. En caso de ser una confección, debe elaborar un acta por cada persona, un listado de confecciones, un modelo AO-1 y un permiso de salidas múltiples solo en caso de que el país de misión del colaborador sea para Venezuela, aunque no todo el personal que viaje para ese país debe llevar necesariamente este permiso. En caso de ser una rehabilitación se elabora un acta donde se registran 10 personas, un modelo AO-1 y también un permiso de salidas múltiples solo en caso de que sea necesario.

2.1.3 Proceso Emitir Libro de Entrada

Se efectúa con el objetivo de emitir un libro de entrada con los reportes de los colaboradores que salieron o no de misión. Inicia con la recepción que realiza la especialista de los pasaportes confeccionados, rehabilitados o prorrogados, después de tener estos pasaportes, los entrega a los colaboradores para que salgan a cumplir su misión. Por último, realiza un reporte con los datos de los colaboradores que se presentaron a recoger su pasaporte para salir de misión y también los que por algún motivo no se presentaron.

2.1.4 Proceso Entrada de Pasaporte

Se realiza con el objetivo de registrar la entrada de los pasaportes al archivo pasivo. Se inicia al recibir la funcionaria de archivo un listado de colaboradores pasivos por parte de la especialista, es decir, colaboradores que concluyeron su misión, con sus respectivos pasaportes. Luego de recibir esta información registra el nombre del especialista que lo entregó, la firma y la fecha de entrega. Por último, inserta los datos del pasaporte en la base de datos, registrándose (nombre, apellido, CI, número de pasaporte, destino, fecha de confección, fecha de vencimiento, fecha de fin de misión, entre otros datos).

2.1.5 Proceso Salida de Pasaporte

Se genera con el objetivo de registrar la salida de los pasaportes del archivo. Es iniciado al recibir la funcionaria de archivo un listado de colaboradores en estado activo, es decir, colaboradores a los cuales les fue asignada una misión, luego de obtener el listado registra la salida de los pasaportes en la base de datos y se especifica además, a quién es entregado el pasaporte y la especialidad de salida del colaborador en la nueva misión asignada. Por último, hace entrega de los pasaportes al especialista.

2.1.6 Proceso Incinerar Pasaporte

El proceso tiene como objetivo realizar la incineración de los pasaportes en el archivo. En este caso la funcionaria de archivo busca en la base de datos los pasaportes cuya fecha de vencimiento se haya cumplido, a continuación elabora un listado de los pasaportes vencidos y los manda a incinerar, inmediatamente cambia el estado del pasaporte ha vencido y registra en la base de datos el nombre, los apellidos y el número de los pasaportes incinerados, esta operación es realizada anualmente.

2.2 Modelo del Negocio

Al interactuar con los trabajadores de la entidad que abarcó el alcance de las entrevistas realizadas, se obtuvo como resultado que existían diferentes procedimientos para la manipulación de la información y en la realización de las actividades. Debido a esto se decidió realizar la modelación utilizándose descripciones de procesos de negocio, lo que permitiría obtener modelos mucho más exquisitos en cuanto a información, y proporcionaría al sistema a diseñar, una mayor capacidad de adaptabilidad y mejores niveles de personalización, para lograr así una solución flexible ante las exigencias de los usuarios.

2.2.1 Descripción de Procesos del Negocio

2.2.1.1 Diagrama de Procesos P1 Confeccionar Solicitud de Trámite de Pasaporte

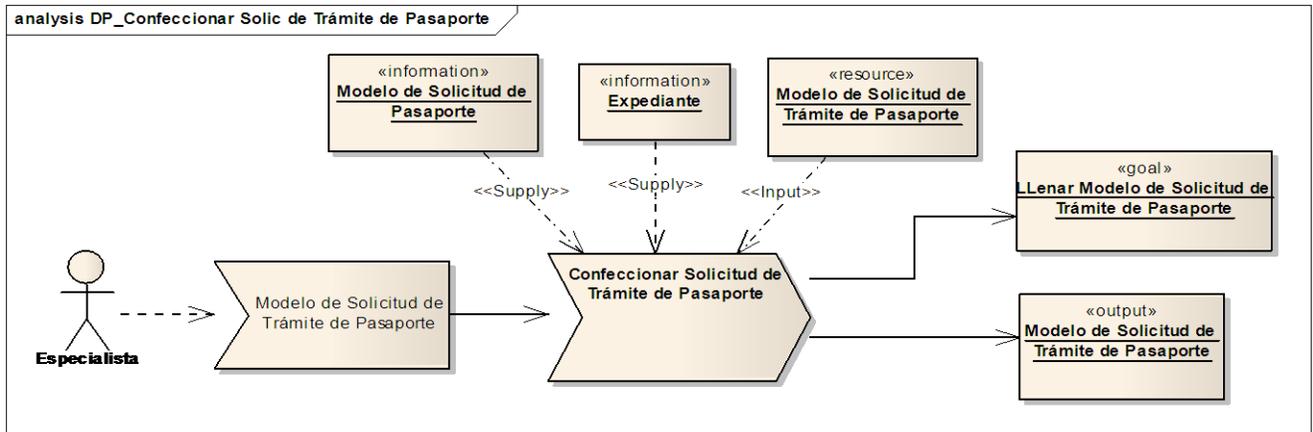


Figura 2: Diagrama de Procesos P1 Confeccionar Solicitud de Trámite de Pasaporte.

2.2.1.2 Diagrama de Procesos P2: Tramitar Pasaporte

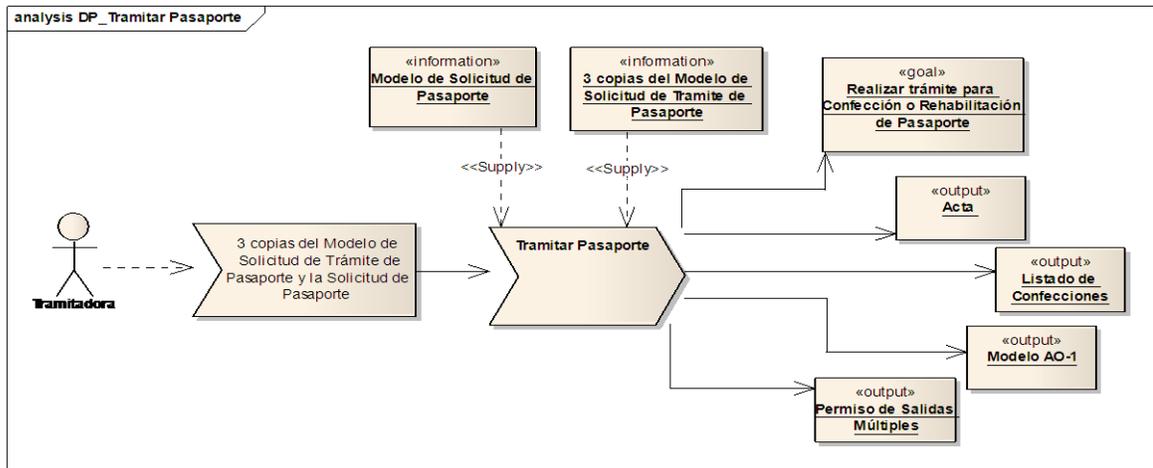


Figura 3: Diagrama de Procesos P2 Tramitar Pasaporte.

2.2.1.3 Diagrama de Procesos P3: Emitir Libro de Entrada

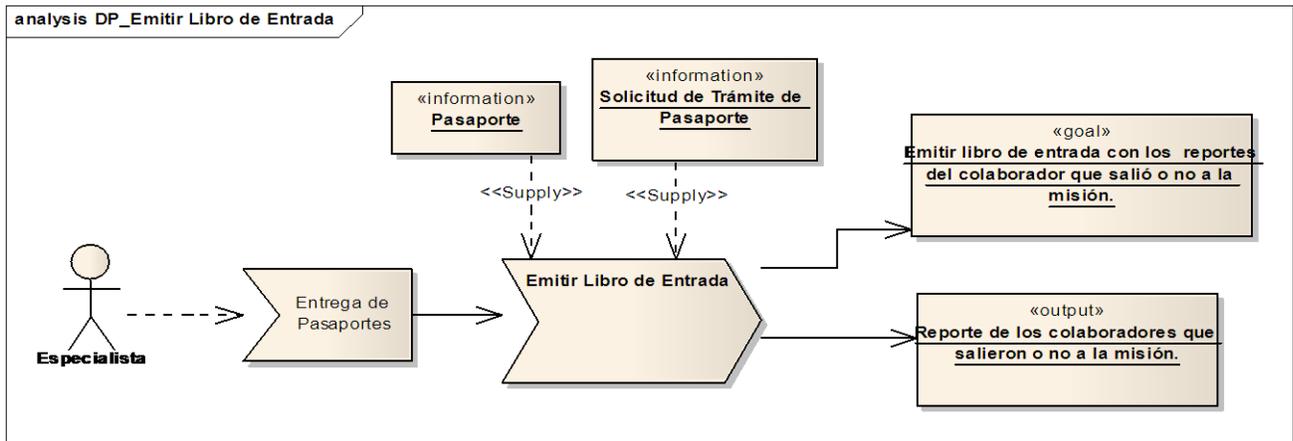


Figura 4: Diagrama de Procesos P3 Emitir Libro de Entrada.

2.2.1.4 Diagrama de Procesos P4: Entrada de Pasaporte

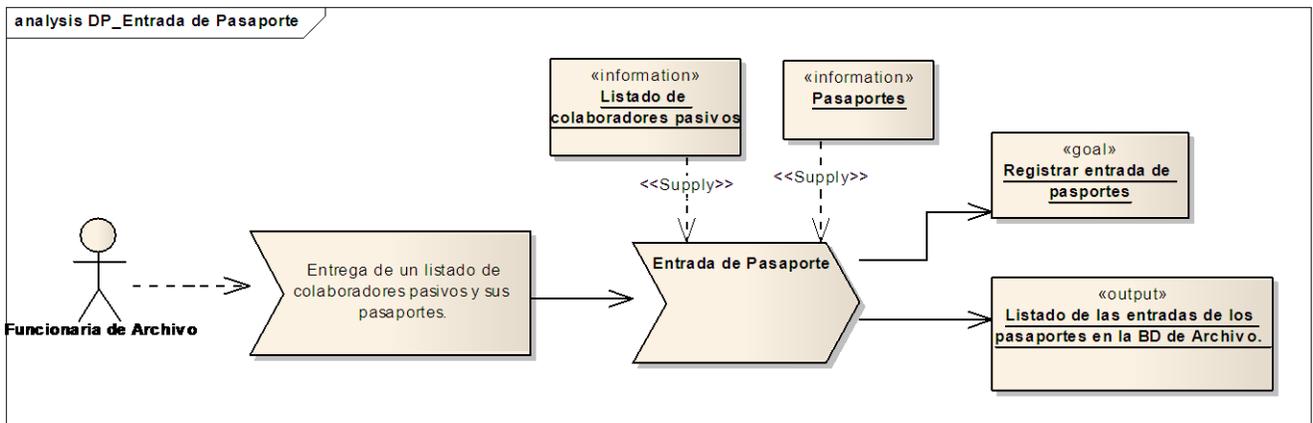


Figura 5: Diagrama de Procesos P4 Entrada de Pasaporte.

2.2.1.5 Diagrama de Procesos P5: Salida de Pasaporte

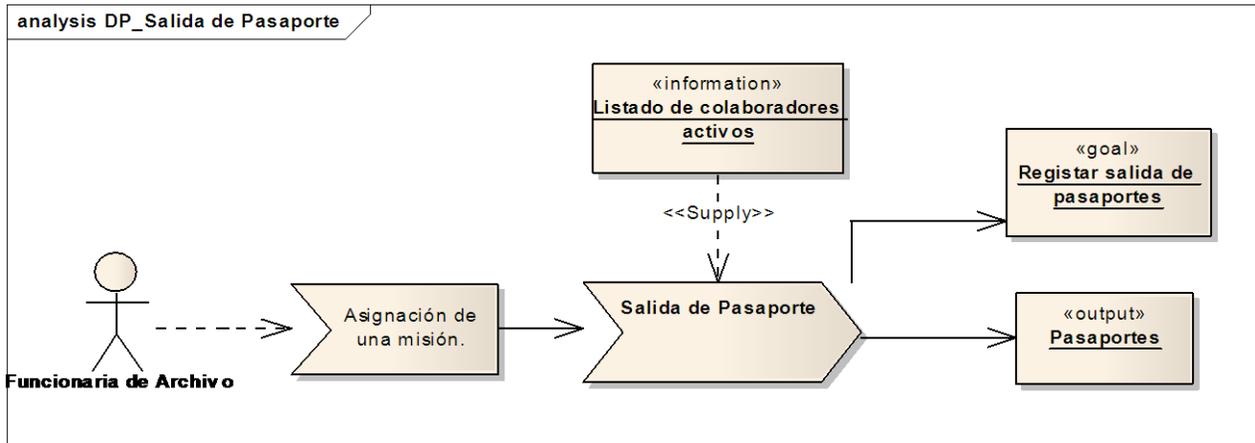


Figura 6: Diagrama de Procesos P5 Salida de Pasaporte.

2.2.1.6 Diagrama de Procesos P6: Incinerar Pasaporte

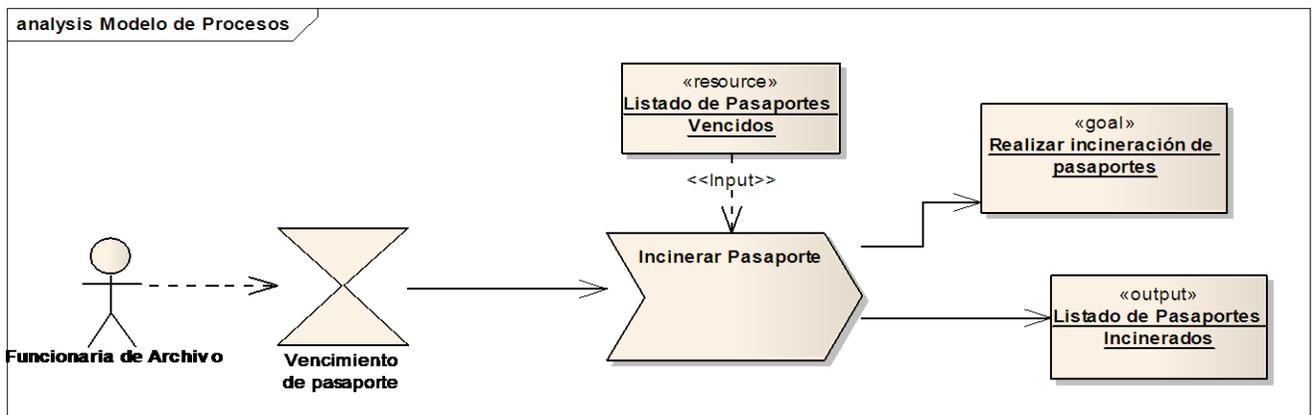


Figura 7: Diagrama de Procesos P6 Incinerar Pasaporte.

2.3 Especificación de Requisitos de Software

Con el propósito de obtener un mejor entendimiento en el desarrollo, se hace la especificación y descripción de los requisitos del software, al ser uno de sus objetivos lograr una negociación por parte del cliente y los desarrolladores, para así poder determinar lo que debe hacer el sistema.

2.3.1 Requerimientos Funcionales

Mediante la elaboración de los procesos del negocio se pueden detectar las posibles actividades a automatizar, que son una propuesta de las futuras funcionalidades del sistema y constituyen la base para la identificación de los requisitos funcionales de la aplicación. Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. A continuación se listan los requisitos identificados:

Listado de Requerimientos Funcionales	
RF1 Autenticar	RF26 Registrar Pasaportes Incinerados
RF2 Crear Modelo de Solicitud de Trámite de Pasaporte	RF27 Rehabilitar Pasaporte
RF3 Buscar Modelo de Solicitud de Trámite de Pasaporte	RF28 Prorrogar Pasaporte
RF4 Listar Modelo de Solicitud de Trámite de Pasaporte	RF29 Adicionar Colaborador
RF5 Anular Modelo de Solicitud de Trámite de Pasaporte	RF30 Buscar Colaborador
RF6 Generar PDF Modelo de Solicitud de Trámite de Pasaporte	RF31 Listar Colaborador
RF7 Generar PDF Listado de Confecciones	RF32 Anular Colaborador
RF8 Crear Acta.	RF33 Eliminar Colaborador de la solicitud
RF9 Buscar Acta	RF34 Generar Reporte de Confecciones
RF10 Listar Acta	RF35 Generar Reporte de Rehabilitaciones
RF11 Generar PDF Acta.	RF36 Generar Reporte de Prórrogas
RF12 Crear Modelo AO-1	RF37 Generar Reporte de Salidas Múltiples
RF13 Buscar Modelo AO-1	RF38 Generar Reporte Histórico del Colaborador
RF14 Listar Modelo AO-1	RF39 Generar Reporte de Pasaportes
RF15 Generar PDF Modelo AO-1	RF40 Buscar Información de Confecciones
RF16 Crear Permiso de Salida Múltiple	RF41 Buscar Información de Rehabilitaciones
RF17 Buscar Permiso de Salida Múltiple	RF42 Buscar Información de Prórrogas
RF18 Listar Permiso de Salida Múltiple	RF43 Buscar Información de Salidas Múltiples
RF19 Generar PDF Permiso de Salida Múltiple	RF44 Buscar Información Histórico de un Colaborador
RF20 Registrar Pasaporte	RF45 Buscar Información de Pasaportes
RF21 Buscar Pasaporte	RF46 Generar PDF Reporte de Confecciones
RF22 Listar Pasaporte	RF47 Generar PDF Reporte de Rehabilitaciones
RF23 Anular Pasaporte	RF48 Generar PDF Reporte de Prórrogas
RF24 Registrar Entrada de Pasaporte	RF49 Generar PDF Reporte de Salidas Múltiples
RF25 Registrar Salida de Pasaporte	RF50 Generar PDF Reporte Histórico del Colaborador
	RF51 Generar PDF Reporte de Pasaportes

Tabla 1. Requisitos Funcionales.

