

Universidad de las Ciencias Informáticas

Facultad 3

Departamento de Aplicaciones Financieras y Aduanales



**Título: “Sistema para documentar y publicar los
procedimientos para realizar trámites comerciales en
Cuba”**

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

Autor: Araí Nieves Delgado

Tutor(es): Ing. Eddie Nelson Beltrán González
Ing. Maylevis Morejón Valdés

Co tutor: Ing. Luis Manuel Betancourt Oliva

LA HABANA, 2015

“AÑO 57 DE LA REVOLUCIÓN”



“[...] la vida nos impondrá siempre la necesidad de una permanente lucha por resolver lo que quede por detrás, lo que se presenta como nuevo y lo que pueda preverse en el futuro [...]”

Fidel Castro Ruz

DEDICATORIA

«Sistema para documentar y publicar los procedimientos para realizar trámites comerciales en Cuba»

DEDICATORIA

A Mima y Papo por confiar siempre en mí, apoyarme en cada momento de mi vida y ser los mejores padres del mundo.

A mi abuela Olga por todo su amor, dedicación y apoyo.

AGRADECIMIENTOS

«Sistema para documentar y publicar los procedimientos para realizar trámites comerciales en Cuba»

AGRADECIMIENTOS

A mi mami linda por ser esa persona que ha sabido guiarme siempre por el mejor de los caminos. Por ser mi amiga, compañera, mi ángel de la guardia. Por todos los consejos, regaños y momentos de felicidad.

A mi papo por ser ese padre tan maravilloso y brindarme todo su amor. Por su comprensión y estar siempre a mi lado a pesar de cualquier circunstancia.

A mi abuela Olga por todo su cariño, preocupación y apoyarme en cada momento de mi vida.

A mi hermana por ser mi amiga y mi apoyo. Por estar siempre en las buenas y las malas y por nunca defraudarme.

A mi sobrina por ser esa niña que me ha permitido ser su ejemplo y por confiar siempre en mí.

A mi tía Adelfa, que a pesar de no encontrarse junto a nosotros, me brindó siempre su apoyo y cariño.

A mi prima Aracelys por todo su apoyo y enseñarme todas las cosas buenas y malas de la vida. Gracias por ser una madre más.

A mi hermanito, por existir y ser esa personita que me ha completado mi vida y siempre demostrarme todo su amor.

A Vicky por todo su cariño incondicional.

A los demás miembros de mi familia por confiar en mí y estar siempre presentes en cada momento.

A mis amigas de toda la vida: Yamila, Leyané, la China, Leanys, Niliem por cada momento que hemos pasado juntas, por todos sus consejos y cada una de sus ocurrencias,

A mi amigo y amor de mi vida Eduardo, por estar siempre presente y ser mi confidente, por todo su amor incondicional.

A mis vecinos, en especial a Rosy y Eduardo por ser parte de mi familia y por esos ricos dulces que siempre disfruté.

AGRADECIMIENTOS

«Sistema para documentar y publicar los procedimientos para realizar trámites comerciales en Cuba»

A mis amigos de mil batallas, en especial a Negrín por siempre estar ahí y sacarme la mejor de mis sonrisas, por todos los buenos momentos y hacerme chistes sin sentidos sabiendo que me reiría de cualquier manera. A Oda por ser la primera persona que conocí y ser en todo este tiempo mi amiga y hermana, apoyarme en cada decisión y ser mi compañera de fiestas. A Aniel por soportar mis pesadeces. A mis compañeras de apto: Marian y Alicia. A Ale por ser una persona especial, ser amigo y estar siempre presente. A los demás chicos del aula: Roly, Yaicel, Jose, Burgos, Sosa, Yuniór, Bauta. A mis compañeros de café: Ernesto, los Jimis, Felox, Jorgito, por todos esos ratos de charlas. A mi amigo Adrián por existir y brindarme todo su amor. Por estar presente y ser mi pañuelo de lágrimas. A todos en especial, les agradezco porque me ayudaron a ser la persona que soy actualmente.

Un agradecimiento a cada uno de los profesores que me acompañaron a lo largo de la carrera, en especial a Maigret, Yeni, Day, Lizandra, Yalice, Yoel y Mavis.

A mis tutores por su paciencia, ayuda, por confiar siempre en mí. A Eddie por soportarme y estar siempre ahí con toda la disposición del mundo y a Levis por ser mi amiga.

A las personas que me ayudaron todo este tiempo, en especial a Diego y a Julio.

Al Team de Yorlen y Leo por toda su ayuda y estar ahí para cualquier duda.

A los miembros de los tribunales que sin sus consejos y experiencias no hubiese sido posible que este trabajo de diploma tuviera la calidad requerida.

A todos muchas gracias.

DECLARACIÓN JURADA DE AUTORÍA

«Sistema para documentar y publicar los procedimientos para realizar trámites comerciales en Cuba»

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

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

Araí Nieves Delgado
Autor

Luis Manuel Betancourt Oliva
Co tutor

Maylevis Morejón Valdés
Tutor

Eddie Nelson Beltrán
Tutor

RESUMEN

El comercio exterior es el intercambio de bienes o servicios existente entre dos o más naciones con el propósito de satisfacer sus necesidades de mercado, tanto internas como externas. En Cuba, es el Ministerio de Comercio Exterior el encargado de controlar dichas actividades, contando con leyes y regulaciones que avalan sus procedimientos. El mismo no contempla un documento rector que contenga descrito de manera centralizada sus procesos, por lo que los interesados en realizar operaciones comerciales deben consultar disímiles fuentes de información para adquirir el conocimiento necesario. Todo ello provoca retardos, equivocaciones y poca transparencia en los procesos.

La presente investigación tiene como objetivo principal desarrollar una herramienta para publicar y actualizar la información referente a los procedimientos de los organismos del país que intervienen en las operaciones del comercio exterior, facilitando la accesibilidad a la misma por parte de los interesados. Para ello, se realizó un estudio sobre las metodologías de desarrollo de software existentes, seleccionando la metodología ágil SXP, así como el empleo del marco de trabajo de PHP Symfony2 y JQuery como librería JavaScript para el desarrollo de interfaces de usuario.

Como resultado de la investigación, se obtuvo una solución informática que permite publicar y documentar la información referente a la realización de trámites comerciales en Cuba, permitiendo un total acceso a esta por parte de los interesados. Todo ello traerá como beneficios una mayor transparencia en los procesos, una mayor agilidad en la consulta de la información y la posibilidad de simplificar los trámites descritos.

PALABRAS CLAVES:

Accesibilidad, comercio exterior, MINCEX, operaciones, pasos, trámites.

INTRODUCCIÓN.....	11
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	15
1.1 Conceptos asociados al dominio del problema	15
1.2 Sistemas de gestión de la información para la realización de trámites.....	17
1.3 Metodologías de desarrollo de software	21
1.4 Captura de requisitos	24
1.5 Herramientas a utilizar	25
1.5.1 Lenguaje de modelado UML.....	25
1.5.2 Herramienta de modelado	25
1.5.3 Marco de trabajo.....	26
1.5.4 Lenguajes de programación	29
1.5.5 Lenguajes de marcado	30
1.5.6 Servidor web.....	30
1.5.7 Sistema gestor de bases de datos.....	31
1.5.8 Entorno de desarrollo integrado (IDE)	31
1.6 Conclusiones parciales.....	32
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	33
2.1 Solución propuesta	33
2.1.1 Concepción inicial del sistema	33
2.2 Captura de requisitos	33
2.2.1 Lista de reserva del producto (LRP)	34
2.3 Diseño de metáfora.....	35
2.3.1 Historias de usuario.....	35
2.3.1.1 Tareas de ingeniería	38
2.4 Arquitectura seleccionada	40
2.5 Modelo de datos	43
2.6 Patrones de diseño	44
2.6.1 Patrones de diseño de la base de datos.....	44
2.6.2 Patrones Patrones de Diseño de Asignación de Responsabilidades o GRASP (Patrones Generales de Software para Asignar Responsabilidades).....	45
2.6.3 Patrones de Comportamiento (Patrones GOF):	47
2.7 Conclusiones parciales	48
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA	49