2.3.2 Diagrama de Paquetes de Requisitos Funcionales

El Diagrama de Paquetes de Requerimientos Funcionales representa una vista global de los requerimientos funcionales agrupados por paquetes según la funcionalidad que realizan. En cada paquete se encuentran los requisitos funcionales relacionados entre sí. A continuación se muestra dicho diagrama y una representación de la estructura de los paquetes que representan las funcionalidades más importantes del sistema; el resto se puede encontrar en el Expediente del Proyecto.

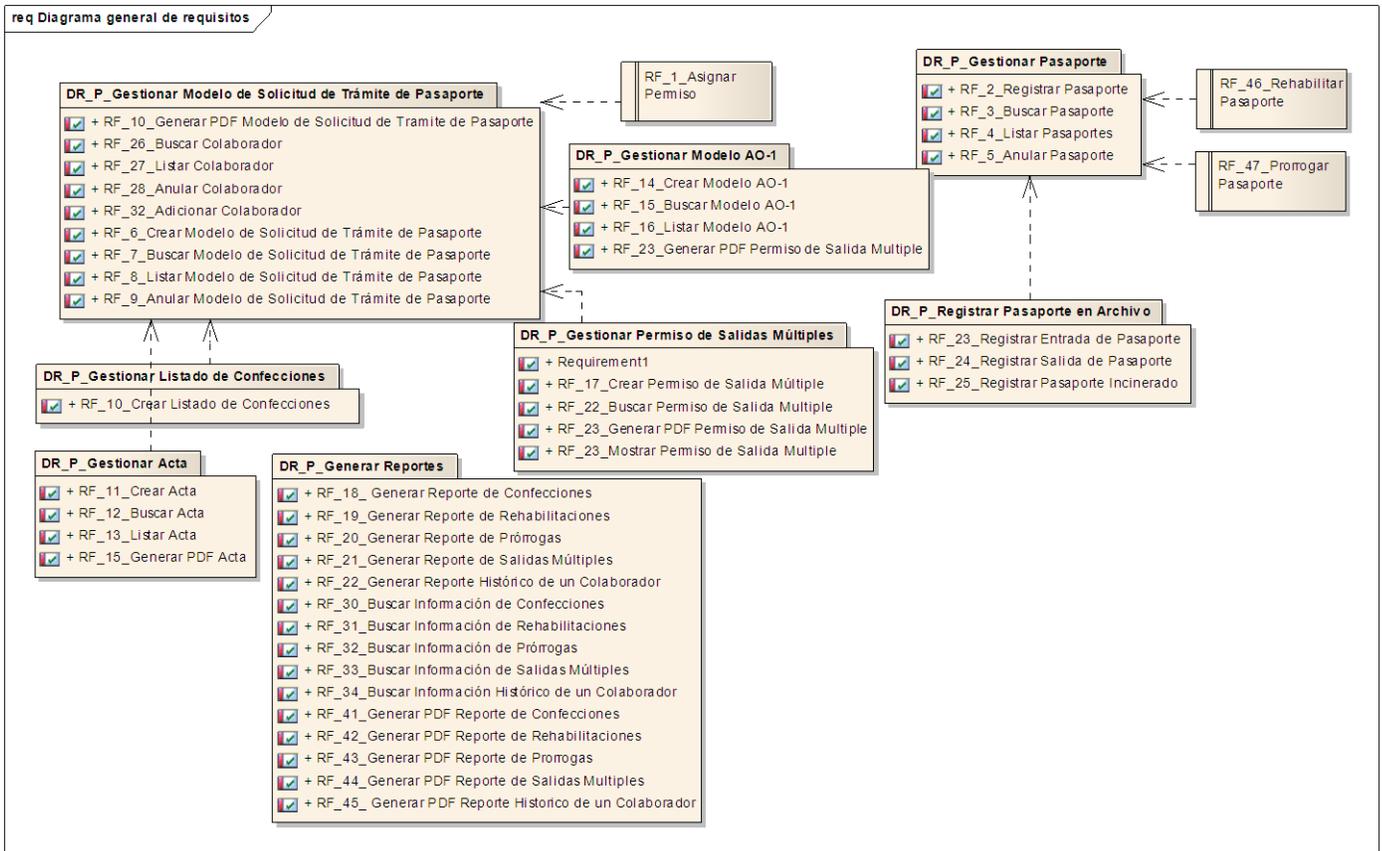


Figura 8: Diagrama de Paquetes de Requisitos Funcionales.

2.3.3 Requerimientos No Funcionales

Los requisitos 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 y confiable. Los requerimientos no funcionales son fundamentales en el éxito del producto y normalmente están vinculados a requisitos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener y cuán rápido o grande debe ser.

Existen múltiples categorías para clasificar a los requisitos no funcionales, a continuación se muestran los más destacados en este sistema:

Usabilidad

RNF 1 El sistema solo podrá ser utilizado por los usuarios creados por la administración. Cada usuario que se autentique en el sistema solo tendrá acceso a la información que le corresponde según su nivel y tipo de usuario.

RNF 2 El sistema debe garantizar un acceso fácil y rápido, podrá ser usado por usuarios con pocos conocimientos informáticos.

Fiabilidad

RNF3 Una vez terminado el Sistema para la Cooperación Médica se realizarán procesos de despliegue, capacitación y mantenimiento de software. El personal que trabaja con el software debe contar con el nivel técnico requerido mediante adiestramiento de servicio.

Confiabilidad: La información manejada por el sistema está protegida de acceso no autorizado.

Integridad: La información no podrá ser divulgada .Se harán copias de respaldo que puedan restaurar el sistema en caso de pérdida de información.

Disponibilidad: Solo los usuarios autorizados tendrán acceso a la información en todo momento.

Eficiencia

RNF4 El sistema deberá responder rápido ante las solicitudes de los usuarios, por lo que el procesamiento de la información y el tiempo de respuesta deberán ser en el menor tiempo posible.

La eficiencia de la aplicación estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor, y la velocidad de las consultas a la base de datos. Se realizará la validación de los datos en el cliente y en el servidor, aquellas que por cuestiones de seguridad o de acceso a los datos lo requieran, lográndose así un tiempo de respuesta más rápido, con mayor velocidad de procesamiento y aprovechamiento de los recursos.

Soporte

RNF5 Se debe acceder al sistema desde cualquier plataforma (Linux o Windows).

RNF6 Se debe garantizar que el sistema sea compatible con el resto de los módulos que se realizan.

RNF7 El sistema debe estar bien documentado de forma tal que el tiempo de mantenimiento sea mínimo en caso de necesitarse.

Restricciones de diseño

RNF 8 El sistema informático estará desarrollado sobre una plataforma Web y podrá ser utilizado desde cualquier sistema operativo, recomendándose para su uso Linux.

RNF 9 Se definen además, estándares de diseño y codificación en el Departamento Sistemas de Apoyo a la Salud por los cuales debe regirse tanto la implementación como los modelos generados, como parte del proceso de desarrollo.

Requisitos para la documentación de usuarios en línea y ayuda del sistema

RNF 10 La aplicación estará documentada con una ayuda dirigida a los diferentes tipos de usuarios con que cuenta el sistema para garantizar la utilización del sistema.

RNF 11 Se dispondrá de un Manual de Usuario que indicará como interactuar con las funcionalidades del sistema.

Interfaz

RNF12 La misma será sencilla, amigable e intuitiva, de fácil navegación por parte del usuario. Estará diseñada para una óptima visualización al ser adaptable a cualquier resolución.

➤ Interfaces de usuario

RNF13 Debe tener claridad y buena organización de la información, además de permitir la interpretación correcta e inequívoca de la misma.

RNF14 Debe permitir la ejecución de acciones de manera rápida.

RNF15 Se debe hacer uso de elementos visuales para seleccionar la información, y minimizar los posibles errores.

RNF16 Debe enviar los avisos y mensajes al usuario.

RNF17 Los errores serán visibles al usuario informándose la ocurrencia de los mismos.

➤ Interfaces Hardware

Para servidor de aplicación y servidor de base de datos.

- **Servidor de aplicación:** Computadora en la que se instala el software servidor web, que se encargará de atender las peticiones de los usuarios del sistema.

RNF18 Debe tener los siguientes requerimientos de hardware:

Tipo de procesador: Intel Pentium IV o superior.

Velocidad del procesador: 3.00 GHz o superior dual core.

Memoria RAM: 2 GB o superior.

Disco Duro: 80 GB o superior.

Se requiere tarjeta de red.

- **Servidor de base de datos:** Computadora en la que se instala el software gestor de bases de datos.

RNF19 Debe tener los siguientes requerimientos de hardware:

Tipo de procesador: Intel Pentium IV o superior.

Velocidad del procesador: 3.00 GHz o superior dual core.

Memoria RAM: 2 GB o superior.

Disco Duro: 160 GB o superior.

Se requiere tarjeta de red.

- **Para estaciones de trabajo**

RNF20 Los ordenadores que serán utilizados por los usuarios del sistema para acceder a la aplicación y operar la misma deben tener los siguientes requerimientos de hardware:

Tipo de procesador: Intel Pentium III o superior.

Velocidad del procesador: 2.0 GHz o superior.

Memoria RAM: 512 MB o superior.

Disco Duro: 40 GB.

Se requiere tarjeta de red.

- **Interfaces Software**

RNF21 Los softwares que serán utilizados en el desarrollo y despliegue de la aplicación serán:

El servidor de aplicaciones debe llevar como sistema operativo Ubuntu Server 10.04.

Sistema gestor de base de datos PostgreSQL 8.3.

Servidor de procesos el Supervisor 3.0.

Servidor de caché Memcached 1.4.

Servidor web Apache 2.2.

Lenguaje de programación PHP 5.3.

Flash Player 10.0 o superior para la visualización de las gráficas.

Interfaces de Comunicación

RNF22 Se debe garantizar por parte de la entidad con una infraestructura de comunicación entre el cliente y el servidor.

Estándares Aplicables

RNF24 Para las descripciones de casos de uso, mensajes y avisos que debe emitir el sistema, se deben seguir las pautas de análisis definidas en el Centro de Informática Médica.

RNF25 Para la implementación del sistema se deberá seguir los estándares de codificación y diseño definidos por el CESIM, fundamentalmente en el caso de los mensajes emitidos por la aplicación.

2.4 Definición de los Casos de Uso del Sistema

2.4.1 Descripción de los Actores del Sistema

Actor	Descripción
Especialista	Es el encargado de confeccionar el modelo Solicitud de Trámite de Pasaporte, donde se registran una serie de colaboradores y el tipo de trámite que les serán realizados a los mismos.
Tramitadora	Es la encargada de realizar las actas y modelos necesarios para realizar cualquier tipo de trámite de pasaporte.
Funcionaria de Archivo	Es el encargado de registrar las Entradas, Salidas e Incineración de los pasaportes en el Archivo Pasivo.

Tabla 2. Descripción de actores del sistema.

2.4.2 Vista Global de los Actores del Sistema

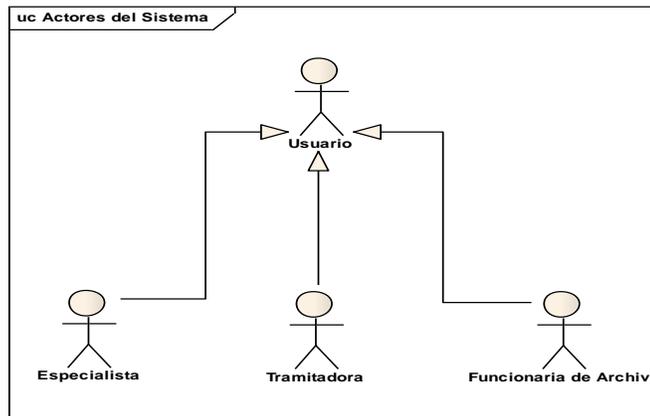


Figura 9: Diagrama de Actores del Sistema.

2.4.3 Lista de Casos de Uso

Casos de Uso	
CU_Autenticar	CU_Crear Permiso de Salidas Múltiples
CU_Anular Solicitud de Trámite de Pasaporte	CU_Generar Reporte de Confecciones
CU_Anular Pasaporte	CU_Generar Reporte de Rehabilitaciones
CU_Anular Colaborador	CU_Generar Reporte de Pasaportes
CU_Buscar Solicitud de Trámite de Pasaporte	CU_Generar Reporte de Prórrogas
CU_Buscar Colaborador	CU_Generar Reporte de Salidas Múltiples
CU_Buscar Pasaporte	CU_Generar Reporte Histórico de un Colaborador
CU_Rehabilitar Pasaporte	CU_Registrar Pasaporte
CU_Prorrogar Pasaporte	CU_Registrar Entrada de Pasaporte
CU_Crear Solicitud de Trámite de Pasaporte	CU_Registrar Salida de Pasaporte
CU_Crear Acta	CU_Registrar Pasaportes Incinerados
CU_Crear Modelo AO-1	CU_Eliminar Colaborador de la solicitud
CU_Generar Listado de Confecciones	

Tabla 3. Lista de Casos de Uso.

2.4.4 Diagrama de Casos de Uso del Sistema

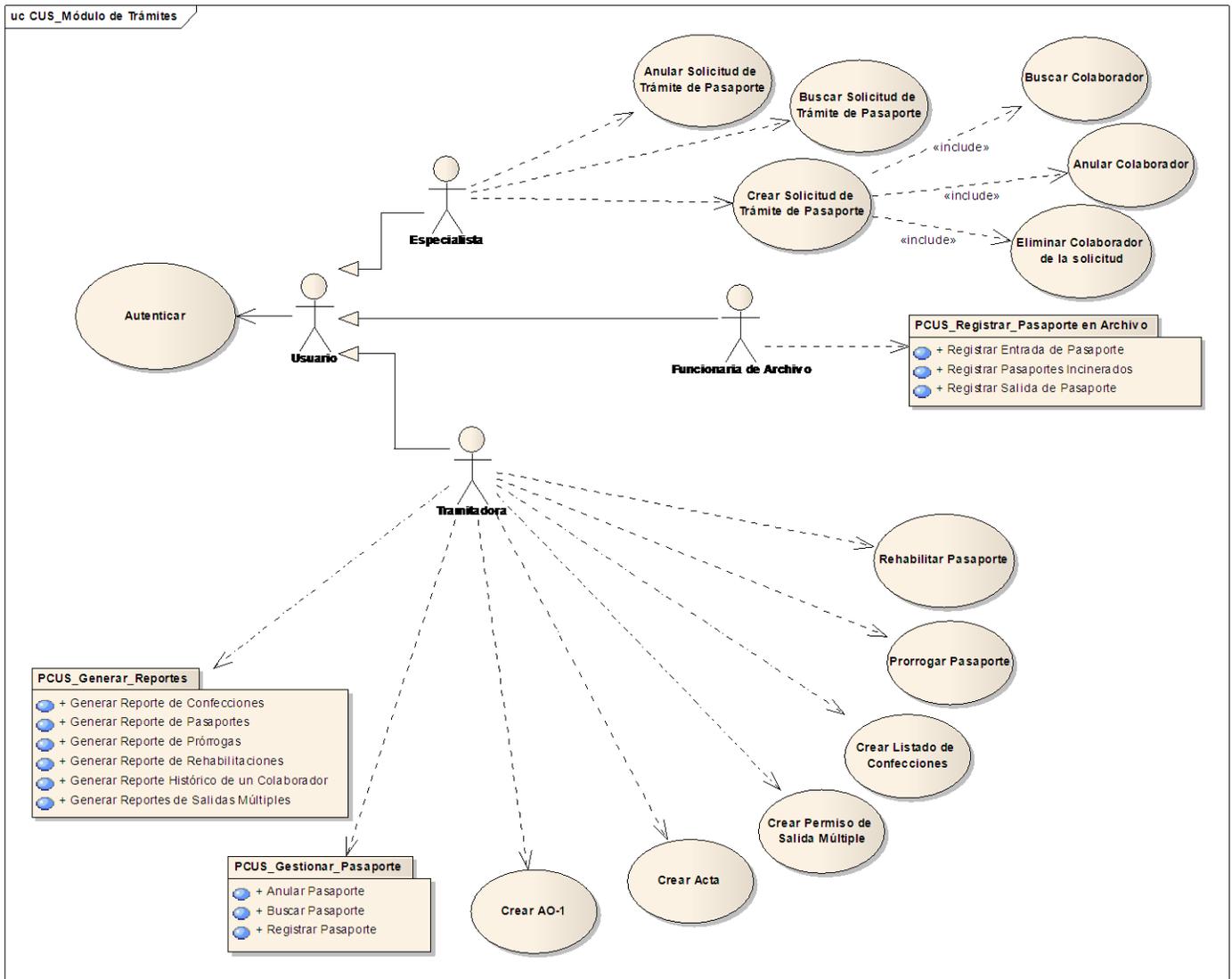


Figura 10: Diagrama de Casos de Uso del Sistema.

2.4.5 Descripción Textual de los Casos de Uso del Sistema

En la descripción textual se reflejan los casos de uso y la interacción de éstos con los actores. En el sistema se conformaron veinticinco casos de uso, de los cuales se realiza la descripción de los que son arquitectónicamente significativos para la aplicación. Los restantes casos de uso se pueden encontrar como parte del Expediente de Proyecto.

2.4.5.1 CU_Crear Solicitud de Trámite de Pasaporte

Caso de Uso:	Crear Solicitud de Trámite de Pasaporte.
Actores:	Especialista
Resumen:	El caso de uso inicia al acceder el actor a la opción Nueva, el sistema brinda la posibilidad de introducir y/o seleccionar los datos para crear la Solicitud, el actor introduce los datos de la Solicitud, el sistema crea la Solicitud de Trámite de Pasaporte, el caso de uso termina.
Precondiciones:	El usuario debe tener permisos asignados en el sistema.
Referencias	RF6, RF32, RF33, RF34
Prioridad	Crítico
Poscondiciones	La Solicitud de Trámite de Pasaporte ha sido creada.

Tabla 5. Descripción del CU_Crear Solicitud de Trámite de Pasaporte.