3.1 Técnicas de validación de requisitos	49
3.2 Validación del diseño de la aplicación web	49
3.2.1 Tamaño Operacional de Clases (TOC)	49
3.2.2 Relaciones entre Clases (RC)	51
3.3 Pruebas	54
3.3.1 Pruebas de funcionalidad	54
3.3.2 Pruebas unitarias	57
3.4 Impacto de la solución propuesta.....	60
CONCLUSIONES	62
RECOMENDACIONES.....	63
REFERENCIAS BIBLIOGRÁFICAS.....	64
ANEXOS.....	68

Índice de tablas:

Tabla 1: Resultado de los sistemas estudiados.....	20
Tabla 2: Fragmento de la plantilla LRP (Prioridad: Muy alta)	34
Tabla 3: HU_1: Autenticar de usuario.....	35
Tabla 4: HU_3: Crear trámite.	36
Tabla 5: T_1: Generación de la base de datos.	38
Tabla 6: T_2 Montaje del ambiente de desarrollo con la integración de los marcos de trabajo seleccionados para el desarrollo.....	39
Tabla 7: T_3: Realizar el Mapeo de la Base de Datos.	39
Tabla 8: T_10: Diseñar las interfaces requeridas para la funcionalidad: Crear trámite.	40
Tabla 9: T_11: Implementar funcionalidad: Crear trámite.	40
Tabla 10: Valores de las variables de calidad: Responsabilidad, Complejidad y Reutilización.	51
Tabla 11: Valores de la variable: Acoplamiento, Complejidad, Reutilización y Cantidad de pruebas.	53
Tabla 12: A-1-1 correspondiente a la HU_1: Autenticación de usuario.....	55
Tabla 13: A-3-1 correspondiente a la HU_3: Crear trámite	55
Tabla 14: Caso de prueba para el Camino básico 1	59
Tabla 15: Caso de prueba para el Camino básico 2.....	59
Tabla 16: Caso de prueba para el Camino básico 3.....	59

Índice de figuras:

Figura 1: Patrón MVC en Symfony2 (42)	41
Figura 2: Clases entidades	41
Figura 3: Plantillas de las vistas	42
Figura 4: Clases controladoras	42
Figura 5: Diagrama de clases	43
Figura 6: Modelo de Datos	44
Figura 7: Uso del patrón Llaves subrogadas.....	45
Figura 8: Ejemplo del patrón Experto en la clase trámite	45
Figura 9: Evidencia del patrón Creador en la clase TramiteController.php.....	46
Figura 10: Evidencia del patrón Bajo acoplamiento en la clase: Entidad.....	46

Figura 11: Evidencia del patrón Alta Cohesión en las clases	47
Figura 12: Cantidad de procedimientos por clases	50
Figura 13: Resumen de cantidad de procedimiento por clases	50
Figura 14: Resultado de las variables: Responsabilidad, Complejidad y Reutilización.....	51
Figura 15: Cantidad de relaciones de uso por clases.....	52
Figura 16: Resumen de relaciones de uso.....	52
Figura 17: Resultado de las variables: Acoplamiento, Complejidad, Reutilización y Cantidad de pruebas.....	53
Figura 18: No conformidades detectadas a través de las pruebas de Caja Negra	56
Figura 19: Funcionalidad updateAction().....	57
Figura 20: Grafo del flujo del código de la función updateAction	58

INTRODUCCIÓN

El comercio es una práctica de tipo económica que consiste en comprar, vender, o intercambiar productos, materiales o servicios. Esta actividad puede realizarse dentro de un país e implicar a individuos y empresas dentro del mismo territorio o espacio geográfico, o por el contrario, efectuarse fuera de los límites de una nación, caso que se conoce formalmente como comercio exterior. La principal característica que presenta el comercio exterior es que el mismo supone el ingreso al país de moneda extranjera, que le significan al estado, la generación de riquezas (1).

Los gobiernos para el desarrollo de cada uno de sus procedimientos, requieren de acciones gubernamentales expresadas en políticas que simplifiquen sus trámites. Actualmente, las empresas se enfrentan a gestiones muy complejas que entorpecen sus negocios y desestimulan su interés por tener presencia internacional, de ahí la necesidad de contar con organizaciones e instituciones especializadas en el control de los trámites necesarios que impacten positivamente en el aumento de la productividad y la competitividad nacional y regional (2).

El avance de la informática y el aumento de su impacto social, traen consigo la creación de organizaciones e instituciones que eligen incorporar sistemas de apoyo a la toma de decisiones, logrando así una mayor dinámica en sus procesos de negocio. Es por ello, que muchos programas de desarrollo y aplicación de las Tecnologías de la Información y las Comunicaciones (TIC), intentan crear o mejorar sus legislaciones con el objetivo de promover las nuevas tecnologías y un entorno “cero papel” en relación con los trámites del comercio exterior. Sin embargo, en este ámbito aún existe un extenso camino por recorrer, y es por ello, que se reconoce el uso principal de las TIC como una herramienta más para otorgar mayor transparencia a los procesos y mejorar el funcionamiento de la administración pública (2).

La mayoría de los países han trabajado para desarrollar un marco jurídico que permita brindar una adecuada seguridad para las transacciones en línea, que no impongan barreras innecesarias al comercio, elimine aquellos obstáculos que limitan el libre desenvolvimiento del mismo y permita competir en los mercados internacionales. Además, han creado un formulario único de comercio exterior, con la finalidad de verificar el cumplimiento de todos los requisitos en un solo lugar, con lo cual se ha logrado un mayor grado de eficiencia en las operaciones de comercio exterior (2).

INTRODUCCIÓN

Sistema para documentar y publicar los procedimientos para realizar trámites comerciales en Cuba

El Ministerio de Comercio Exterior de Cuba (MINCEX) es el encargado de dirigir, ejecutar y controlar la aplicación de la política del estado y del gobierno en cuanto a las actividades del comercio, la inversión y la colaboración en el exterior (4). Además, se encuentra inmerso en el acelerado proceso que vive la sociedad cubana de acercarse a la era digital. Los primeros pasos se dedicaron a la creación de la infraestructura tecnológica que garantiza, en la actualidad, una adecuada comunicación del ministerio, de ahí la necesidad de dedicar esfuerzos y recursos a la informatización de la actividad de comercio exterior. El MINCEX tiene dentro de sus proyecciones el desarrollo de una Ventanilla Única (VU), la cual sería el único punto de acceso de toda la documentación requerida para realizar trámites comerciales. La Conferencia de las Naciones Unidas sobre Comercio y Desarrollo (UNCTAD¹), establece que para llevar a cabo un buen proceso de desarrollo de una VU es necesario que cada trámite se encuentre bien documentado y con fácil acceso para todos los interesados. Luego que estos son descritos deben ser simplificados y como último paso se realiza la implementación de la VU (6).

Para la realización de una operación comercial en Cuba, las autoridades involucradas se rigen por leyes y regulaciones que avalan sus procedimientos, pero no contempla con un documento rector que contenga descrito de manera centralizada dichos procesos, es decir, se desconoce el cómo poder acceder a la información sobre los procedimientos para realizar un trámite, debido a que cada operación tiene características diferentes en dependencia del producto o mercancía a comercializar y para hacer un procedimiento se requiere de cumplir con determinados pasos.

Teniendo en cuenta lo anteriormente expuesto se plantea el siguiente **problema a resolver**:

¿Cómo publicar y actualizar la información referente a los procedimientos de las entidades y organismos del país que intervienen en las operaciones del comercio exterior, de manera que se facilite la accesibilidad a la misma por parte de los interesados?

Partiendo del problema planteado, se tiene como **objeto de estudio** de la presente investigación: Los sistemas de gestión de contenidos.

Enmarcado en el **campo de acción**: Los sistemas de gobierno electrónico.

¹ UNCTAD: Del inglés United Nations Conference on Trade and Development.

INTRODUCCIÓN

Sistema para documentar y publicar los procedimientos para realizar trámites comerciales en Cuba

El **objetivo general** del presente trabajo es:

Desarrollar una herramienta para publicar y actualizar la información referente a los procedimientos de las entidades y organismos del país que intervienen en las operaciones del comercio exterior, de manera que se facilite la accesibilidad a la misma por parte de los interesados.