2.4.5.2 CU_Generar Listado de Confecciones

Caso de Uso:	Generar Listado de Confecciones.
Actores:	Tramitadora
Resumen:	El caso de uso inicia al acceder el actor a la opción Generar Listado de Confecciones, el sistema brinda la posibilidad de seleccionar la opción generar Listado de Confecciones, el sistema genera el Listado de Confecciones, el caso de uso termina.
Precondiciones:	Debe estar creada la Solicitud de Trámite de Pasaporte.
Referencias	RF11, RF7, RF8
Prioridad	Crítico
Poscondiciones	El Listado de Confecciones ha sido generado.

Tabla 6. Descripción del CU_Generar Listado de Confecciones.

2.4.5.3 CU_ Crear Acta

Caso de Uso:	Crear Acta
Actores:	Tramitadora
Resumen:	El caso de uso inicia al acceder el actor a la opción Crear Acta, el sistema brinda la posibilidad de introducir y/o seleccionar los datos para crear el Acta, el actor introduce los datos del Acta, el sistema crea el Acta, el caso de uso termina.
Precondiciones:	Debe estar creada la Solicitud de Trámite de Pasaporte.
Referencias	RF12, RF7, RF8, RF34
Prioridad	Crítico.
Poscondiciones	El Acta ha sido creada.

Tabla 7. Descripción del CU_ Crear Acta.

2.4.5.4 CU_ Crear Modelo AO-1

Caso de Uso:	Crear Modelo AO-1.
Actores:	Tramitadora
Resumen:	El caso de uso inicia al acceder el actor a la opción Crear AO-1, el sistema brinda la posibilidad de introducir y/o seleccionar los datos para crear el AO-1, el actor introduce los datos del AO-1, el sistema crea el AO-1, el caso de uso termina.
Precondiciones:	Debe estar creada la Solicitud de Trámite de Pasaporte.
Referencias	RF16, RF7, RF8, RF34
Prioridad	Crítico
Poscondiciones	El Modelo AO-1 ha sido creado.

Tabla 8. Descripción del CU_ Crear Modelo AO-1.

2.4.5.5 CU_ Registrar Entrada de Pasaporte

Caso de Uso:	Registrar Entrada de Pasaporte
Actores:	Funcionaria de Archivo
Resumen:	El caso de uso inicia al acceder el actor a la opción Entrada de Pasaporte, el sistema brinda la posibilidad de seleccionar la opción registrar Entrada, el

	actor selecciona la opción, el sistema registra la Entrada, el caso de uso termina.
Precondiciones:	Debe existir el pasaporte en la base de datos.
Referencias	RF29, RF3, RF4
Prioridad	Crítico
Poscondiciones	La Entrada del Pasaporte ha sido registrada.

Tabla 9. Descripción del CU_Registrar Entrada de Pasaporte.

En este capítulo, a partir de la descripción de la solución propuesta, se detallaron los principales procesos, las consecuencias que trae el funcionamiento poco eficiente de los mismos y las posibles mejoras a realizar mediante la detección de las actividades a automatizar. Además se identificaron los requisitos que debe cumplir el sistema para satisfacer las necesidades del cliente, y así conformar las funcionalidades del sistema mediante la obtención de los casos de uso, que ayudarán, de conjunto con los requisitos no funcionales a definir la arquitectura, diseñar e implementar la aplicación.

Capítulo 3. Análisis y Diseño del Sistema

En el presente capítulo se profundiza en el análisis y diseño del sistema, el cual tiene como objetivos fundamentales transformar los requerimientos en un diseño que muestre cómo va a ser implementado el sistema, evolucionar hacia una arquitectura del software robusta y flexible ante la aplicación de nuevos cambios, y realizar la propuesta de interfaz de usuario del software adaptando al diseño para que coincida con el ambiente de implementación, y diseñar el sistema con un enfoque hacia el rendimiento y la eficiencia. Para lograr lo anteriormente expuesto se realizan artefactos como: los Diagramas de Clases de Diseño, Diagramas de Interacción y de Clases del Diseño, y la Descripción de la Arquitectura del Sistema.

3.1 Descripción de la Arquitectura

El buen funcionamiento de un software está determinado en gran medida por la organización y estructuración de los componentes del mismo y de las relaciones entre ellos. Realizar un sistema con un diseño y un orden establecido es factible para el desarrollo en sí y para alcanzar un grado de calidad. Es en este punto donde la elección de una arquitectura adecuada para el software que está realizándose, se convierte en una tarea sumamente importante.

La arquitectura de software es el diseño de más alto nivel de la estructura de un sistema. Permite definir los componentes que llevan a cabo alguna función y como se relacionan entre ellos, diseñar un sistema en base a los objetivos y las restricciones, teniéndose en cuenta el contexto en el que se establecerá, y los principios que orientan su diseño y evolución. Los objetivos son aquellos fijados para el sistema de información, y se tienen en cuenta no solo a los de tipo funcional sino también a los no funcionales, como son la flexibilidad e interacción con otros sistemas, entre otros. Las restricciones son aquellas limitaciones impuestas por la tecnología existente para implementar el sistema.

El desarrollo de la aplicación se basa en la utilización del Framework Symfony, el cual implementa el patrón Modelo Vista Controlador (MVC). Los proyectos desarrollados con este Framework, basan su arquitectura en dicho patrón. Brindándole al sistema una clara definición entre cómo se muestra la información, el manejo de las acciones que el usuario desea hacer sobre el sistema y cómo se realizan estas acciones al modificar y validar la información. También utiliza otro patrón de diseño de gran importancia, como es:

- El Decorator: Es la unión del archivo llamado layout.php con las plantillas, donde el primero almacena el código HTML que es común para todas las páginas de la aplicación, y el segundo contiene los formularios y datos específicos de cada interfaz. El resultado de la aplicación de este

patrón de diseño puede estar integrado también por parciales, slots, componentes y slots de componentes en caso de existir.

Al analizar las características propias de Symfony, la aplicación cuenta en el Modelo, la Vista y el Controlador, con los elementos propios del framework.

La capa del modelo se divide en la Capa de Acceso a los Datos y en la Capa de Abstracción de la Base de Datos. La primera contiene todas las funcionalidades que responden a la lógica del negocio y la segunda provee una abstracción de la base de datos que posibilita que la aplicación no dependa de un sistema gestor determinado. De esta forma, las funciones que acceden a los datos no utilizan sentencias ni consultas que dependen de una base de datos, sino que utilizan otras funciones para realizar las consultas. Así, si se cambia de sistema gestor de base de datos, solamente es necesario actualizar la capa de abstracción de la base de datos, pues el acceso a datos se realiza mediante el uso del mapeo de objetos relacional (ORM) Doctrine, que abstrae al sistema del uso de cualquier sistema gestor de base de datos, mediante el uso de la programación orientada a objeto, y hacer posible tratar las tablas como objetos del sistema.

En la Vista también se puede aprovechar la separación del código, por lo que Symfony la separa en capa externa (layouts), plantillas (templates) y lógica de la vista, al agrupar en las primeras la parte de esta, que permanece invariable para todas o parte de las páginas de la aplicación, mientras que las segundas solo se encargan de visualizar las variables definidas en el controlador. En los archivos de la lógica se definen elementos de la configuración de la Vista. En esta capa se utiliza además, la librería ExtJS.

En las aplicaciones Web, el controlador, generalmente se encuentra muy cargado, debido a que es el encargado de las tareas comunes como el manejo de las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación y otras similares.

Por tal motivo, el controlador en Symfony normalmente se divide en un Controlador Frontal, que constituye el único punto de entrada a la aplicación y cuya principal función es re-direccionar las peticiones realizadas por la Vista, hacia las Acciones, que constituyen los controladores de los diferentes módulos que conforman la aplicación. Así, en caso de que sea necesario impedir el acceso a la aplicación, solamente es preciso editar el script correspondiente al controlador frontal. Si la aplicación no dispone de un controlador frontal, se debe modificar cada uno de los controladores.

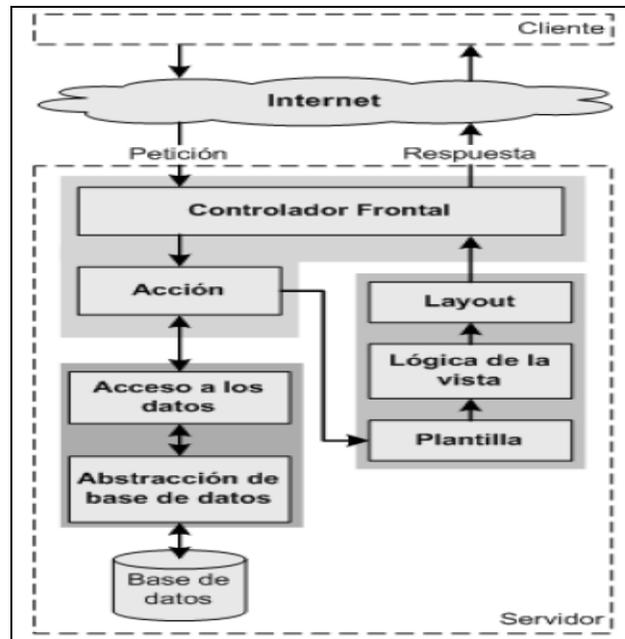


Figura 11: Implementación del patrón MVC según el Framework Symfony.

De esta forma el usuario al interactúa con la aplicación lo hace mediante la Vista, que envía las peticiones al Controlador Frontal, que las re-direcciona hacia la Acción correspondiente, encargándose esta de llamar a la función implementada en el Modelo, donde se obtiene la respuesta, que es enviada a través del Controlador hacia la Vista.

Otros patrones utilizados fueron los llamados Patrones Generales de Software para Asignar Responsabilidades (GRASP, por sus siglas en inglés), que tuvieron una importante utilidad en el diseño realizado. A cada clase le fueron asignadas las tareas que podían realizar según la información que poseían, además de crear las instancias de otras clases en correspondencia con la responsabilidad dada, poniéndose de manifiesto los patrones Experto y Creador.

De esta manera, los objetos logran valerse de la información que contienen para realizar las tareas que se les pide, además de utilizar el patrón Creador para definir quién será el responsable de crear una nueva instancia de una clase. Este diseño obtenido cumple con los patrones de Bajo Acoplamiento y de Alta Cohesión lo que facilita la centralización de actividades de cada elemento, y logra que estos realicen una única tarea dentro del sistema. (29)

3.2 Modelo de Análisis

En la construcción del Modelo de Análisis se identifican las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el Diagrama de Clases del Análisis, que por lo general se descompone para agrupar las clases en paquetes. Esta descomposición tiene impacto en el diseño e implementación de la solución.

Los Diagramas de Clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones. Estos diagramas son el pilar básico del modelado con UML, pues son utilizados tanto para mostrar lo que el sistema puede hacer en el análisis, como para mostrar cómo puede ser construido en el diseño.

3.2.1 Clases del Análisis

Las clases del análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos. Las clases del análisis están siempre identificadas con uno de los tres estereotipos existentes, los cuales son:

- Interfaz: Se encargan de la modelación de toda la interacción que puede existir entre los actores y el sistema; constituyen las fronteras del mismo.
- Control: Representan la coordinación, secuenciación, transacciones y a veces la lógica del negocio; se emplean a menudo para encapsular el control referido a un caso de uso.
- Entidad: representa la información de larga duración y a menudo persistente que se maneja en el sistema. (30)

3.2.2 Diagrama de Clases del Análisis

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa el funcionamiento del mundo real, no de la implementación automatizada del mismo. (31)

A continuación se muestran los diagramas de clases del análisis de algunos casos de uso del sistema. Los diagramas restantes se encuentran en el Expediente de Proyecto.

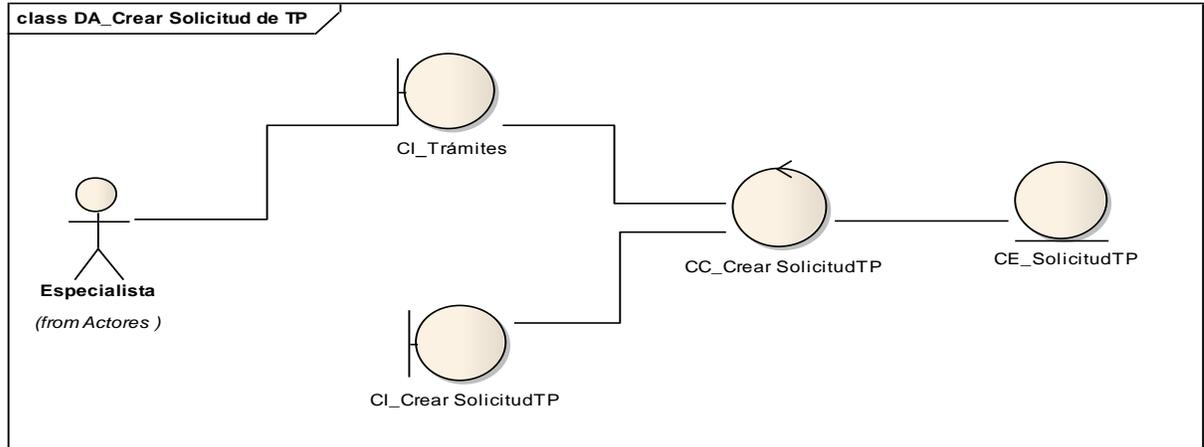


Figura 12. Diagrama de Clases de Análisis. CU_Crear Solicitud de TP.

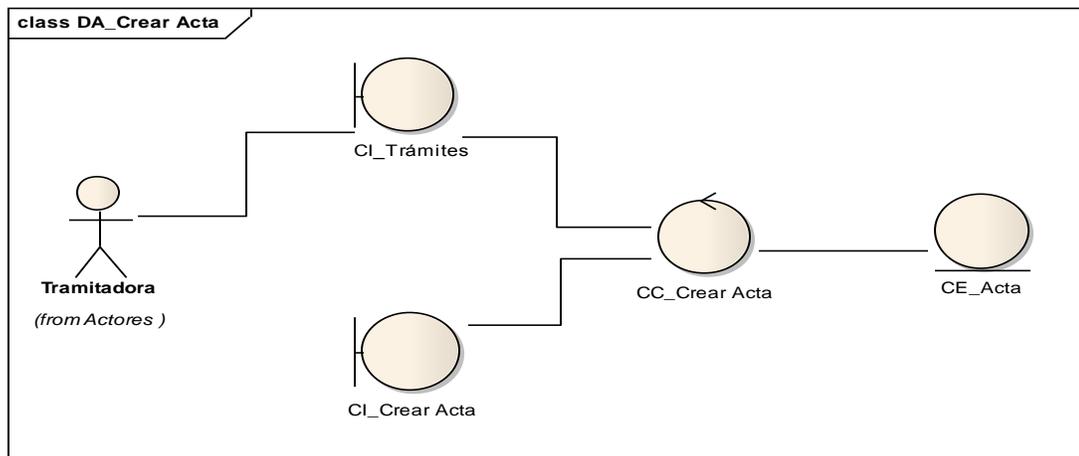


Figura 13. Diagrama de Clases de Análisis. CU_Crear Acta.

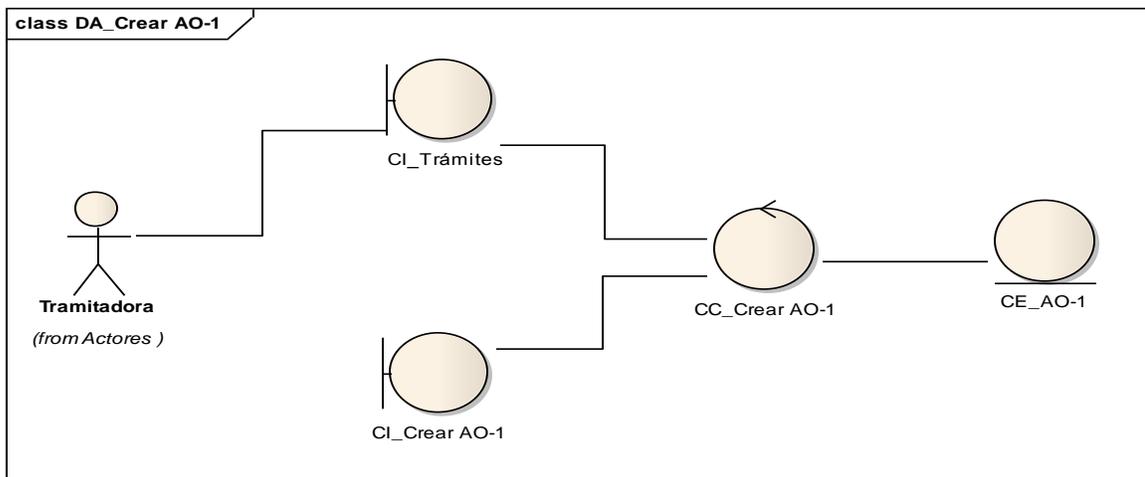


Figura 14. Diagrama de Clases de Análisis. CU_Crear AO-1.

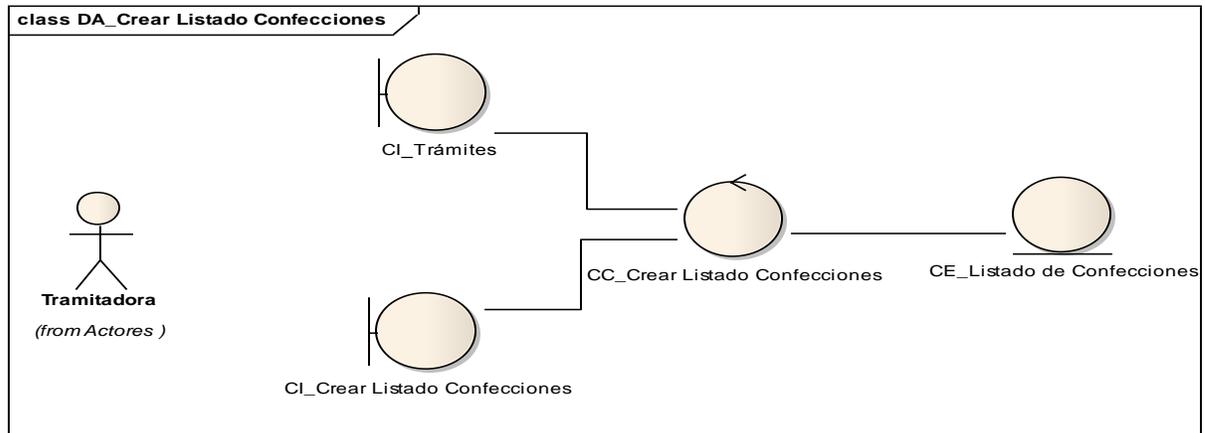


Figura 15. Diagrama de Clases de Análisis. CU_Generar Listado de Confecciones.

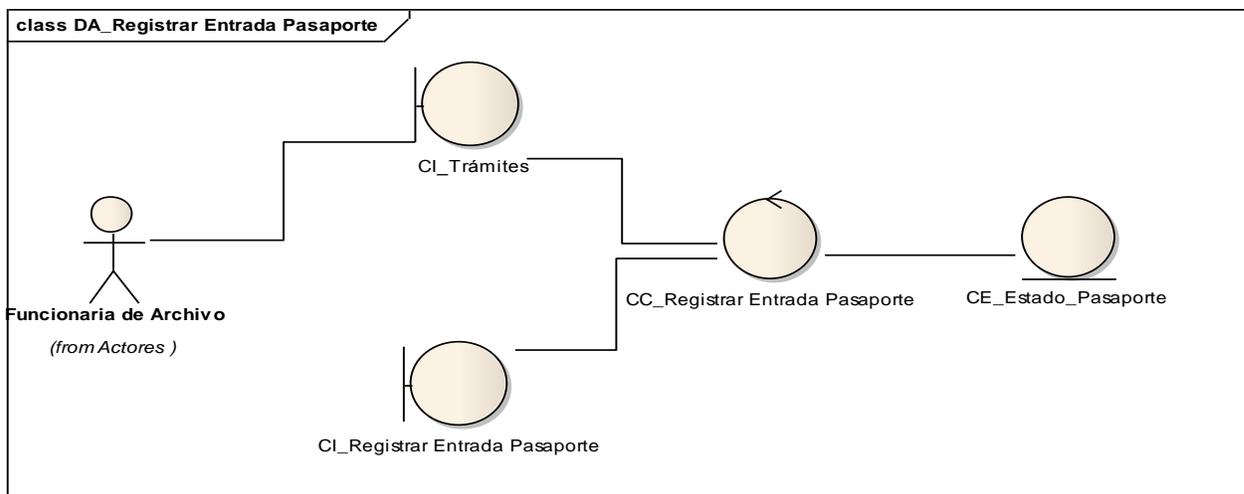


Figura 16. Diagrama de Clases de Análisis. CU_Registrar Entrada Pasaporte.

3.2.3 Diagramas de Comunicación

Un diagrama de comunicación es un diagrama de interacción que destaca la organización estructural de los objetos que envían y reciben mensajes.

A continuación se muestran los diagramas de colaboración de algunos casos de uso del sistema. Los diagramas restantes se encuentran en el Expediente de Proyecto.

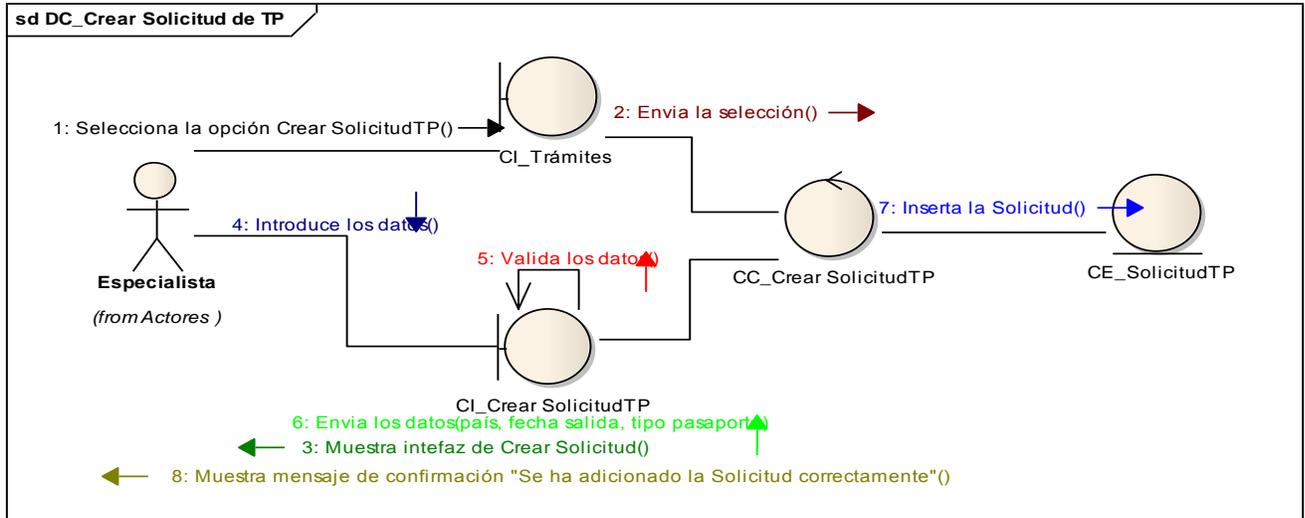


Figura 17. Diagrama de Colaboración. CU_Crear Solicitud de TP.

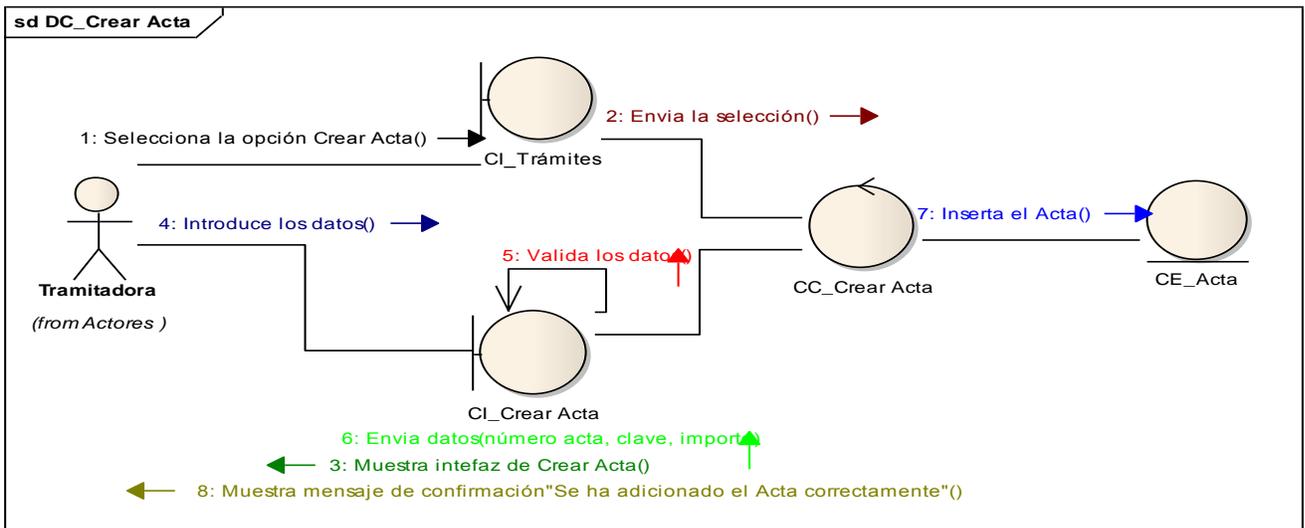


Figura 18. Diagrama de Colaboración. CU_Crear Acta.

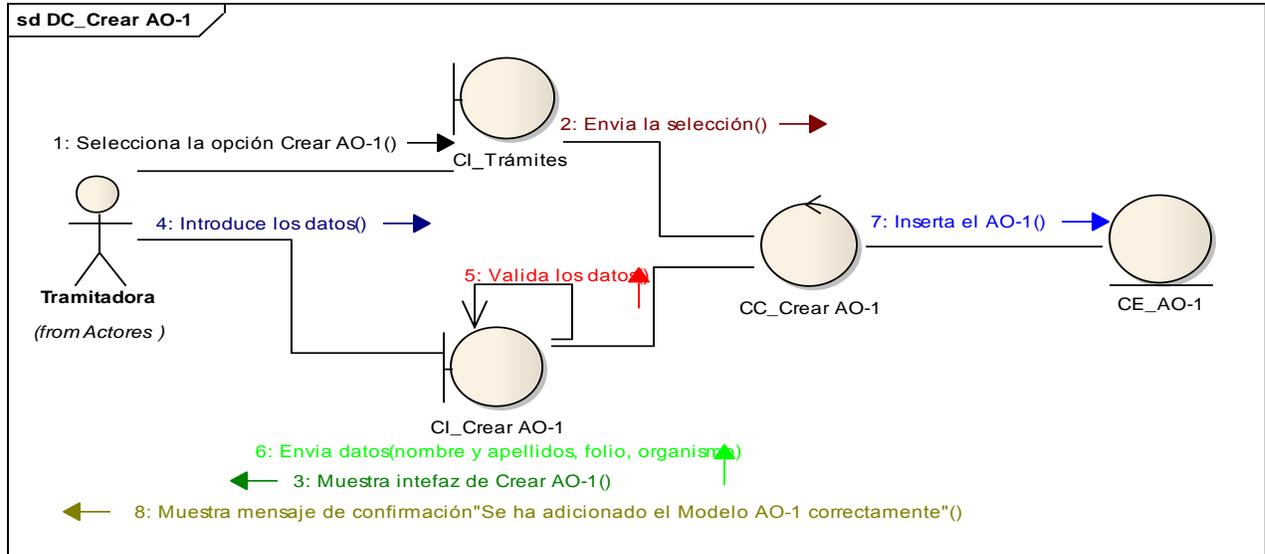


Figura 19. Diagrama de Colaboración. CU_Crear AO-1.

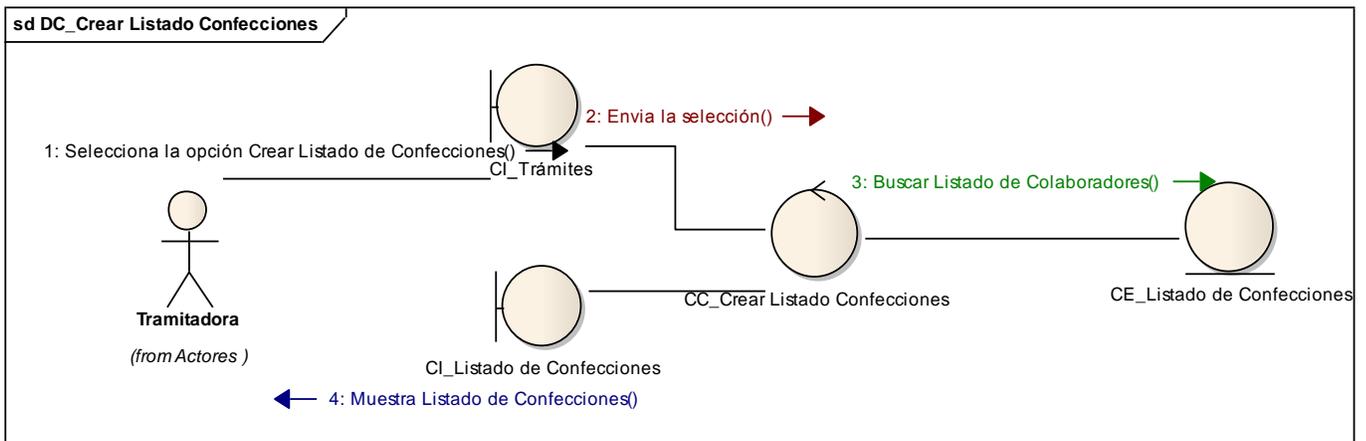


Figura 20. Diagrama de Colaboración. CU_Generar Listado de Confecciones.

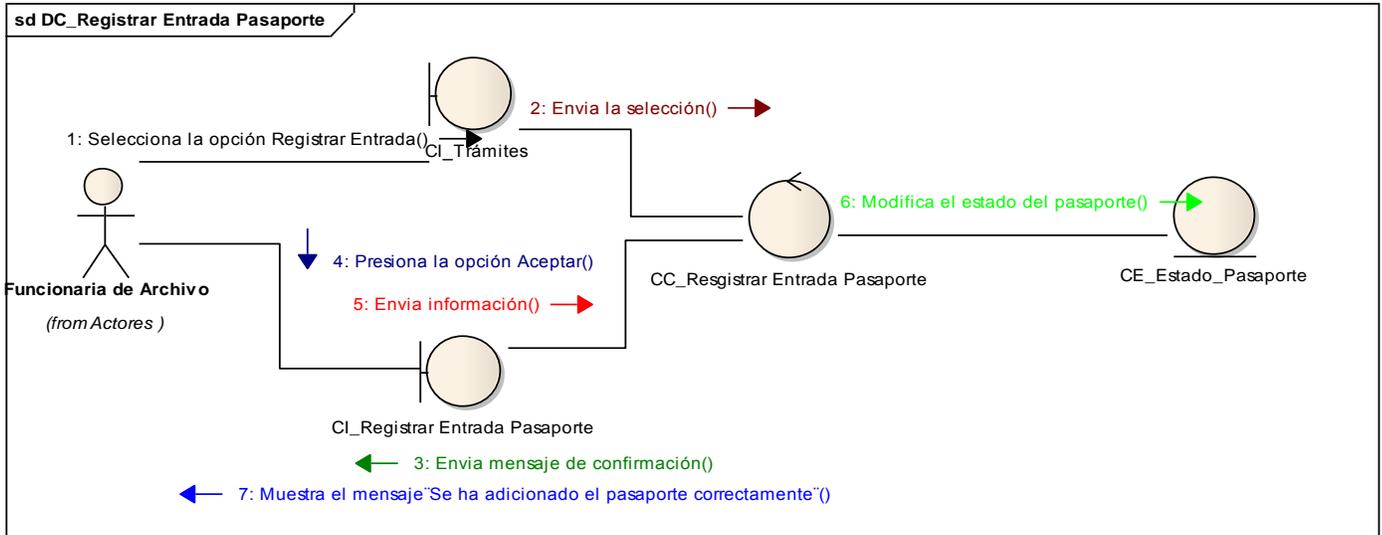


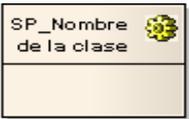
Figura 21. Diagrama de Colaboración. CU_Registrar Entrada Pasaporte.

3.3 Modelo de Diseño

El Modelo de Diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación tienen impacto en el sistema, intentando consolidar una arquitectura sólida y estable para que el sistema sea implementado sin errores. Sirve de abstracción de la implementación y para concebir y documentar el diseño del sistema de software.

3.3.1 Descripción de Elementos del Diseño

Para la realización de los Diagramas de Clases del Diseño, las clases están representadas con estereotipos como: Server Page (Página Servidor), Client Page (Página Cliente) y Form (Formulario), debido a que son los elementos que se utilizan para la representación de una página web.

Estereotipos Web para las Clases del diseño	
Estereotipo	Descripción
 <p>Server Page</p>	<p>Representa una página Web que tiene scripts ejecutados por el servidor. Estos scripts interactúan con los recursos que se encuentran al alcance del servidor. Sólo puede mantener relaciones con objetos que se encuentren en el servidor.</p>

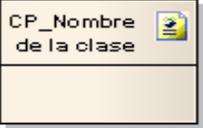
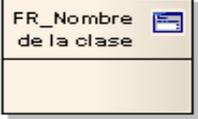
 <p>Client Page</p>	<p>Representan páginas que son mostradas por el navegador web y pueden ser una combinación de algún o algunos lenguajes de marcado, scripts del lado del cliente.</p>
 <p>Form</p>	<p>Representa una colección de campos de entrada que forman parte de la página cliente (Client Page).</p>

Tabla 10. Estereotipos Web para las Clases del Diseño.

<p>Estereotipos para las Relaciones entre las Clases</p>	
<p>Link</p>	<p>Representa un apuntador desde una “client page” hacia una “client page” o “server page”.</p>
<p>Submit</p>	<p>Esta relación siempre ocurre entre una “form” y una “server page”, la “server page” procesa los datos que la “form” le envía (submit).</p>
<p>Build</p>	<p>Sirve para la “server page” que crea la “client page”. Una “server page” puede crear varias “client page”, pero una “client page” sólo puede ser creada por una sola “server page”. Esta relación siempre es unidireccional.</p>
<p>Redirect</p>	<p>Esta es también una relación unidireccional que indica que una página Web redirige hacia otra.</p>

Tabla 11. Estereotipos para las relaciones entre las Clases. (32)

3.3.2 Diagramas de Clases del Diseño

Uno de los artefactos más importantes a obtener del flujo de trabajo de diseño son los diagramas de clases de diseño, donde se exponen las clases que intervienen en las realizaciones de los casos de uso del sistema. Este modelo se puede utilizar para visualizar la implementación y para soportar las técnicas de programación gráfica de la aplicación debido a que en ellos se muestran los atributos y métodos de cada clase y se representa de una forma sencilla la colaboración y las responsabilidades de las distintas clases que forman el sistema. (33)

A continuación se muestran algunos de los Diagramas de Clases del Diseño de casos de uso arquitectónicamente significativos; los restantes pueden encontrarse en el Expediente de Proyecto.