Se plantea como **idea a defender**:

Si se desarrolla una herramienta para documentar y publicar la información referente a los procedimientos que intervienen en las operaciones del comercio exterior se facilitará el acceso a la misma por parte de los interesados.

Para complementar el objetivo general se definen como **objetivos específicos** los siguientes:

1. Elaborar el marco teórico referencial sobre los sistemas para la documentación y publicación de procedimientos.
2. Diseñar las funcionalidades del sistema para documentar y publicar los procedimientos de los trámites comerciales.
3. Implementar el sistema para documentar y publicar los procedimientos de los trámites comerciales.
4. Validar la solución propuesta mediante la aplicación de métricas y pruebas que evidencien la viabilidad de la solución.

Para desarrollar la presente investigación se han empleado los siguientes **métodos científicos de investigación**.

Métodos teóricos:

- **Análisis histórico-lógico:** Se empleó con el objetivo de analizar la evolución, de forma cronológica, de los sistemas de gestión de la información para la realización de trámites existentes en el ámbito nacional e internacional.
- **Analítico-Sintético:** Se utilizó para procesar la información contenida en documentos y otras fuentes bibliográficas durante la investigación y a partir de ello se realizó un estudio acerca de las características y necesidades vigentes para la realización de la solución.

Métodos empíricos:

- **Observación:** Se empleó como método para observar los distintos tipos de sistemas existentes, que sirvieron como objeto de análisis y comparación para establecer las características y elementos fundamentales.

El presente trabajo se encuentra estructurado en tres capítulos, los cuales se anuncian a continuación:

Capítulo 1: Abarca lo relacionado con la fundamentación teórica de la investigación. Se enuncian los principales conceptos abordados. Se realiza además un estudio de las soluciones ya existente, así como las herramientas, metodologías y lenguajes a usar en el desarrollo de la solución.

Capítulo 2: Se realiza el análisis y diseño de la propuesta de solución, describiendo los diferentes artefactos esenciales que se generan durante el desarrollo de la misma. Se describe además la especificación de las funcionalidades, la arquitectura, los modelos de diseño y de datos, entre otros elementos que conforman el ciclo de desarrollo del sistema.

Capítulo 3: Se hace un análisis de los resultados obtenidos con las pruebas realizadas al sistema, para validar su funcionamiento y desempeño, teniendo como base los artefactos concebidos en las etapas de análisis y diseño para la implementación de la solución.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El presente capítulo recoge los elementos teóricos que fundamentan el desarrollo de un sistema para documentar y publicar de forma centralizada los procedimientos para realizar trámites comerciales en Cuba, que favorezca la accesibilidad a la información. Se reflejan las características esenciales de las diferentes soluciones informáticas existentes, así como los elementos más relevantes de la metodología, lenguajes y herramientas definidas para el desarrollo del producto final.

1.1 Conceptos asociados al dominio del problema

Conferencia de las Naciones Unidas sobre Comercio y Desarrollo (UNCTAD)

Órgano principal de la Asamblea General de las Naciones Unidas que se ocupa de las cuestiones relacionadas con el desarrollo, el comercio y la inversión, desempeñando un rol principal sobre los países menos desarrollados (8).

El análisis de políticas y de investigación sobre los desafíos de la globalización y la integración de los países en desarrollo al sistema de comercio internacional, permite la ayuda en la toma de decisiones para acabar con las desigualdades económicas mundiales. Brinda un foro para que los representantes de todos los países puedan dialogar y debatir sobre las maneras de instaurar un mejor equilibrio económico (10).

La organización ha evolucionado gradualmente hasta llegar a convertirse en una autoridad basada en el conocimiento, cuya labor tiene como objetivo orientar los debates actuales sobre las políticas y la reflexión en materia de desarrollo, velando especialmente a que las políticas nacionales y la acción internacional se complementen mutuamente para lograr un desarrollo sostenible.

Ministerio de comercio exterior de Cuba

El Ministerio del Comercio Exterior y la Inversión Extranjera es el encargado en la República de Cuba, de dirigir, ejecutar y controlar la aplicación de la política del Estado y del Gobierno en cuanto a las actividades del comercio, la inversión y la colaboración en el exterior (12). Es por ello que el comercio exterior de Cuba ejerce un fuerte y determinante impacto en la economía del país. Entre sus funciones se encuentra promover y crear las condiciones para el desarrollo del comercio

exterior con todos los países que acepten tener relaciones comerciales con Cuba y estimular las transacciones comerciales. Otra función es la de llevar a cabo las negociaciones comerciales con otros países y suscribir los convenios y documentos, conforme a lo que en cada caso instruya el Gobierno (14).

En el año 2009, al igual que el resto de los países subdesarrollados, sufrió el impacto de la crisis económica mundial, unido a la permanencia del bloqueo norteamericano impuesto al país desde hace más de 50 años, a los daños provocados por fenómenos asociados al cambio climático y al incremento de los precios y la reducción de la demanda (4).

Dentro de los principales socios comerciales de Cuba, se encuentran Venezuela, China, Rusia, España y Brasil. Desde los años 90, la exportación de servicios ha tenido un creciente papel en la economía cubana, lo que se manifiesta en su influencia cada vez más importante en los resultados del comercio exterior cubano. La venta de servicios profesionales, especialmente en salud, educación, en la biotecnología y en la informática, se consolidan como la mayor fuente de divisas del país. El ambiente sano y seguro que brinda Cuba, para los negocios en general, es un punto a favor determinante para promover las exportaciones de bienes y servicios, la colaboración, la inversión extranjera y avanzar en la sustitución de importaciones, contribuyendo de esta forma al desarrollo productivo y científico-técnico del país (4).

Sistema de gobierno electrónico

Según la Organización de las Naciones Unidas, se refiere al uso de las TIC, por parte de las instituciones de Gobierno, para mejorar los servicios de información que se ofrecen a los ciudadanos; aumentar la eficiencia y eficacia de la gestión pública; incrementar sustantivamente la transparencia del sector público y la participación ciudadana (16).

Su objetivo es contribuir a la óptima aplicación de las TIC para simplificar y mejorar los servicios e información ofrecidos por el Estado a las personas y los procesos transversales de las instituciones públicas, permitiendo la existencia de información sistematizada y oportuna que apoye los procesos de toma de decisiones.

El Gobierno Electrónico organiza las TIC de la manera más conveniente para mejorar la forma de relacionarse de los cuatro actores principales (Gobierno, ciudadanos, el sector productivo y los servicios públicos), eliminando de esta forma las barreras de comunicación y fortaleciendo las relaciones y alianzas con los actores de una sociedad cada vez más interconectada y globalizada (16).

1.2 Sistemas de gestión de la información para la realización de trámites

Actualmente, existen a nivel mundial diferentes sistemas que gestionan y generan gran cantidad de información de gran utilidad para el ciudadano. Un sistema de gestión de trámites permite hacer más eficiente la información, además, ayuda a determinar cómo y dónde realizar cambios o procedimientos.

Sistema internacional de gestión de información para la realización de trámites. Sistema eRegulations

El sistema eRegulations es una herramienta electrónica concebida por la UNCTAD que ayuda a los países a mostrar, de manera transparente, los procedimientos, así como facilitar los negocios, el comercio y las inversiones (18).

Los procedimientos están presentados paso a paso, desde el punto de vista del usuario y la información se muestra con un diseño amigable donde los usuarios pueden interactuar con el mismo, y sus administradores por email o chat en línea para hacer cualquier pregunta, quejarse o proponer ideas de simplificación. Es completamente configurable mediante el sitio de administración y puede ser administrado sin conocimiento informático con pocas horas de capacitación. Los datos son registrados y actualizados en la base de datos y el sistema se instala en un servidor local, según escoja el país usuario.

Beneficios que ofrece:

- Provee total transparencia sobre las reglas y procedimientos, mostrando una descripción detallada, práctica y actualizada de los pasos que hay que seguir desde el punto de vista del usuario.
- Facilita la simplificación de los procedimientos.
- Promueve conocimiento público y estimula un diálogo público-privado sobre las reglas y procedimientos.