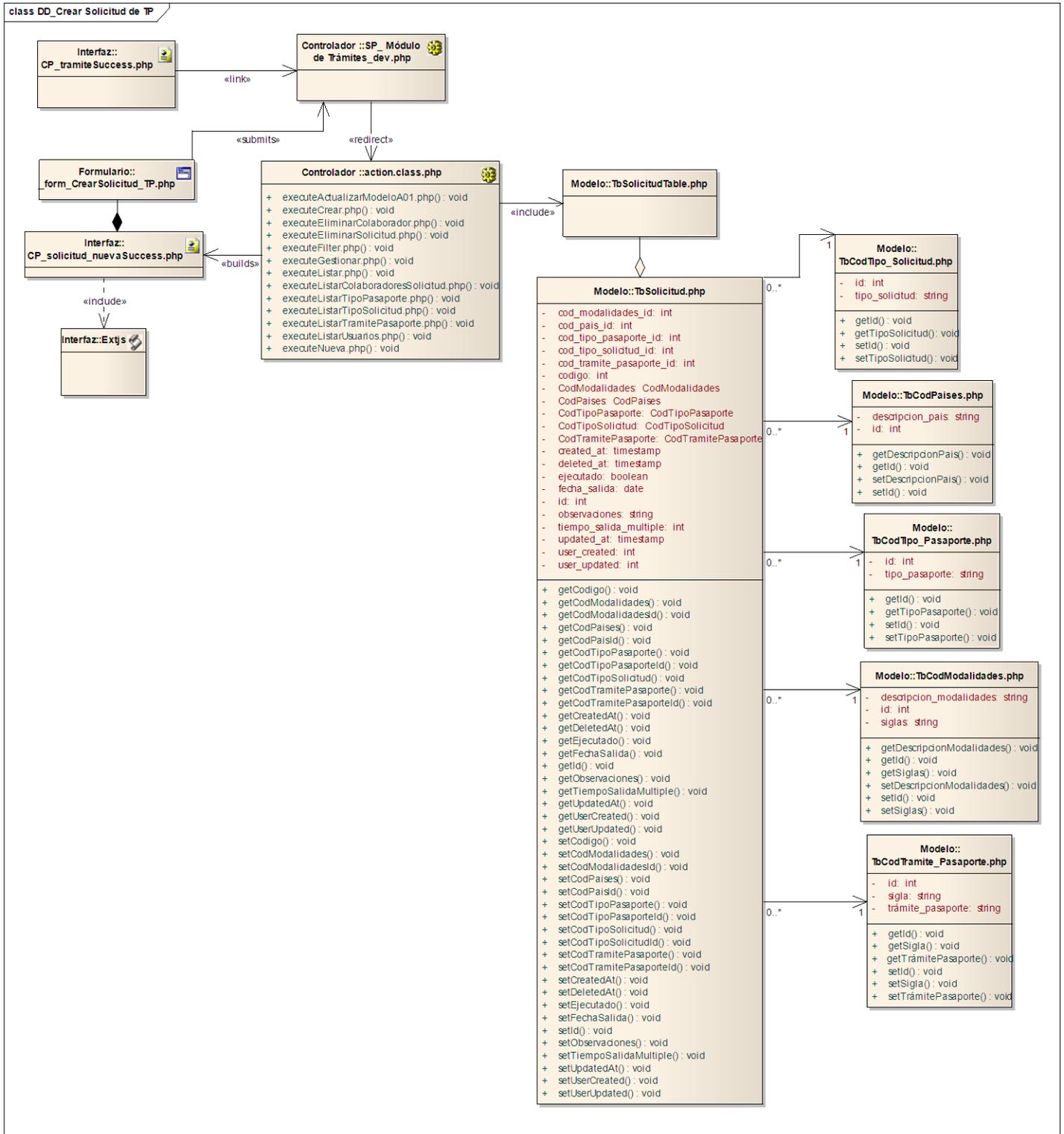


Figura 22. Diagrama de Clases de Diseño. CU_Crear Solicitud de TP.

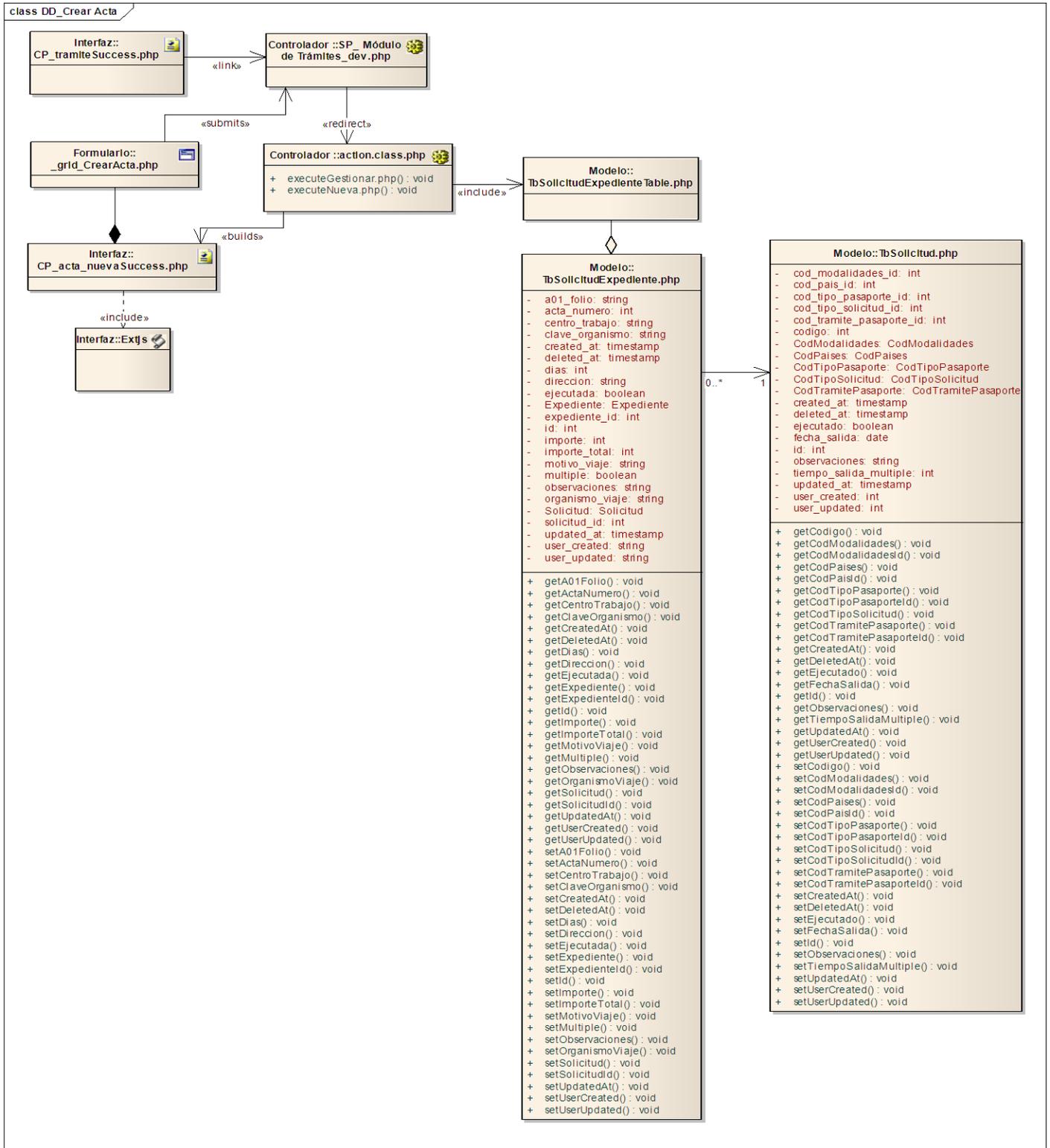


Figura 23. Diagrama de Clases de Diseño. CU_Crear Acta.

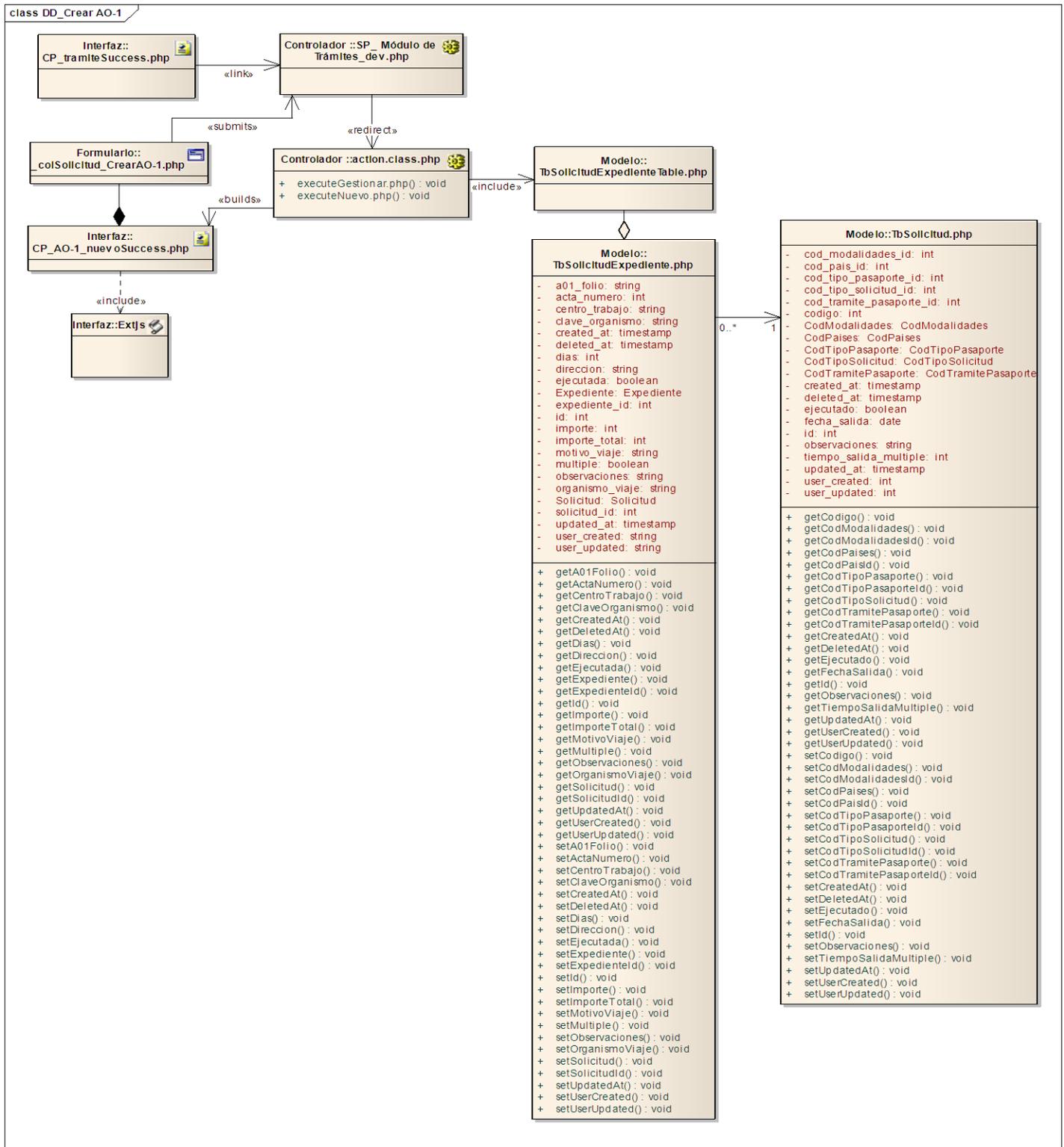


Figura 24. Diagrama de Clases de Diseño. CU_Crear AO-1.

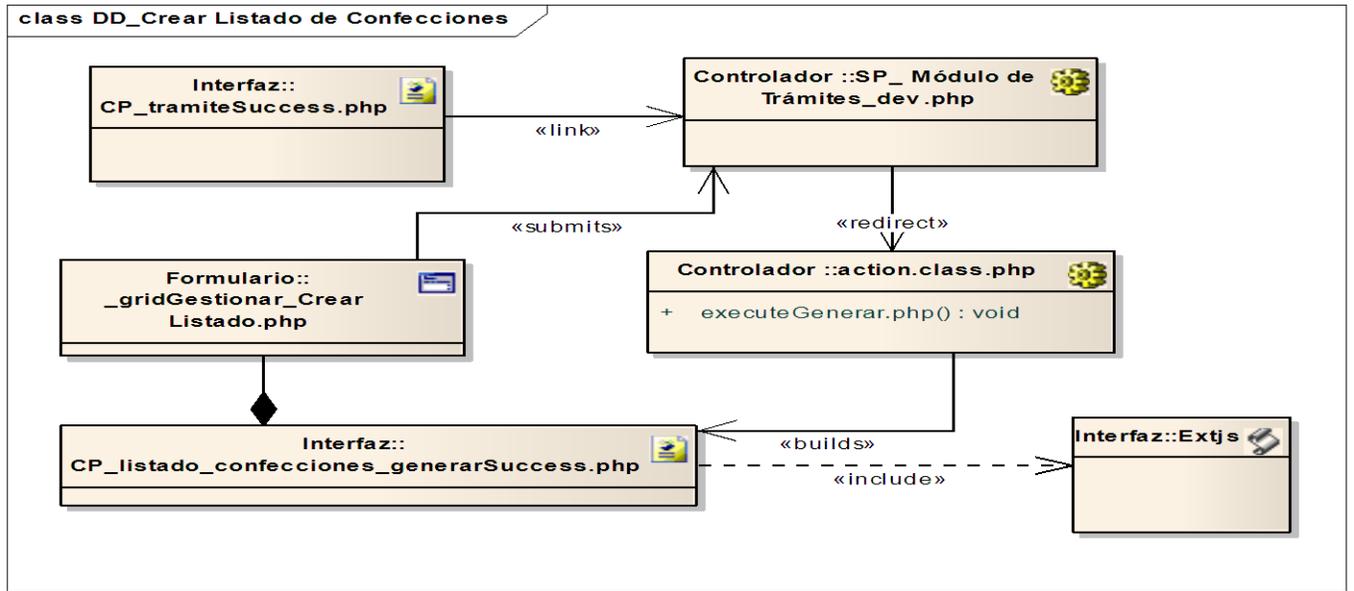


Figura 25. Diagrama de Clases de Diseño. CU_Generar Listado de Confecciones.

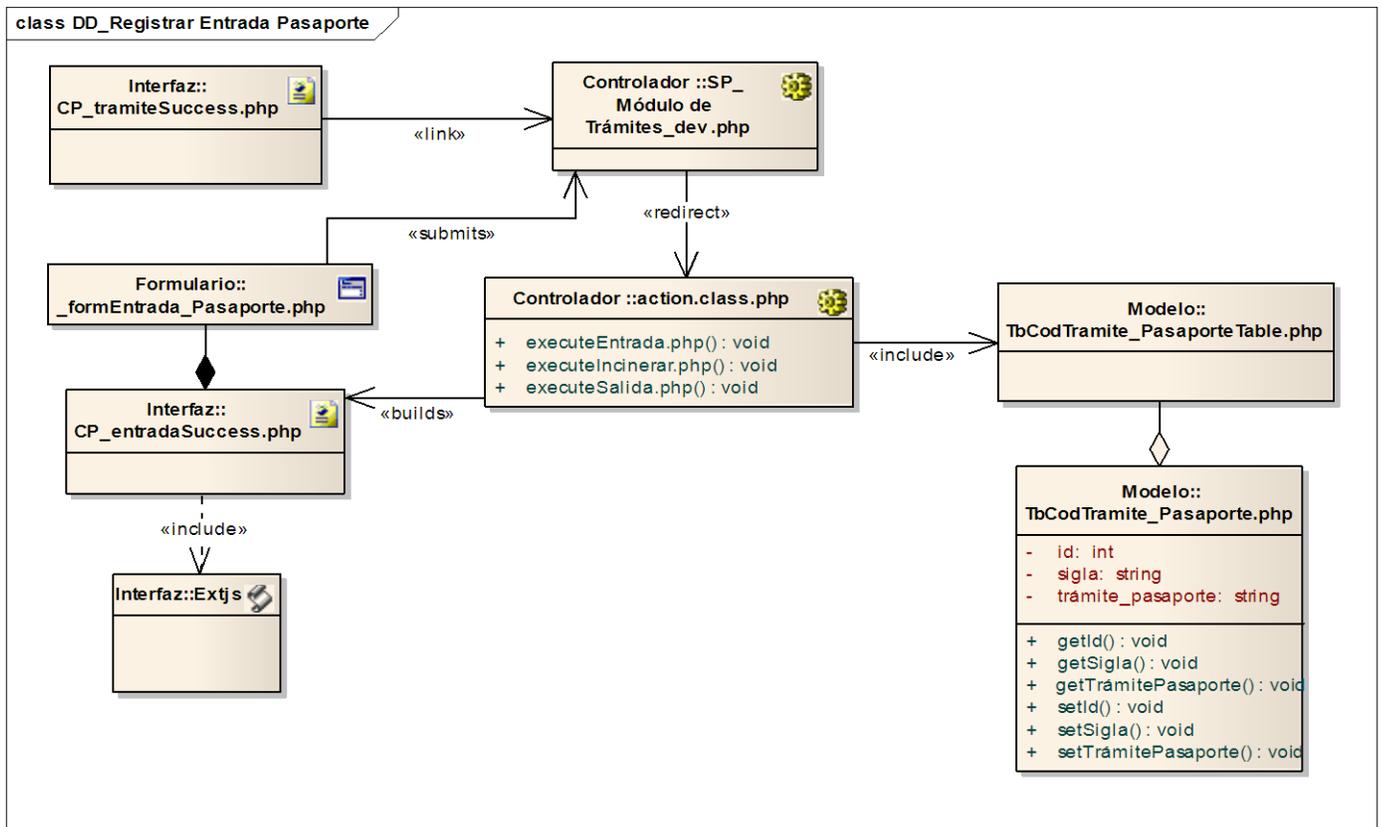


Figura 26. Diagrama de Clases de Diseño. CU_Registrar Entrada de Pasaporte.

3.3.3 Descripción de las Clases del Diseño

Las descripciones de las clases del diseño ayudan a comprender mejor las responsabilidades de las mismas y al mismo tiempo tener una visión más exacta del diseño del sistema.

3.3.3.1 Descripción de Páginas Clientes

Nombre: tramiteSuccess.php
Tipo de clase: Client Page
Descripción general: Esta clase es la encargada de representar el acceso del usuario al sistema, además de atender las selecciones y peticiones que pueda realizar un determinado usuario.

Tabla 12. Descripción de la Client Page: tramiteSuccess.php.