Sistemas nacionales de gestión de información para la realización de trámites

A nivel nacional también se han desarrollado diferentes sistemas que han ayudado a brindar información sobre la realización de trámites. Se describen brevemente aquellos que tienen relación con el propósito de esta investigación.

Portal del ciudadano cubano

El portal del ciudadano cubano es una vía cómoda y eficaz de conocer la información necesaria para iniciar un trámite. El sitio muestra la documentación que debe presentar el individuo a la hora de realizar el trámite así como, hacia dónde dirigirse y los horarios de atención. De igual forma puede conocer acerca de los servicios que brindan diferentes empresas del país y las características principales de los mismos, incluyendo sus tarifas actualizadas.

El sistema le permite al administrador:

- Publicar información relacionada con diferentes entidades agrupadas por organismos.
- Publicar y actualizar información de los servicios públicos de las entidades.
- Gestionar quejas emitidas por los usuarios.
- Identificar y brindar privilegios de acceso de los usuarios del perfil administración.

Sitio del gobierno de la República de Cuba

El sitio de gobierno de la República de Cuba muestra información directa de Cuba, su gobierno, las relaciones exteriores, así como el desarrollo económico y social del país.

Con la ayuda de un árbol de navegación dinámico, el usuario puede indagar de manera sencilla y conocer sobre los principales temas referentes a la nación. Ofrece además información sobre cada una de las provincias y municipios así como de los símbolos nacionales (20).

Cámara de comercio de la República de Cuba

La Cámara es una asociación de empresas vinculadas al comercio, la industria y los servicios, con reconocimiento ante los organismos del Estado, que permite orientar las mejores alternativas para el desarrollo de la actividad empresarial de las entidades que a esa asociación pertenecen. El sitio constituye una herramienta para la reinserción de la economía cubana en el mundo de las relaciones económicas internacionales, pues potencia e intercambia información valiosa en torno a las posibilidades de negocios a escala mundial. En él se muestra cómo promover ofertas exportables de productos y servicios, así como oportunidades de negocios e inversión de la empresa cubana y la sustitución de importaciones, en beneficio de la economía nacional (22).

Centro para la promoción del comercio exterior y la inversión extranjera de Cuba (CEPEC)

CEPEC contribuye al desarrollo del comercio exterior y la inversión extranjera en Cuba, la gestión de la información comercial, la promoción de las oportunidades de comercio e inversión extranjera y la realización de investigaciones de mercado de acuerdo con los objetivos y prioridades del país (24).

Entre los principales servicios que presta se encuentran:

- Atención a solicitudes de información.
- Servicios de internet.
- Diseminación de información.
- Información sobre proveedores extranjeros.
- Consulta y préstamo de bibliografía especializada.

- Centro de referencia de la OMC.

Conclusión referente a los sistemas informáticos estudiados:

Tabla 1: Resultado de los sistemas estudiados

Variable/Sitio	Portal del Ciudadano Cubano	Gobierno de la República de Cuba	Cámara de Comercio de la República de Cuba	CEPEC	eRegulations
Información de cómo realizar trámites.	✓	✓	✓	✓	✓
Gestión de la información.	✓	X	X	X	✓
Distribución de la información por pasos.	X	X	X	X	✓
Transparencia sobre las reglas.	✓	X	X	X	✓
Simplificación de los procedimientos.	X	X	X	X	✓
Costo de adquisición.	✓	✓	✓	✓	X

Luego del estudio realizado a los principales sistemas de información, tanto internacionales como nacionales, referente a la ejecución de un trámite, se pudo llevar a cabo un análisis de cómo se ve reflejada la información de acuerdo a las necesidades de cada ciudadano. Como se puede apreciar en la Tabla 1, el sistema eRegulations cuenta con todos los elementos necesarios para

darle solución a la problemática planteada pero no se puede hacer uso del mismo puesto que es privativo y le cuesta mucho al país.

1.3 Metodologías de desarrollo de software

Después de conceptualizar los principales elementos del negocio así como el estudio de los principales sistemas similares existentes, lo cual respalda la necesidad de crear una solución acorde a las necesidades, es importante abordar acerca de la metodología de desarrollo de software, así como las herramientas y tecnologías a emplear durante la realización del sistema.

Un proceso de software detallado y completo suele denominarse “Metodología”. Las metodologías se basan en una combinación de los modelos de proceso genéricos. Al desarrollar un buen software es necesario enfocarse en trabajar de forma organizada, donde se controle y documente todo lo que tenga relación con el proyecto en cuestión y puedan eliminarse los riesgos que podrían presentarse durante el desarrollo del mismo (26). Todo esto se logra con el empleo de una metodología eficaz, que se adapte a las características propias del software que se desarrolla.

Una metodología debería definir con precisión los artefactos, roles y actividades involucradas, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto y guías para el uso de herramientas de apoyo.

Las metodologías de desarrollo se pueden dividir en dos grandes grupos:

Metodologías pesadas: Son las más tradicionales, se centran en la definición detallada de los procesos y tareas a realizar y requiere de una extensa documentación. Son más eficaces y necesarias en cuanto al proyecto que se pretende desarrollar teniendo en cuenta el tiempo y los recursos necesarios.

- El **Proceso unificado de desarrollo de software (RUP²)** es un marco de trabajo genérico que puede especializarse para una variedad de sistemas de software, áreas de aplicación, diferentes tipos de organizaciones, niveles de aptitud, así como diferentes tamaños de proyectos. Provee un acercamiento disciplinario para asignar tareas y responsabilidades

² RUP. Del inglés: Rational Unified Process

dentro de una organización de desarrollo. Está basado en componentes interconectados a través de interfaces (28).

El ciclo de vida de RUP se caracteriza por 3 frases fundamentales: iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML (28).

Metodologías ágiles: Las metodologías ágiles son uno de los temas recientes en ingeniería de software que están acaparando gran interés. Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan (30).

- **Programación extrema (XP³)** es la más destacada de los procesos ágiles de desarrollo de software. Se diferencia de las metodologías tradicionales en que realiza más énfasis en la adaptabilidad que en la previsibilidad (32). En XP todos los requisitos son expresados como escenarios (historias de usuarios), los cuales son implementados directamente como una serie de tareas. Esta metodología resulta apropiada para proyectos con requisitos imprecisos y cambiantes, en los que existe un alto riesgo técnico. Los programadores trabajan en parejas y se desarrollan pruebas para cada una de las tareas.
- **SCRUM** es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software, realizándose de forma iterativa e incremental y priorizando el trabajo en función del valor que tenga para el negocio. Su equipo busca la creación de un software de calidad, definiendo las características que debe tener el producto a construir y remueve cualquier obstáculo que pudiera entorpecer su tarea.

Ambas metodologías buscan el mismo fin, producir software con altos estándares de calidad requeridos por los clientes, contando siempre con herramientas y métodos para su desarrollo.

En la Universidad de las Ciencias Informáticas se desarrolló la metodología ágil **SXP** en el 2008 por UNICORNIOS: un grupo de proyectos de investigación y desarrollo que formó parte del polo

³ XP. Del inglés: Extreme Programming,

productivo de Software Libre (SWL) perteneciente a la misma entidad, la cual se basa en SCRUM para la gestión eficiente de los proyectos, y en XP para la ingeniería de software (3).

Las tres metodologías se caracterizan por dividir el desarrollo de software en pequeñas iteraciones, denominadas Sprint en el caso de SCRUM Y SXP e iteración en el caso de XP; y por ser el cliente parte del equipo de desarrollo.

En el caso de RUP, este brinda numerosas ventajas, sin embargo genera muchos artefactos que podrían suponer un atraso por la reducida composición del equipo de trabajo con que se cuenta y el escaso tiempo que se dispone para la ejecución del proyecto. Además, se torna rígida a la hora de introducir cambios.