3.3.3.2 Descripción de Páginas Servidoras

Nombre: Módulo de Trámites_dev.php
Tipo de clase: Server Page
Descripción general: Esta página es el controlador frontal del sistema y constituye el punto de entrada único de toda la aplicación en un entorno determinado, al ser la encargada de detectar la acción que se debe ejecutar.

Tabla 13. Descripción de la Server Page: Módulo de Trámites_dev.php.

3.4 Diagrama de Despliegue

Un diagrama de despliegue muestra la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Es utilizado para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos.

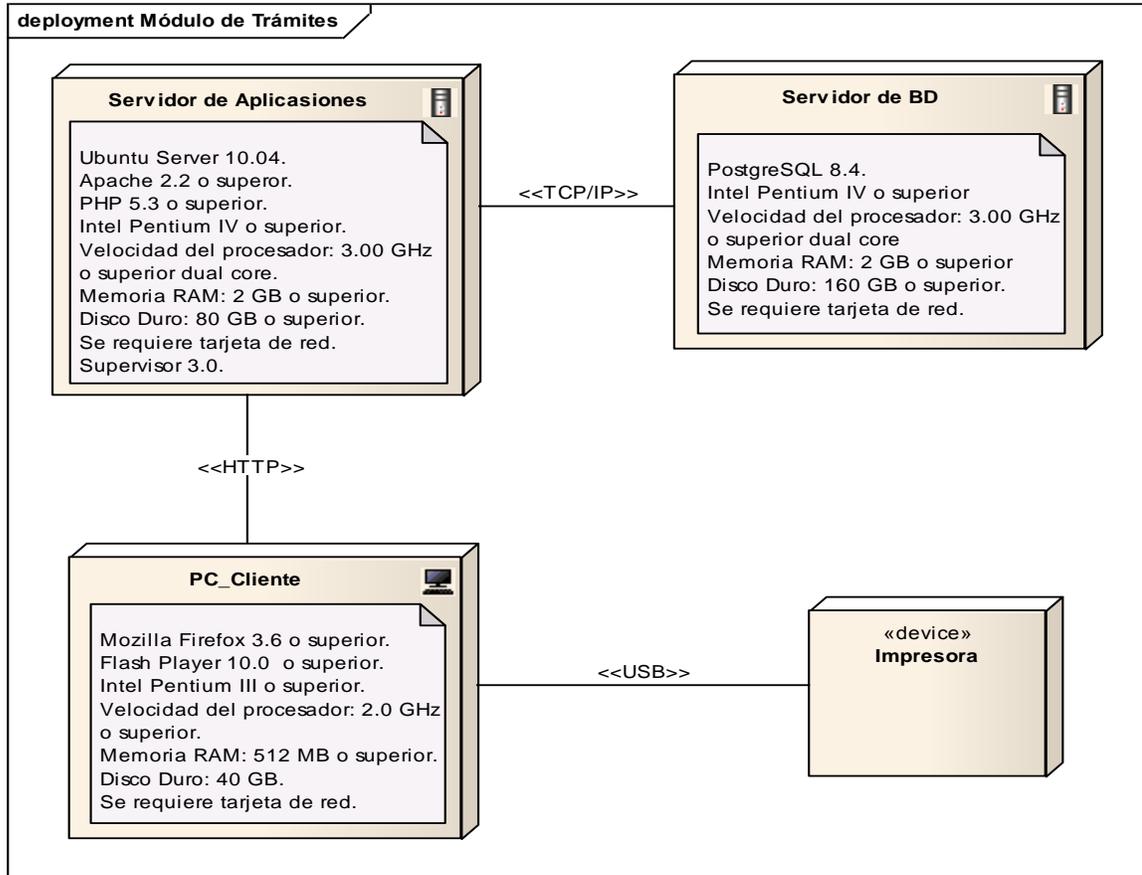


Figura 29. Diagrama de Despliegue.

A partir del desarrollo de este capítulo se llevó a cabo el flujo de trabajo Análisis y Diseño, el cual generó artefactos fundamentales para la elaboración de la solución propuesta. Durante esta etapa fueron definidas las clases del diseño y la relación e interacción entre estas; detallándose así, qué va a hacer el sistema y cómo va a lograr su correcto funcionamiento. Además, se realizó la descripción de la arquitectura del software de acuerdo a las características y estructura, propuestas por el Framework Symfony.

Capítulo 4. Implementación

En este capítulo se exponen los componentes del sistema, así como la forma en la que interactúan para posibilitar su correcto funcionamiento. Basándose en el resultado del diseño del sistema se procederá a implementar las clases y subsistemas en términos de componentes, tales como: ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. Se explican las estrategias de codificación, tratamiento de errores y seguridad. Se muestra una vista del Diagrama de Componentes y se describe el Modelo de Datos sobre el cual se va a realizar la implementación.

4.1 Modelo de Datos

Un modelo de datos es básicamente, una descripción de un contenedor de datos, así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son elementos físicos, son abstracciones que permiten la implementación de un sistema eficiente de base de datos. Por lo general, se refieren a algoritmos y conceptos matemáticos; y aportan la base conceptual para diseñar aplicaciones que hacen uso intensivo de datos.

Para el funcionamiento del sistema se diseñó un modelo de datos de clases persistentes que describe los distintos datos, de la información que se manejará en el software. A continuación se muestra el Modelo de Datos definido para describir la representación lógica y física de la información persistente que se maneja en el Módulo de Trámites.

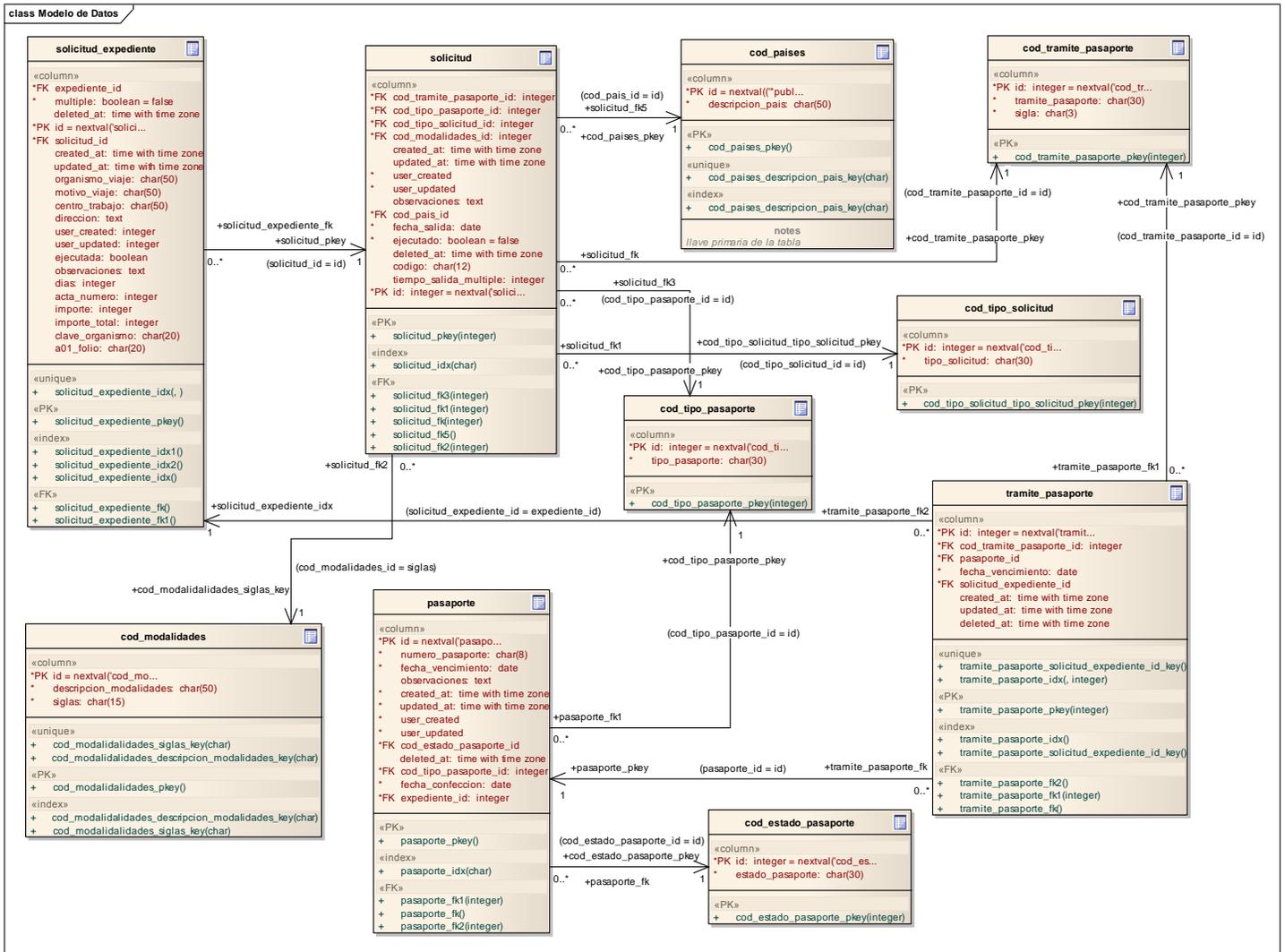


Figura 30. Modelo de Datos.

4.2 Modelo de Implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se encuentran: datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. Este artefacto describe cómo se implementan los componentes, congregándolos en subsistemas organizados en capas y jerarquías, y señala las dependencias entre éstos. (34)

4.2.1 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. Modelan la vista estática de un sistema. Muestran la organización y las dependencias entre un conjunto de componentes. (35)

El diagrama que se muestra a continuación, representa la estructura y organización del Módulo de Trámites en términos de componentes, y la aplicación mediante la arquitectura Modelo Vista Controlador que propone el Framework Symfony.

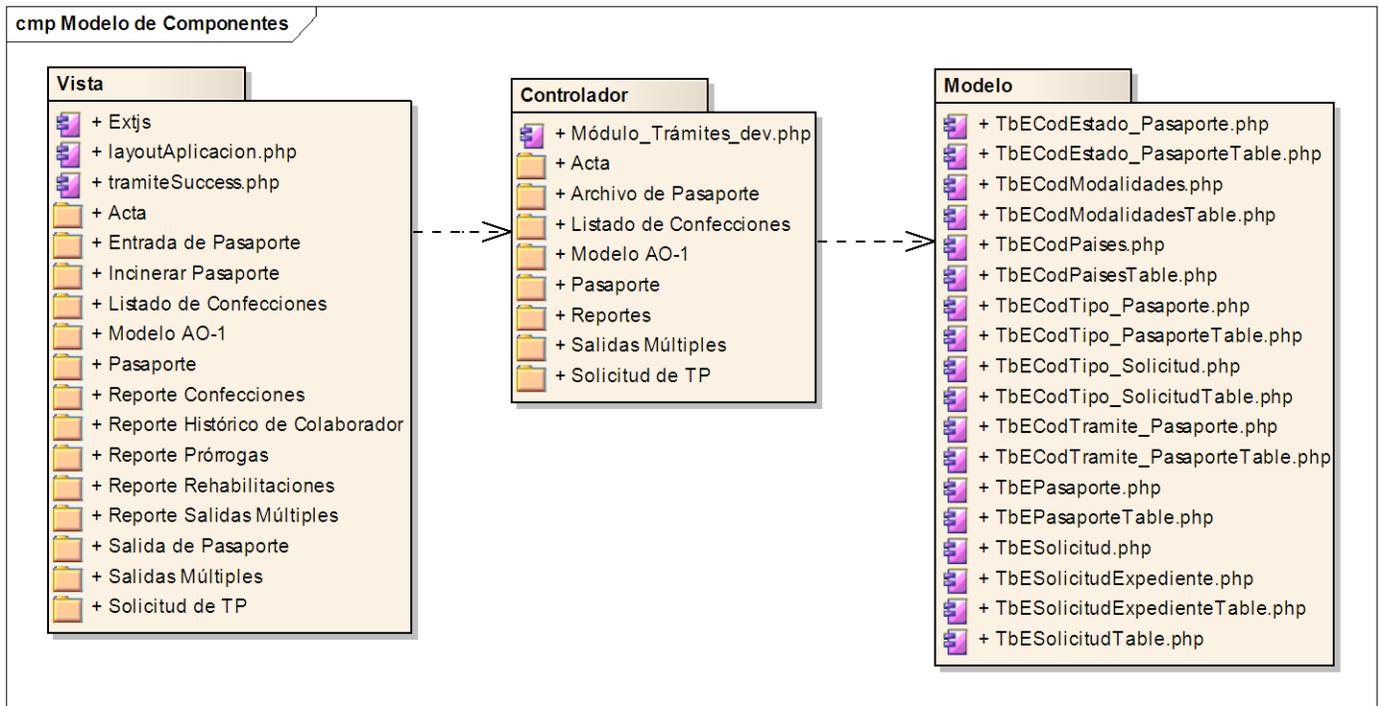


Figura 31. Diagrama de Componentes.

Para una mejor visualización de los componentes, se muestran a continuación los diagramas de componentes referentes a la Vista y al Controlador. El resto de los diagramas, así como cada uno de los paquetes en detalle, se pueden encontrar en el Expediente del Proyecto.

En la Vista se encuentran agrupados todos los formularios y las clases Success que representan la forma en que el usuario puede interactuar con la aplicación.

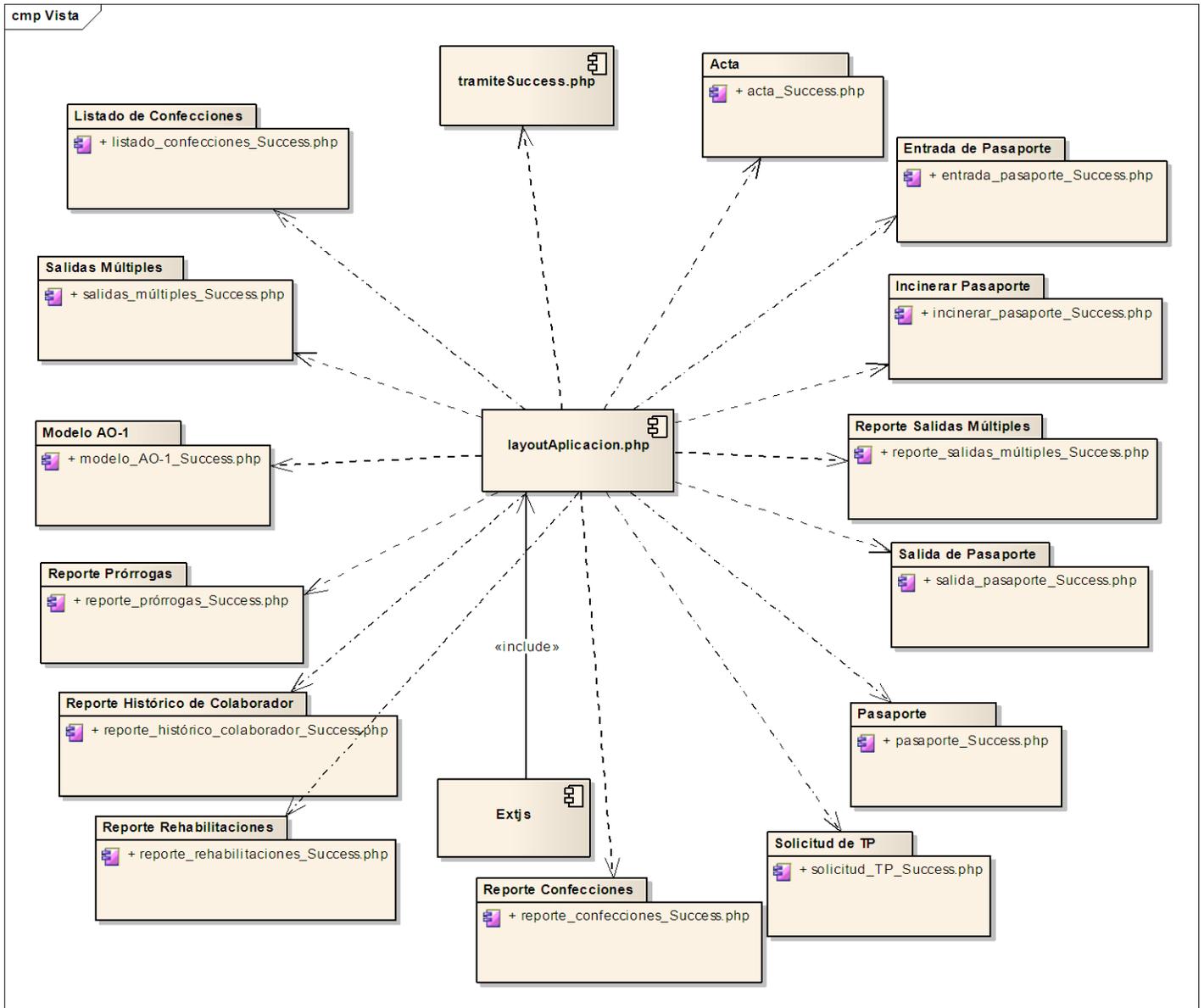


Figura 32. Diagrama de Componentes Paquete Vista.

En el Controlador se encuentran agrupados el Controlador Frontal y las clases actions de cada uno de los módulos de la aplicación, en los cuales se gestionan las peticiones y se realizan los cambios apropiados en el modelo o en la vista.

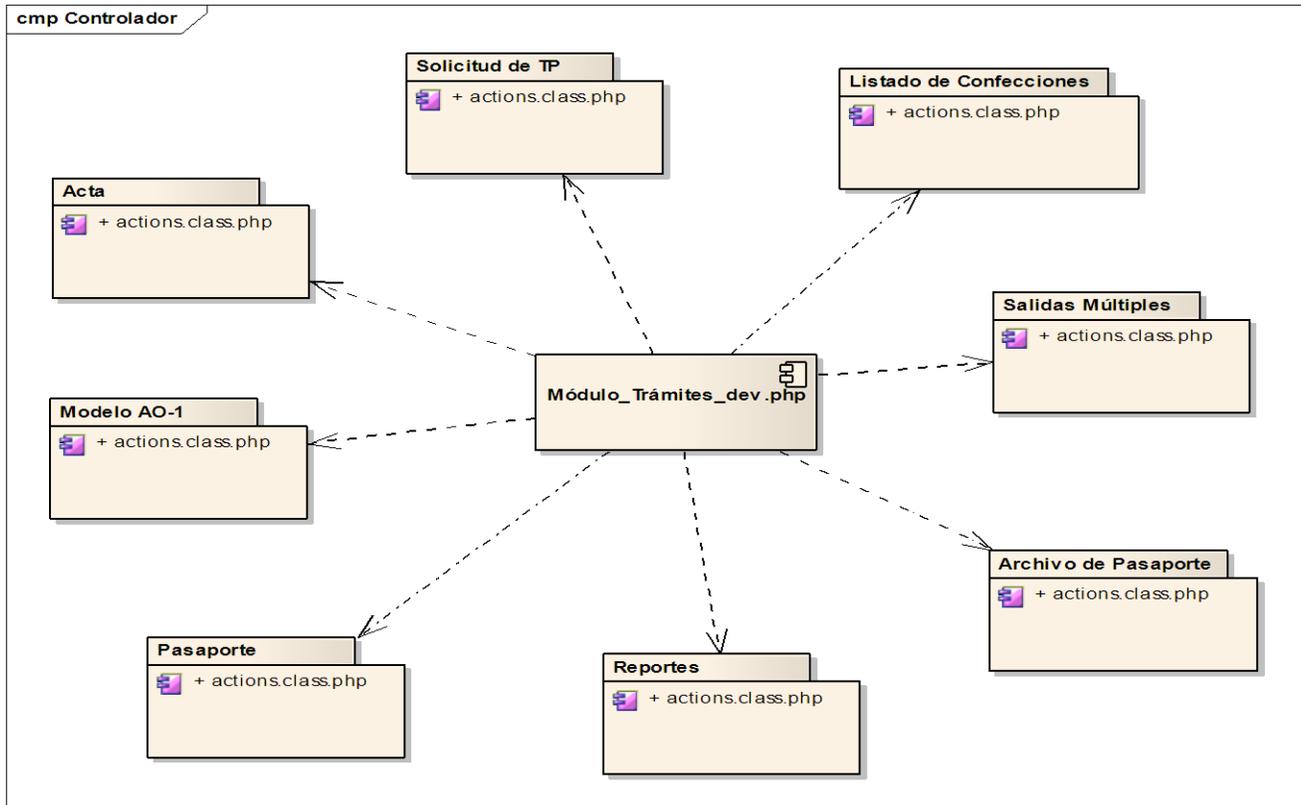


Figura 33. Diagrama de Componentes Paquete Controlador.

4.3 Tratamiento de Errores

El tratamiento de errores es un paso fundamental para obtener una buena elaboración del sistema, debido a que garantiza la integridad y confidencialidad de la información que se maneja en él. Para minimizar estos errores uno de los métodos utilizados fue el trabajo con el lenguaje de programación Java Script, ya que mediante el mismo se le informan al usuario muchos de estos errores.

En este módulo serán emitidos los mensajes de errores en el momento en que el usuario cometa alguna equivocación, por ejemplo: que no complete los datos obligatorios, que introduzca los datos erróneamente, que busque un elemento que exceda la cantidad de caracteres permitidos, que realice una búsqueda y no arroje resultados, también se realizaron validaciones de parte del servidor evitando errores lógicos para asegurar que los datos introducidos sean realmente los correctos.

4.4 Seguridad

La seguridad e integridad del sistema se garantizan una vez que se creen los usuarios con los permisos necesarios para la gestión de la información en la base de datos del sistema. A partir de los roles definidos

para el sistema, solo acceden a cada módulo de la aplicación aquellos usuarios definidos previamente por un administrador. Además, el sistema gestor de base datos debe mantenerse en un lugar restringido y asegurado, además de realizarle salvadas continuas para evitar las pérdidas de datos que hayan sido insertados en el sistema.

En este capítulo se mostraron los resultados obtenidos durante la etapa de implementación, donde los principales artefactos obtenidos fueron los Diagramas de Componentes y el Modelo de Datos, que representan los componentes del software y la representación lógica y física de la información. Se logró alcanzar una aplicación con todas las funcionalidades previstas, que compense las principales necesidades de los clientes. Se analizó también la importancia que representa el tratamiento de errores y la seguridad para la construcción de la aplicación.

Conclusiones

La realización del presente trabajo de diploma ha cumplido con el objetivo propuesto, contribuyendo a mejorar el proceso de gestión de trámites de pasaportes de los colaboradores de la salud, por lo que se puede concluir que:

- Luego de realizar un estudio de los sistemas de gestión de trámites de pasaportes se analizaron los procesos realizados en la Unidad Central de Cooperación Médica, obteniéndose una guía para el posterior análisis, diseño e implementación del sistema.
- Mediante la confección de los artefactos pertenecientes a las Disciplinas de Trabajo: Modelado de Negocio, Gestión de Requerimientos, Análisis y Diseño, Implementación y Prueba, según lo propuesto por el Proceso Unificado de Desarrollo, se generó la documentación asociada a la solución propuesta.
- Se desarrolló el módulo de Trámites, basado en las tecnologías y arquitectura definidas por el Centro de Informática Médica y el Departamento SAS, que cumple con los requerimientos para la gestión de trámites de pasaportes en el Departamento de Trámites de la Unidad de Central de Cooperación Médica.

Recomendaciones

Por las experiencias alcanzadas durante la realización de este trabajo y con vista a enriquecer la solución propuesta, se recomienda:

- Crear los roles de Especialista, Tramitador, Funcionaria de Archivo y Supervisor para una segunda versión del sistema, con el objetivo de garantizar que cada trabajador solamente tenga acceso a la información que le corresponde, siguiendo el principio de mínimo privilegio de seguridad informática.

Referencias Bibliográficas

1. Historia Colaboración Médica en Cuba. [En línea] [Citado el: 20 de 10 de 2010.] <http://www.sld.cu/sitios/bmn/temas.php?idv=12389>.
2. Idem 1. [En línea]
3. **Hernández Sampieri, Roberto, Fernández Collado, Carlos y Baptista Lucio, Pilar.** Metodología de la investigación científica. s.l. : Segunda edición, 1998. pág. 501.
- 4.Cooperación Internacional. [En línea] [Citado el: 22 de 10 de 2010.] <http://www.cendeisss.sa.cr/cooperacion/coope1a.html>.
- 5.Gestión de la Información. [En línea] [Citado el: 5 de 1 de 2011.] informationmanagement.wordpress.com/category/gestion/gestion-de-la-informacion.
6. SAIME. [En línea] [Citado el: 2 de 11 de 2010.] <http://www.saime.gob.ve/pasaportes.php>.
- 7.SITRON. [En línea] [Citado el: 2 de 11 de 2010.] http://www.consuladoargentinoenmiami.org/sitron/tramites_pasaportes.php.
8. **Mejias Ayala, Reiseer, Quintero Ruiz, Keytia.** Sistema Informatizado de Cooperación Internacional. 2008.
9. RUP. [En línea] [Citado el: 10 de 11 de 2010.] www.utvm.edu.mx/Organoinformativo.
10. Qué es RUP. [En línea] [Citado el: 10 de 11 de 2010.] www.scribd.com/doc/12812403/Que-es-RUP.
- 11.Introducción a UML 2.1 . [En línea] [Citado el: 10 de 11 de 2010.] www.sparxsystems.com.ar/platforms/uml_tool.html.
- 12.SGBD. [En línea] [Citado el: 15 de 10 de 2010.] www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/.
- 13.PostgreSQL.Manual del usuario de PostgreSQL. [En línea] [Citado el: 15 de 11 de 2010.] www.scribd.com/doc/5703210/Manual-del-usuario-de-PostgreSQL.
- 14.Definición de Ajax. [En línea] [Citado el: 10 de 3 de 2011.] <http://www.ajaxya.com.ar/temarios/descripcion.php?cod=8&punto=1>.

15. Symfony, la guía definitiva | LibrosWeb.es. [En línea] [Citado el: 20 de 11 de 2010.] www.librosweb.es/symfony.
16. Características de Symfony. [En línea] [Citado el: 20 de 11 de 2010.] es.wikipedia.org/wiki/Symfony.
17. ExtJS. [En línea] [Citado el: 20 de 11 de 2010.] blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/.
18. PHP5. [En línea] [Citado el: 25 de 11 de 2010.] es.wikipedia.org/wiki/PHP.
19. Manual de PHP. [En línea] [Citado el: 25 de 11 de 2010.] www.desarrolloweb.com/manuales/58/.
20. JavaScript. [En línea] [Citado el: 25 de 11 de 2010.] www.ulpgc.es/otros/tutoriales/JavaScript/cap1.htm.
21. Características de Apache. [En línea] [Citado el: 27 de 11 de 2010.] acsblog.es/articulos/trunk/.../Apache/html/x31.html.
22. Manual de Apache. [En línea] [Citado el: 27 de 11 de 2010.] www.linux-cd.com.ar/manuales/.../ch-httpd.html.
23. Mozilla Firefox. [En línea] [Citado el: 27 de 11 de 2010.] www.maestrosdelweb.com/editorial/firefox/.
24. Enterprise Architect. [En línea] [Citado el: 27 de 11 de 2010.] www.freedownloadmanager.org/es/downloads/Arquitecto_de_Empresa_para_UML_2.1_3280_p/.
25. NetBeans. [En línea] [Citado el: 5 de 12 de 2010.] es.wikipedia.org/wiki/NetBeans.
26. [En línea] [Citado el: 5 de 12 de 2010.] blogultura.com/java/netbeans-6-8-
27. Patrón "Modelo-Vista-Controlador". [En línea] [Citado el: 5 de 12 de 2010.] www.neleste.com/modelo-vista-controlador/.
28. Arquitectura Cliente-Servidor. [En línea] [Citado el: 10 de 12 de 2010.] www.desarrolloweb.com/.../arquitectura-cliente-servidor.html.
29. Patrones GRASP. [En línea] [Citado el: 10 de 1 de 2011.] jorgesaavedra.wordpress.com/category/patrones-grasp/.
30. Clases del Análisis. [En línea] [Citado el: 15 de 1 de 2011.] www.kybele.etsii.urjc.es/docencia/AIR.../%5BAIR-1011%5DTema4.pdf.
31. Idem 28. [En línea]

32. **Macías, C. Milestone. Milestone.** [En línea] [Citado el: 5 de 2 de 2011.] www.milestone.com.mx/articulos/uso_de_uml_en_aplicaciones_web.htm.
33. Diagramas de clases del diseño. [En línea] [Citado el: 10 de 2 de 2011.] es.wikipedia.org/wiki/Ingeniería_del_diseño.
34. Modelo de Implementación. [En línea] [Citado el: 15 de 2 de 2011.] merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=96&Itemid=297.
35. Diagrama de Componentes. [En línea] [Citado el: 15 de 2 de 2011.] www.sparxsystems.com.ar/resources/tutorial/uml2_componentdiagram.html.

Bibliografía

- Ajax. [En línea] [Citado el: 10 de 5 de 2011.] <http://es.wikipedia.org/wiki/AJAX>.
- Arquitectura Cliente-Servidor. [En línea] [Citado el: 10 de 12 de 2010.] www.desarrolloweb.com/arquitectura-cliente-servidor.html.
- Características de Apache. [En línea] [Citado el: 27 de 11 de 2010.] acsblog.es/articulos/trunk/.../Apache/html/x31.html.
- Características de Symfony. [En línea] [Citado el: 20 de 11 de 2010.] es.wikipedia.org/wiki/Symfony.
- Clases del Análisis. [En línea] [Citado el: 15 de 1 de 2011.] www.kybele.etsii.urjc.es/docencia/AIR.../%5BAIR-1011%5DTema4.pdf.
- Clases del Diseño. [En línea] [Citado el: 20 de 1 de 2011.] es.wikipedia.org/wiki/Ingeniería_del_diseño.
- Cooperación Internacional. [En línea] [Citado el: 22 de 10 de 2010.] <http://www.cendeiss.com.co/cooperacion/coope1a.html>.
- Definición de Ajax. [En línea] [Citado el: 10 de 3 de 2011.] <http://www.ajaxya.com.ar/temarios/descripcion.php?cod=8&punto=1>.
- Diagrama de Componentes. [En línea] [Citado el: 15 de 2 de 2011.] www.sparxsystems.com.ar/resources/tutorial/uml2_componentdiagram.html.
- Diagramas de clases del diseño. [En línea] [Citado el: 10 de 2 de 2011.] es.wikipedia.org/wiki/Ingeniería_del_diseño.
- Eguíluz Pérez, Javier.** www.librosweb.es. [En línea] [Citado el: 15 de Enero de 2009.]
- Enterprise Architect. [En línea] [Citado el: 27 de 11 de 2010.] www.freedownloadmanager.org/es/downloads/Arquitecto_de_Empresa_para_UML_2.1_3280_p/.
- ExtJS. [En línea] [Citado el: 20 de 11 de 2010.] blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/.
- Gestión de la Información. [En línea] [Citado el: 5 de 1 de 2011.] informationmanagement.wordpress.com/category/gestion/gestion-de-la-informacion.

-
- Hernández Sampieri, Roberto, Fernández Collado, Carlos y Baptista Lucio, Pilar.** Metodología de la investigación científica. s.l. : Segunda edición, 1998. pág. 501.
- Hernández, Rolando Alfredo y Coello, Zayda.** El paradigma cuantitativo de la investigación científica. s.l. : Editorial Universitaria, 2002. pág. 112.
- Herramientas Web para la enseñanza. Protocolos de Comunicación. neo.lcc.uma.es. [En línea] [Citado el: 26 de Enero de 2009.] <http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>.
- Historia Colaboración Médica Cubana. [En línea] [Citado el: 13 de Octubre de 2010.] <http://www.ain.cu/vida/historia/historia.htm>.
- Historia Colaboración Médica en Cuba. [En línea] [Citado el: 20 de 10 de 2010.] <http://www.sld.cu/sitios/bmn/temas.php?idv=12389>.
- Introducción a UML 2.1 . [En línea] [Citado el: 10 de 11 de 2010.] www.sparxsystems.com.ar/platforms/uml_tool.html.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** biblioteca.uci.cu. [En línea] [Citado el: 2009 de Abril de 15.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.
- JavaScript. [En línea] [Citado el: 25 de 11 de 2010.] www.ulpgc.es/otros/tutoriales/JavaScript/cap1.htm.
- La Web del Programador. [En línea] http://www.lawebdelprogramador.com/news/mostrar_new.php?id=72&texto=PostgreSQL&n1=394636&n2=1&n3=0&n4=0&n5=0&n6=0&n7=0&n8=0&n9=0&n0=0.
- Macías, C. Milestone.** [En línea] [Citado el: 5 de 2 de 2011.] www.milestone.com.mx/articulos/uso_de_uml_en_aplicaciones_web.htm.
- Manual de Apache. [En línea] [Citado el: 27 de 11 de 2010.] www.linux-cd.com.ar/manuales/.../ch-httpd.html.
- Manual de PHP. [En línea] [Citado el: 25 de 11 de 2010.] www.desarrolloweb.com/manuales/58/.
- Manual del usuario de PostgreSQL. [En línea] [Citado el: 15 de 11 de 2010.] www.scribd.com/doc/5703210/Manual-del-usuario-de-PostgreSQL.
- Mejias Ayala, Reiseer, Quintero Ruiz, Keytia.** Sistema Informatizado de Cooperación Internacional. 2008.

- .Metodologías. [En línea] Instituto Nacional Estadísticas Informáticas. [Citado el: 26 de Enero de 2009.] <http://www.inei.gov.pe/web/metodologias/attach/lib616/INDE.HTM>.
- .**Minnick, Chris y Valentine, Chelsea.** XHTML Serie Práctica. Nueva York. Estados Unidos : s.n., 2000.
- .Modelo de Implementación. [En línea] [Citado el: 15 de 2 de 2011.] merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=96&Itemid=297.
- .MozillaFirefox. [En línea] [Citado el: 27 de 11 de 2010.] www.maestrosdelweb.com/editorial/firefox/.
- .NetBeans. [En línea] [Citado el: 5 de 12 de 2010.] es.wikipedia.org/wiki/NetBeans.
- .Patrón "Modelo-Vista-Controlador". [En línea] [Citado el: 5 de 12 de 2010.] www.neleste.com/modelo-vista-controlador/.
- .Patrones GRASP. [En línea] [Citado el: 10 de 1 de 2011.] jorgesaavedra.wordpress.com/category/patrones-grasp/.
- .PHP5. [En línea] [Citado el: 25 de 11 de 2010.] es.wikipedia.org/wiki/PHP.
- .Portal de Salud de Cuba. [En línea] Sistema Nacional de Salud. [Citado el: 27 de Enero de 2009.] http://www.sld.cu/sistema_de_salud/aspectos.html#misi3n.
- .PostgreSQL. [En línea] [Citado el: 27 de Enero de 2009.] <http://www.postgresql.org/about/press/presskit82.html.es>.
- .Qué es Internet. www.civila.com. [En línea] [Citado el: 27 de Enero de 2009.] <http://www.civila.com/desenredada/que-es.html>.
- .Qué es RUP. [En línea] [Citado el: 10 de 11 de 2010.] www.scribd.com/doc/12812403/Que-es-RUP.
- .RUP. [En línea] [Citado el: 10 de 11 de 2010.] www.utvm.edu.mx/Organoinformativo.
- .SAIME. [En línea] [Citado el: 2 de 11 de 2010.] <http://www.saime.gob.ve/pasaportes.php>.
- .SGBD. [En línea] [Citado el: 15 de 10 de 2010.] www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/.
- .SITRON. [En línea] [Citado el: 2 de 11 de 2010.] http://www.consuladoargentinoenmiami.org/sitron/tramites_pasaportes.php.
- .SPARX systems. [En línea] [Citado el: 2009 de Abril de 15.] http://www.sparxsystems.com.ar/resources/tutorial/component_model.html.