Teniendo en cuenta lo antes expuesto, haciendo referencia a las características de las metodologías XP, SCRUM y SXP, se decide utilizar SXP, como metodología de desarrollo, la cual consta de 4 fases:

- Planificación-Definición: fase en la que se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
 - ✓ Artefactos:
 - Concepción del sistema.
 - Lista de reserva del producto (LRP).
 - Lista de riesgos.
 - Historias de usuario.
 - Modelo de historia de usuario del negocio.
 - Modelo de diseño.
- Desarrollo: fase en la que se realiza la implementación del sistema hasta que esté listo para ser entregado.
 - ✓ Artefactos:
 - Tareas de ingeniería.
 - Cronograma de producción.
 - Estándar de programación.
 - Caso de prueba de aceptación.
 - Plan de liberaciones.

- Entrega: fase donde se pone en marcha el producto desarrollado y se hace la entrega al cliente.
 - ✓ Artefactos:
 - Manual de desarrollo.
 - Manual de identidad.
 - Manual de usuario.
- Mantenimiento: fase en la que se realiza el soporte para el cliente.
 - ✓ Artefactos:
 - Gestión de cambio

Esta metodología ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo (3).

1.4 Captura de requisitos

La definición de requisitos del sistema especifica qué es lo que el sistema debe hacer (sus funciones) y sus propiedades esenciales y deseables (5).

Existe un gran número de técnicas para obtener requerimientos. A continuación se describen las que serán usadas en la captura de los requisitos necesarios para la confección del sistema:

Taller: es una reunión en el que un grupo de personas interesadas y expertos en el tema trabajan juntos para definir, crear, refinar y alcanzar el cierre de entregables. Los talleres facilitan sesiones con múltiples actores y roles formales ayudando además, a la obtención de requisitos.

Observación: Por medio de esta técnica el analista obtiene información de primera mano sobre la forma en que se efectúan las actividades. Este método permite observar la forma en que se llevan a cabo los procesos. Los observadores experimentados saben qué buscar y cómo evaluar la relevancia de lo que observan (7).

Tormentas de ideas: Consiste en reuniones de cuatro a diez personas donde primeramente se sugieren toda clase de ideas sin juzgar su validez, se recopilan y se realiza un análisis detallado de cada propuesta. Esta técnica se puede utilizar para identificar un primer conjunto de requisitos en aquellos casos donde no están muy claras las necesidades que hay que cubrir (7).

1.5 Herramientas a utilizar

Para lograr el desarrollo de un producto que satisfaga las necesidades del cliente es importante tener en cuenta una serie de tendencias, tecnologías, lenguajes y herramientas indispensables para garantizar la entrega de un software competitivo. A continuación se explican cada una de las tecnologías que fueron seleccionadas para llevar a cabo la documentación e implementación de la aplicación a desarrollar.

1.5.1 Lenguaje de modelado UML

El lenguaje unificado de diagrama o notación (UML⁴) sirve para especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos. UML no es un método de desarrollo, ayuda a visualizar el diseño y a hacerlo más accesible para otros.

Los elementos UML se utilizan para crear diagramas, que representan alguna parte o punto de vista del sistema. Es por ello que su utilización es de gran importancia para el diseño del componente debido a que sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema (9).

1.5.2 Herramienta de modelado

Visual Paradigm

Es una herramienta CASE⁵. La misma propicia características de modelado de casos de uso externo, además de un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ofrece un diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad. Cuenta con un modelo y código que permanece sincronizado en todo el ciclo de desarrollo, además de la disponibilidad de múltiples versiones, múltiples plataformas y capacidad de integrarse en los principales IDEs⁶. Permite la creación de

⁴ UML. Del inglés: Unified Modeling Language™

⁵ CASE. Del inglés: Computer Aided Software Engineering

⁶ PHP. Del inglés: Hypertext Pre-processor.

diagramas mucho más rápido que cualquier herramienta en el mercado, además de la extensibilidad y diseño personalizado de apoyo (11).

Se hará uso de la herramienta Visual Paradigm en su versión. 8.0.

Axure RP Pro

Es la principal herramienta para la rápida creación de, prototipos y especificaciones para aplicaciones y sitios web. Permite a los diseñadores de aplicaciones crear diagramas de flujo, prototipos y especificaciones. Es utilizado por los analistas de negocios, arquitectos de la información y expertos en usabilidad. Axure RP se ha convertido en una herramienta esencial para el diseño de aplicaciones con éxito (13).

Se hará uso de la herramienta Axure RP Pro en su versión 6.5.

1.5.3 Marco de trabajo

Un marco de trabajo es una estructura de software compuesta por componentes personalizables e intercambiables para el desarrollo de una aplicación. Se puede considerar como una aplicación genérica incompleta y configurable a la que se añade las últimas piezas para construir una aplicación concreta. Tiene dentro de sus objetivos principales acelerar el proceso de desarrollo, reutilizar el código existente y promover buenas prácticas de desarrollo como el uso de patrones, agilizando así el desarrollo (15).

Symfony2

Symfony2 es un marco de trabajo diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web, además, proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Otra de las características que posee es que automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Está desarrollado completamente con PHP⁶ y es compatible con la mayoría de los gestores de bases

de datos, como MySQL, PostgreSQL, Oracle y SQL⁷ Server de Microsoft. Es fácil de instalar y configurar en la mayoría de las plataformas (Windows y *nix) (17).

A este marco de trabajo se le aplican diferentes patrones de diseño los cuales se agrupan en 3 categorías: creacionales, estructurales y de comportamiento (19). A continuación se describen los mismos.

Creacionales: Los patrones creacionales abstraen el proceso de creación de instancias y ocultan los detalles de cómo los objetos son creados o inicializados.

- ✓ **Factory Method** (Método de fabricación): Centraliza en una clase constructora la creación de objetos de un subtipo determinado, ocultando al usuario la manera de elegir el que desea crear.

Estructurales: Los patrones estructurales describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos.

- ✓ **Adapter** (Adaptador): Adapta una interfaz para que pueda ser empleada por una clase que de otro modo no se podría utilizar.
- ✓ **Composite** (Objeto compuesto): Permite tratar objetos compuestos.
- ✓ **Decorator** (Envoltorio): Añade funcionalidades a una clase dinámicamente.

Comportamiento: Los patrones de comportamiento ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos.

- ✓ **Iterator** (Iterador): Permite realizar recorridos sobre objetos compuestos independientemente de la implementación de estos.
- ✓ **Mediator** (Mediador): Define un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto.
- ✓ **Strategy** (Estrategia): Permite disponer de varios métodos para resolver un problema y elegir cuál utilizar en tiempo de ejecución.

⁷ SQL. Del inglés: Structured Query Language.

- ✓ **TemplateMethod** (Método plantilla): Define en una operación el esqueleto de un algoritmo, delegando en las subclases algunos de sus pasos, esto permite que las subclases redefinan ciertos pasos de un algoritmo sin cambiar su estructura.

Se hará uso del marco de trabajo Symfony2 en su versión 2.6.7.

ORM Doctrine 2

Doctrine2 es un mapeador objeto relacional (ORM⁸) para PHP 5.3.0+ que proporciona una persistencia transparente de objetos PHP. Se sitúa en la parte superior de una poderosa capa de abstracción de base de datos (DBAL⁹). Brinda a cada uno de los programadores la habilidad de concentrarse únicamente en la lógica de negocio orientada a objetos y dejar la persistencia como tarea secundaria (21).

Twig

Twig es un motor de plantillas desarrollado para el lenguaje de programación PHP y que nace con el objetivo de facilitar, a los desarrolladores de aplicaciones web que utilizan la arquitectura Modelo Vista Controlador (MVC¹⁰), la construcción de interfaces más dinámicas y extensibles. Es un sistema que resulta muy sencillo de aprender y capaz de generar plantillas con un código preciso y entendibles (23).

Bootstrap 3

Bootstrap se ha convertido en uno de los marcos de trabajo más populares y proyectos de código abierto en el mundo. Es un marco de trabajo CSS¹¹ y JavaScript de interfaz de usuario que contiene un conjunto de herramientas de software libre para el diseño de sitios y aplicaciones web (27).

⁸ ORM. Del Inglés: Object Relational Mapping

⁹ DBAL. Del inglés: DataBase Abstraction Layer

¹⁰ MVC. Del inglés: Model View Controller

¹¹ CSS. Del inglés: Cascading Style Sheets

FOSUserBundle

Añade soporte a un sistema respaldado por el usuario de base de datos en Symfony2. Proporciona un marco flexible para la gestión de usuarios que tiene como objetivo manejar tareas comunes, tales como el registro de usuarios y la recuperación de contraseña. El componente *security* de Symfony2 proporciona una plataforma de seguridad flexible que permite cargar usuarios desde la configuración a una base de datos (29).

1.5.4 Lenguajes de programación

Facilita la tarea de programación, ya que dispone de formas adecuadas que permiten ser leídas y escritas por personas y a su vez resultan independientes del modelo a utilizar. Representan de forma simbólica los códigos que podrán ser leídos por una persona.

JavaScript

Es una mezcla entre Java y HTML y se utiliza principalmente para la creación de páginas web. Es muy diferente a Java, puesto a que es orientado a objeto y no tiene herencia, está basado en prototipos con el que se pueden crear pequeños programas que luego son insertados en una página web y en programas más grandes.

Este lenguaje permite crear páginas HTML dinámicas que procesen la entrada del usuario y además que sean capaces de gestionar datos usando archivos y bases de datos relacionales (31).

Se hará uso del lenguaje JavaScript en su versión 1.5.

Procesador de hipertexto (PHP)

Es un lenguaje interpretado que funciona bajo un servidor web Apache y genera un código HTML. Crea programas que se ejecutan en un servidor web y que dan respuesta en función a las peticiones hechas desde un navegador web. Tiene la ventaja de ser un lenguaje especialmente diseñado para la programación en un entorno web, además, de una buena facilidad de aprendizaje y utilización. Al ejecutarse desde un navegador web el usuario trabaja en un entorno conocido y no tiene que aprender nada nuevo para poder utilizar el programa (33).

Se hará uso del lenguaje PHP en su versión 5.5.

1.5.5 Lenguajes de marcado

Lenguaje de marcado de hipertexto (HTML¹²)

Es un lenguaje bastante sencillo para describir hipertextos y ayuda a promover el uso generalizado de internet. La descripción se basa en especificar en el texto, la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.) así como los diferentes efectos que se quieren dar. Construyen un conjunto de reglas que definen todo aquello que es parte de un documento digital, pero que no pertenece al texto del mismo. Se trata de un lenguaje orientado a definir la estructura y la semántica de un documento (34).

Hojas de estilo en cascada (CSS)

Es usado para dar mayor personalización a las páginas web y mantener el mismo estilo en múltiples páginas. Al utilizar CSS, se pueden definir los colores, fondos, bordes, margen, alineaciones, letra y tamaño de una página web. Funciona empleando reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las ventajas que ofrece la utilización de hojas de estilo son: se diseñan de forma independiente al documento HTML, se pueden aplicar enteras o en alguno de sus niveles a los documentos que se desee, facilitando la consistencia y homogeneidad en el diseño y la imagen del sitio. Se pueden adecuar a los distintos medios de presentación de documentos. Una ventaja adicional, es que reducen el tiempo de carga de las páginas web en el ordenador, puesto que se cargan en el navegador o visualizador (35).

1.5.6 Servidor web

Un servidor web es un ordenador que está al servicio de otras máquinas llamadas clientes y que le suministran todo tipo de información. Se comunica con variados clientes, todos demandando algún tipo de información, puede ser desde archivos de texto, video, audio, imágenes, emails, aplicaciones, programas y consultas a base de datos (36).

¹² HTML. Del ingles: HyperText Markup Language

Apache

Apache es una tecnología gratuita de código fuente abierto, altamente configurable de diseño modular. Presenta características como mensajes de errores altamente configurables, bases de datos de autenticación y negociado de contenido. El servidor consta de una sección y diversos módulos que aportan muchas de la funcionalidades que podrían considerarse básicas para un servidor web (37).

Se hará uso del servidor web Apache en su versión 2.2.22.

1.5.7 Sistema gestor de bases de datos (SGBD)

Es una serie de recursos para manejar grandes volúmenes de información. Tiene como característica la independencia de los datos, es decir, que no dependen del programa y por tanto cualquier aplicación puede hacer uso de los mismos (38).

PostgreSQL

Es un SGBD con código fuente disponible libremente. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. PostgreSQL funciona correctamente con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema (39).

Se hará uso de este sistema gestor de base de datos en su versión 9.3.

1.5.8 Entorno de desarrollo integrado (IDE¹³)

Es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

¹³ IDE. Del inglés: Integrated Development Environment

Netbeans

Es un programa que sirve como IDE que permite programar en diversos lenguajes. Es un entorno de desarrollo completo, eficaz, fácil de usar y gratuito. Es ideal para trabajar con el lenguaje de desarrollo java, ofrece además, un excelente entorno para programar en PHP. Posee un equilibrado balance entre las interfaces y el editor puede autocompletar el código (40).

1.6 Conclusiones parciales

Con el estudio de los principales conceptos asociados al dominio del problema se logró una mejor comprensión del negocio. El análisis de los sistemas existentes en Cuba, determinó que los mismos a pesar de mostrar información sobre cómo realizar trámites, no poseen elementos relevantes para el desarrollo del sistema, ya que ninguno muestra los procedimientos con una total transparencia. A nivel mundial, se realizó un estudio de los sistemas eRegulations de los cuales se determinaron varios elementos que sirven guía para la solución que se desea, pero su utilización requiere antes la compra del mismo.

Para el desarrollo de esta aplicación, se realizó un estudio sobre las metodologías de desarrollo de software existentes, seleccionando la metodología ágil SXP, debido a los beneficios que brinda al combinar los elementos de SCRUM para la gestión del proyecto con las facilidades de XP para la ingeniería de software. El uso de esta metodología permitirá lograr mayor productividad, mejores resultados y la obtención de un producto que satisfaga las necesidades del cliente. Se seleccionó como herramientas al Axure RP 6.5 para la realización de prototipos web con mayor facilidad y calidad. Además, al Visual Paradigm 8.0, para ayudar a automatizar el proceso de desarrollo de software y al marco de trabajo Symfony2, para la construcción de un sistema escalable, flexible y sencillo para el usuario.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Para la construcción y desarrollo de un sistema informático, es necesario la utilización de métodos y técnicas que permitan la solución de problemas presentes a lo largo del ciclo de vida de un software. El correcto modelado del sistema y una buena captura de requisitos permiten que se eliminen las fallas que puedan aparecer durante su desarrollo.

El siguiente capítulo presenta una propuesta del sistema a desarrollar, se exponen las principales características que presenta el negocio, obteniéndose además los principales artefactos que guiarán el proceso de implementación de la solución.

2.1 Solución propuesta

Teniendo en cuenta los objetivos que persigue el presente trabajo de diploma y en correspondencia con lo planteado en el epígrafe anterior, se propone como solución un sistema informático para publicar y documentar la información referente a la realización de trámites comerciales en Cuba. Esto permitirá el total acceso a la información por parte de los usuarios.

2.1.1 Concepción inicial del sistema

En la primera fase de SXP se genera la plantilla de Concepción del sistema, en ella se refleja la visión general del producto a implementar. Se especifican además, la concepción del sistema, su objetivo, principales involucrados, así como las tecnologías usadas. Ver anexo 1.

2.2 Captura de requisitos

Para la adquisición de los requisitos se llevaron a cabo talleres con funcionarios de la Aduana General de la República de Cuba (AGR) y con especialistas del proyecto VUCEC, logrando captar en gran medida las necesidades existentes, así como un mejor entendimiento del proceso que se maneja. La tormenta de ideas es una técnica que se combinó en los talleres donde se acumularon las ideas de los participantes, anotándose y respetándose cada criterio y opinión para tener una perspectiva de las necesidades del sistema. En el caso de la técnica de observación, se realizó un estudio a diferentes sistemas especialmente al sistema eRegulations, analizando qué características se muestran durante los trámites comerciales de un país. Haciendo uso de estas técnicas, se realizó una buena captura de los requisitos funcionales del sistema. Ver anexo 3.