- Symfony,la guía definitiva|LibrosWeb.es. [En línea] [Citado el: 20 de 11 de 2010.] www.librosweb.es/symfony.
- Tutorial de Doctrine. [En línea] [Citado el: 25 de 3 de 2011.] <http://web.ontuts.com/tutoriales/utilizando-doctrine-como-orm-en-php/>.
- Valerio, Adrián Anacleto**. www.epidataconsulting.com. [En línea] [Citado el: 28 de Enero de 2009.] http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15.
- Vázquez, José Antonio Gallego**. Desarrollo Web con PHP y MySQL. Madrid : Ediciones Anaya Multimedia, 2003. pág. 20.
- Zaninotto, Francois y Potencier, Fabien**. www.librosweb.es. [En línea] [Citado el: 28 de Enero de 2009.] http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html.

Anexos

Anexo 1. Diagrama de actividades.

Diagrama de Actividad: Confeccionar Solicitud de Trámite de Pasaporte

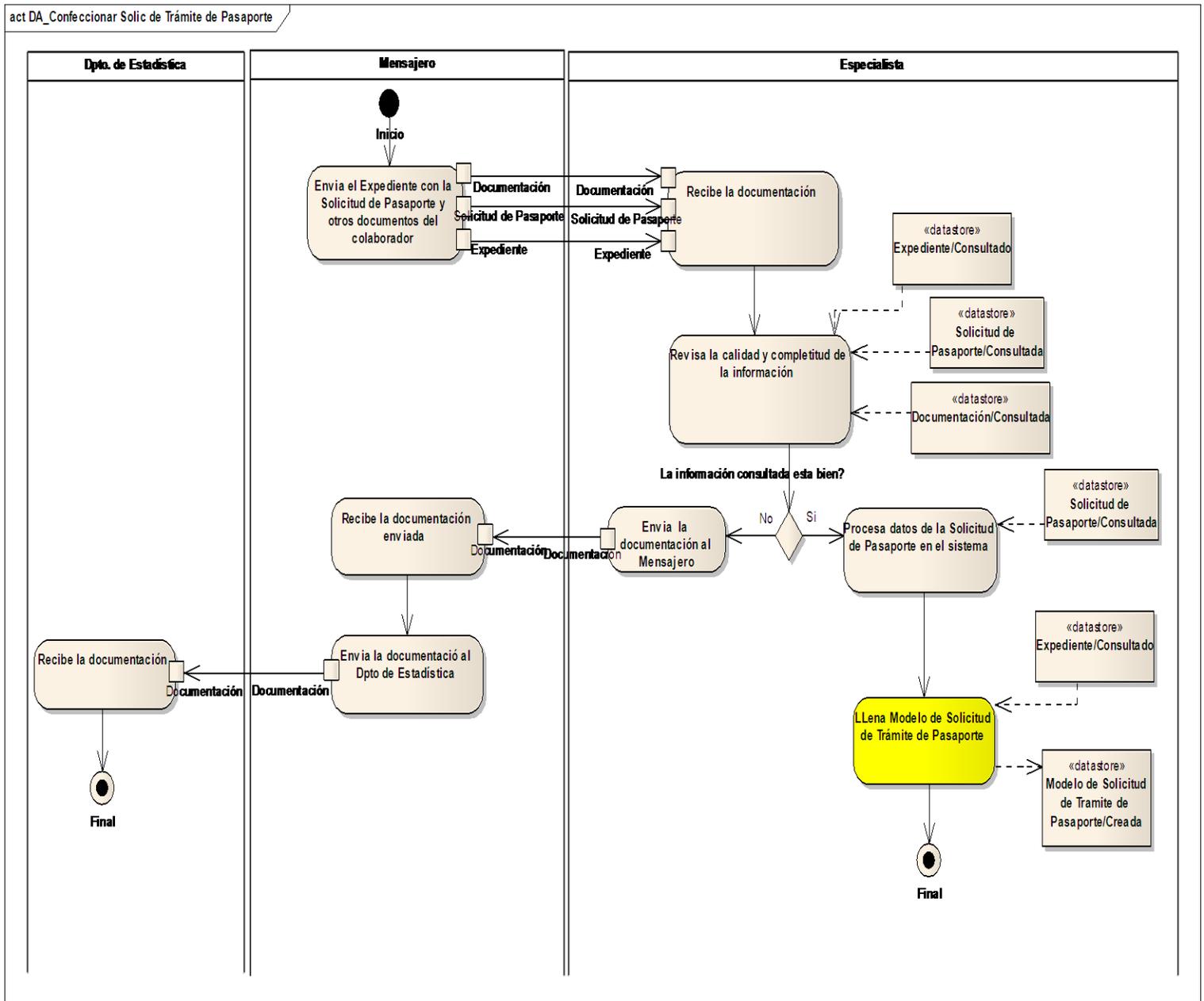


Figura 34: Diagrama de Actividad. Confeccionar Solicitud de Trámite de Pasaporte .

Diagrama de Actividad: Tramitar Pasaporte

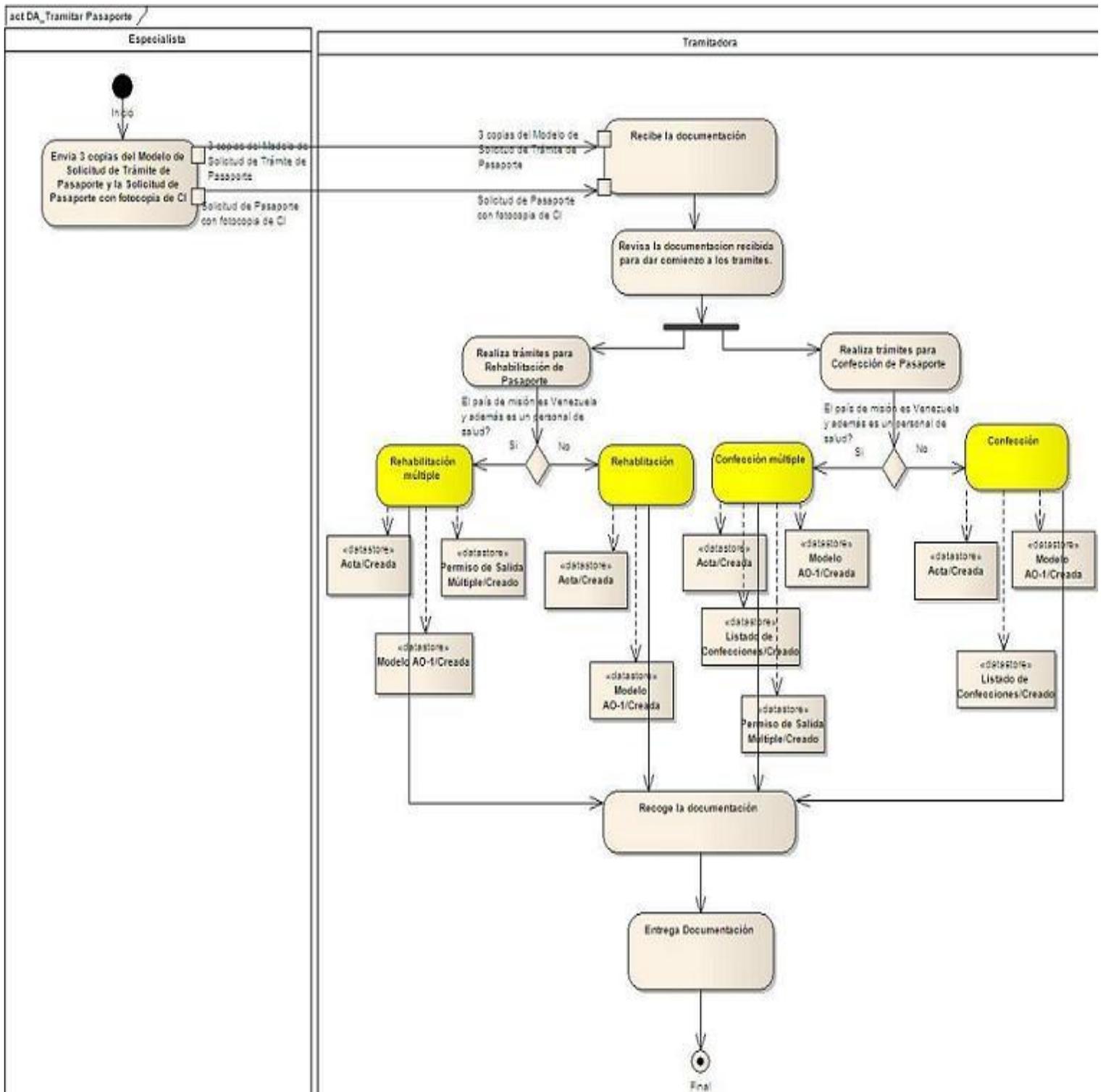


Figura 35: Diagrama de Actividad. Tramitar Pasaporte.

Diagrama de Actividad: Emitir Libro de Entrada

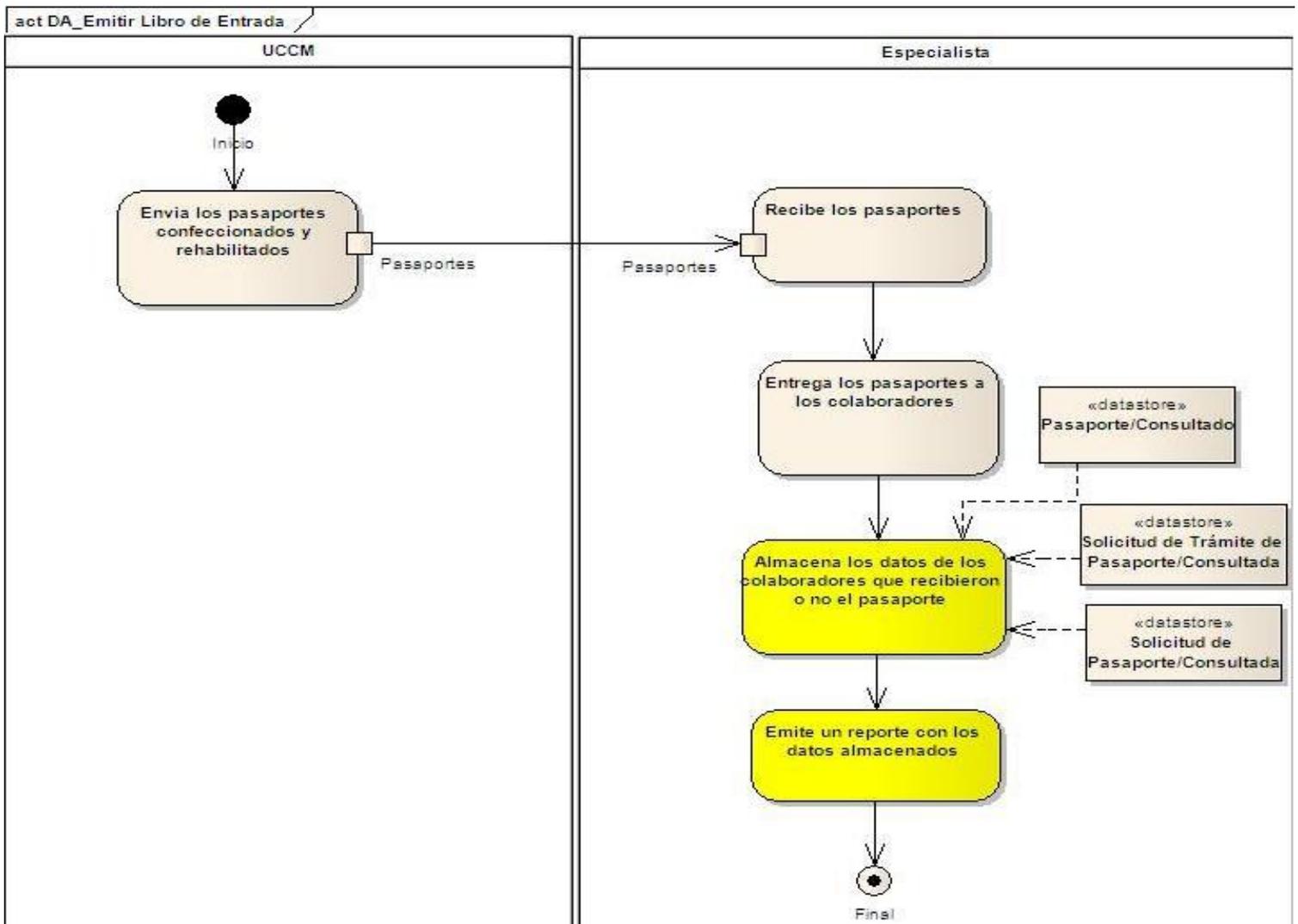


Figura 36: Diagrama de Actividad. Emitir Libro de Entrada.

Diagrama de Actividad: Entrada de Pasaporte

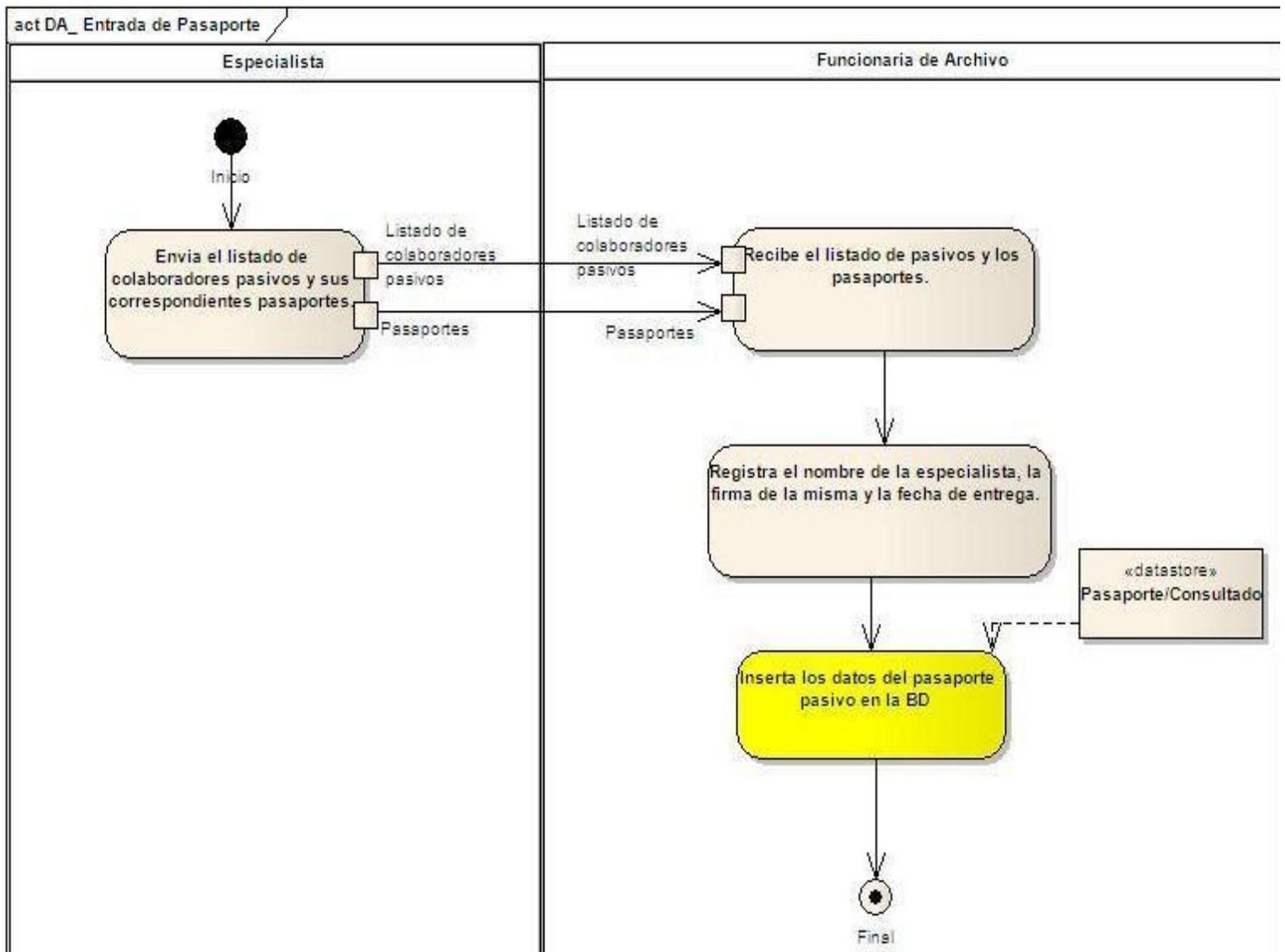


Figura 37: Diagrama de Actividad. Entrada de Pasaporte.

Glosario de Términos

Casos de uso arquitectónicamente significativos: Casos de uso que ayudan a mitigar los riesgos más importantes, aquellos que son los más importantes para los usuarios del sistema y aquellos que ayudan a cubrir todas las funcionalidades significativas.

Centro de Informática Médica (CESIM): CESIM surge como parte de la informatización de la salud cubana, que tiene como misión ser un centro de excelencia dedicado al desarrollo de productos, sistemas, servicios y soluciones de alta calidad y competitividad, para la optimización del trabajo y mejoramiento de la atención médica.

Clases: Conjunto de objetos que comparten atributos, operaciones, relaciones y semántica, las mismas representan los conceptos fundamentales del sistema.

Expediente de proyecto: Conjunto de documentos y plantillas estructuradas siguiendo una jerarquía, que constituyen referencia para la documentación de los proyectos de la Universidad de las Ciencias Informáticas.

Framework: Es una estructura de soporte definida en la cual un proyecto de software puede ser organizado y desarrollado.

HTTP: Acrónimo en inglés de Hypertext Transfer Protocol (protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web.

Informatizar: Proceso de aplicar sistemas o equipos informáticos al tratamiento de la información.

Software Libre: Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

TCP/IP: Conjunto de protocolos de red en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras.

Unidad Central de Cooperación Médica (UCCM): La UCCM es la institución encargada de gestionar y almacenar la información referente a los colaboradores cubanos de la salud.

USB: Acrónimo en inglés de Universal Serial Bus (bus universal en serie), es un puerto que sirve para conectar periféricos a un ordenador. Es un estándar que permite la transferencia de información desde o hacia otro periférico.