2.2.1 Lista de reserva del producto (LRP)

En la lista de reserva se describen los requisitos como funcionalidades que el sistema debe cumplir en su desarrollo. Esta lista puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y del cliente. Muestra de cada requisito el número identificativo, una breve descripción, la estimación puntual y los roles que realizaron la estimación. La siguiente tabla es un fragmento de la LRP, para una mejor comprensión de la misma:

Tabla 2: Fragmento de la plantilla LRP (Prioridad: Muy alta)

Prioridad				
No.	Requisito	Descripción	Estimación (semanas)	Estimado por:
Requisitos funcionales				
Muy alta				
RF#1	Autenticar usuario.	Permite autenticar con el usuario y contraseña.	1	Programador
RF#10	Añadir trámite.	Permite añadir un trámite comercial.	2	Programador
RF#19	Añadir índice.	Permite añadir un índice.	2	Programador
RF# 27	Añadir pasos.	Permite añadir un paso.	2	Programador
RF#32	Añadir bloques de información.	Permite añadir un bloque de información.	2	Programador
RF#77	Mostrar resumen del trámite.	Muestra un resume del trámite.	2	Programador
RF#78	Mostrar listado de trámites.	Muestra un listado de trámites, agrupados de manera jerárquica.	1	Programador
RF#79	Mostrar desglose de un paso.	Muestra un listado de los bloques de información de un paso.	2	Programador
RF#80	Establecer orden de pasos para un trámite.	Establece el orden de los pasos para un trámite.	1	Programador

2.3 Diseño de metáfora

La metáfora produce una descripción simple y entendida de lo que hay que construir. De esta forma, guía a los desarrolladores sin entrar en complejos documentos funcionales. Es fácil de comprender por el cliente y el equipo, y proporciona suficiente información para guiar la arquitectura del proyecto (41).

A partir de la definición de las funcionalidades descritas en la LRP es posible establecer las historias de usuarios, prototipos del sistema y las tareas ingenieriles que permiten su desarrollo, siendo estas las actividades que se realizan y se describen a continuación.

2.3.1 Historias de usuario

Es la técnica utilizada para especificar los requisitos del software. Las mismas son escritas por los clientes como las tareas que el cliente debe hacer y su construcción depende de la habilidad que tenga el cliente para definir las. Las tablas 3 y 4 representan las historias de usuario correspondiente a los requisitos Autenticar usuario y Realizar trámite respectivamente.

Tabla 3: HU_1: Autenticar de usuario

Historia de Usuario	
Número: HU_1	Nombre historia de usuario: Autenticar usuario.
Modificación de historia de usuario: Ninguna.	
Usuario: Araí Nieves Delgado.	Iteración asignada: 1
Prioridad en negocio: Muy Alta.	Puntos estimados: 1
Riesgo en desarrollo: Alto.	Puntos reales: 1
<p>Descripción: El usuario accede al sistema insertando un nombre y una contraseña, el sistema comprobará que estos datos son correctos y se le dará el acceso a las funcionalidades a las cuales tiene permiso.</p> <p>En el caso de que el usuario inserte los datos incorrectamente, el sistema mostrará un mensaje informando la falla.</p>	
Observaciones:	

Prototipo de interface

Autenticación

Usuario:

Contraseña:

Aceptar

✓ **RF# 1:** Autenticar usuario

Tabla 4: HU_3: Realizar trámite

Historia de Usuario	
Número: HU_3	Nombre historia de usuario: Realizar trámite
Modificación de historia de usuario: Ninguna.	
Usuario: Araí Nieves Delgado.	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 3 días
Riesgo en desarrollo: Medio	Puntos reales: 3 días
Descripción: Permite añadir, modificar, eliminar, mostrar y buscar trámite, así como establecer orden de los pasos y asignar pasos a un trámite.	
Observaciones: Se muestra un listado con todos los trámites existentes, especificando de cada uno el título y la descripción. Para realizar las operaciones de modificar y eliminar es necesario señalar el trámite y luego especificar qué acción ejecutar. En el caso que se desee añadir un trámite, se muestra la interfaz para introducir los datos necesario: título, descripción. Una vez finalizada la acción se da clic en el botón Adicionar. Para establecer el orden de los pasos de cada trámite, es necesario que cada uno esté registrado en el sistema. Es válido señalar que dos pasos del mismo trámite no deben poseer el mismo orden. Para asignar un nuevo paso al trámite se debe seleccionar del listado de pasos registrados	

en el sistema.

Prototipo de interface

- ✓ **RF# 4** Mostrar trámite

Añadir	Modificar	Eliminar	Asignar Pasos	Buscar
Título		Descripción		
<input type="checkbox"/>				
<input checked="" type="checkbox"/>				
<input checked="" type="checkbox"/>				
<input checked="" type="checkbox"/>				

- ✓ **RF# 1:** Añadir trámite

Añadir trámite
✕

Título:

Descripción:

+

Pasos

Paso 1

Paso 2

▶

◀

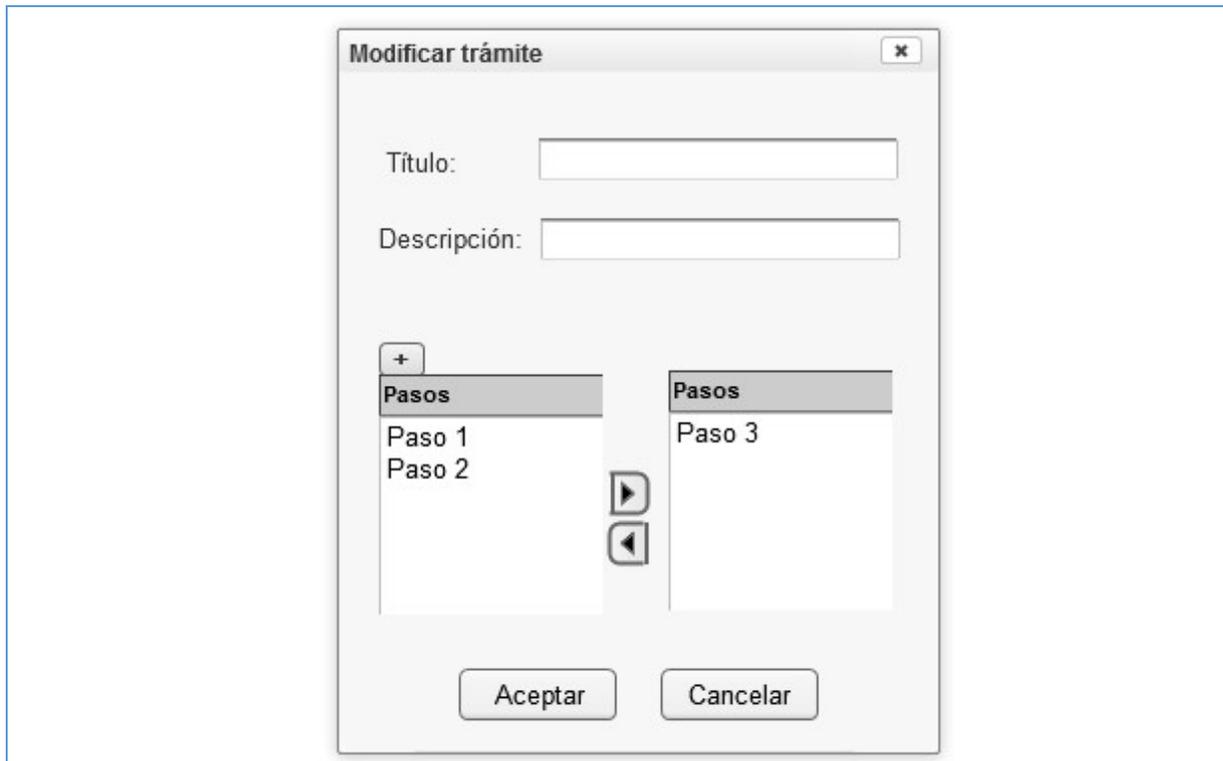
Pasos

Paso 3

Aceptar

Cancelar

- ✓ **RF# 2:** Modificar trámite



.2.3.1 Tareas de ingeniería

Partiendo de las historias de usuarios, se hace necesario la definición de un grupo de tareas de ingeniería para cada una de ellas, sentando las bases para el posterior desarrollo de la solución.

Algunas de estas tareas están relacionadas con todas las historias de usuario, preparando el ambiente de implementación, estas son:

Tabla 5: T_1: Generación de la base de datos

Tareas de la ingeniería	
Número tarea: T_1	Número historia de usuario: Todas.
Nombre tarea: Generación de la base de datos.	
Tipo tarea: Desarrollo.	Puntos estimados: 1.
Fecha inicio: 2 – abril - 2015	Fecha fin: 8- abril- 2015
Programador responsable: Araí Nieves Delgado	
Descripción: Se genera la base de datos a partir del diseño realizado, esta debe	

implementarse sobre el sistema gestor definido para el desarrollo de la aplicación.

Tabla 6: T_2 Montaje del ambiente de desarrollo con la integración de los marcos de trabajo seleccionados para el desarrollo.

Tareas de la ingeniería	
Número tarea: T_2	Número historia de usuario: Todas.
Nombre tarea: Montaje del ambiente de desarrollo con la integración de los marcos de trabajo seleccionados para la implementación.	
Tipo tarea: Desarrollo.	Puntos estimados: 1.
Fecha inicio: 3 – marzo - 2015	Fecha fin: 1- abril- 2015
Programador responsable: Araí Nieves Delgado	
Descripción: Instalar y configurar para el desarrollo de la aplicación.	

Tabla 7: T_3: Realizar el Mapeo de la Base de Datos

Tareas de la ingeniería	
Número tarea: T_3	Número historia de usuario: Todas.
Nombre tarea: Realizar el Mapeo de la Base de Datos.	
Tipo tarea: Desarrollo.	Puntos estimados: 1.
Fecha inicio: 3 – marzo - 2015	Fecha fin: 1- abril- 2015
Programador responsable: Araí Nieves Delgado	
Descripción: Realizar el mapeo de la base de datos.	

Otras tareas de ingeniería están asociadas a cada historia de usuario para poder dar cumplimiento con la implementación de las funcionalidades definidas en ellas, por ejemplo, las relacionadas con la HU_3 son:

Tabla 8: T_10: Diseñar las interfaces requeridas para la funcionalidad: Crear trámite

Tareas de la ingeniería	
Número tarea: T_10	Número historia de usuario: HU_3
Nombre tarea: Diseñar las interfaces requeridas para la funcionalidad: Crear trámite.	
Tipo tarea: Desarrollo.	Puntos estimados: 1.
Fecha inicio: 25 – marzo - 2015	Fecha fin: 27- marzo- 2015
Programador responsable: Araí Nieves Delgado	
Descripción: Realizar el diseño de las interfaces para la funcionalidad Crear trámite.	

Tabla 9: T_11: Implementar funcionalidad: Crear trámite

Tareas de la ingeniería	
Número tarea: T_11	Número historia de usuario: HU_3
Nombre tarea: Implementar funcionalidad: Crear trámite.	
Tipo tarea: Desarrollo.	Puntos estimados: 1.
Fecha inicio: 7– abril - 2015	Fecha fin: 9- abril- 2015
Programador responsable: Araí Nieves Delgado	
Descripción: Realizar la implementación para la funcionalidad Crear trámite.	

2.4 Arquitectura seleccionada

La arquitectura de software representa la estructuración del sistema, la cual se crea en etapas tempranas del desarrollo. Esta organización representa un diseño de alto nivel del sistema que tiene dos propósitos primarios: satisfacer los atributos de calidad (desempeño, seguridad y modificabilidad), y servir como guía en el desarrollo.

El desarrollo de la solución propuesta está basada en una arquitectura en capas con el uso del patrón arquitectónico Modelo Vista Controlador (MVC), separando los conceptos para que el desarrollo esté estructurado de una mejor manera. Esto facilita la programación en diferentes capas de manera paralela e independiente.

Symfony2, según su creador Fabien Potencier, basa su funcionamiento interno en este patrón, proporcionando herramientas para la parte del Controlador y de la Vista. La capa del Modelo es

incorporada mediante el uso de DoctrineBundle, el cual integra la librería Doctrine al marco de trabajo. El siguiente esquema muestra de manera simplificada el funcionamiento de este patrón en Symfony2. (42)

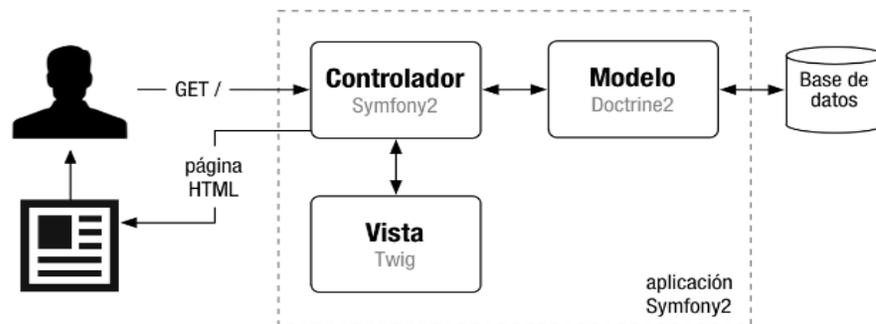


Figura 1: Patrón MVC en Symfony2 (42)

Modelo: Es la representación de la información que maneja el sistema. Son los datos que proveen información al usuario de la aplicación. Esto se evidencia en las clases entidades, las que manejan la información del objeto y en ocasiones representan una tabla en la base de datos. En el sistema estas clases se encuentran en “src/AppBundle/Entity/” como se muestra en la figura 2.

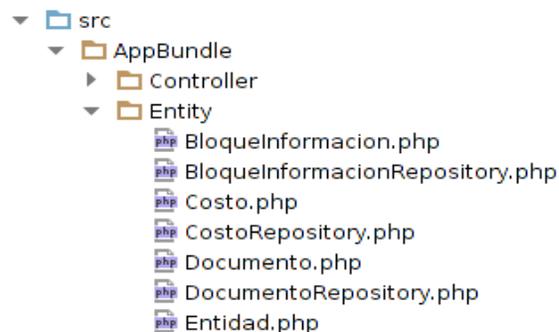


Figura 2: Clases entidades

Vista: Son las páginas HTML con contenido dinámico sobre el cual el usuario puede realizar operaciones. El marco de trabajo implementa las vistas usando el motor de plantilla Twig, dichas vistas se encuentran en “src/AppBundle/Resources/views/”, como se muestra en la figura 3.

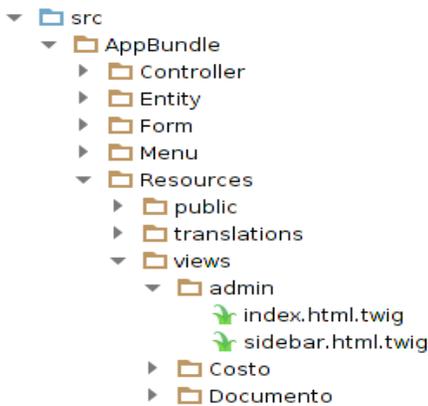


Figura 3: Plantillas de las vistas

Controlador: Capa encargada de manejar las solicitudes del usuario, procesando la información y modificando el modelo en caso de ser necesario. Es el enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de la aplicación. (43). En el sistema las clases controladoras se encuentran en “src/AppBundle/Controller”, como se muestra en la figura 4.

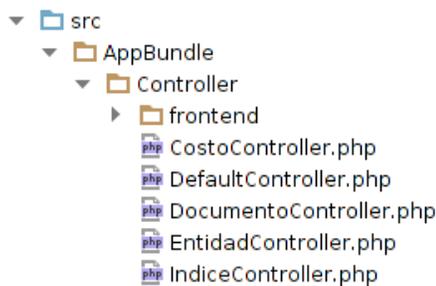


Figura 4: Clases controladoras

2.5 Diagrama de clases

El propósito del diagrama de clases es representar las diferentes clases que se utilizan en el sistema, así como las relaciones que existen entre ellas. A continuación se muestra el diagrama de clases de la solución propuesta.

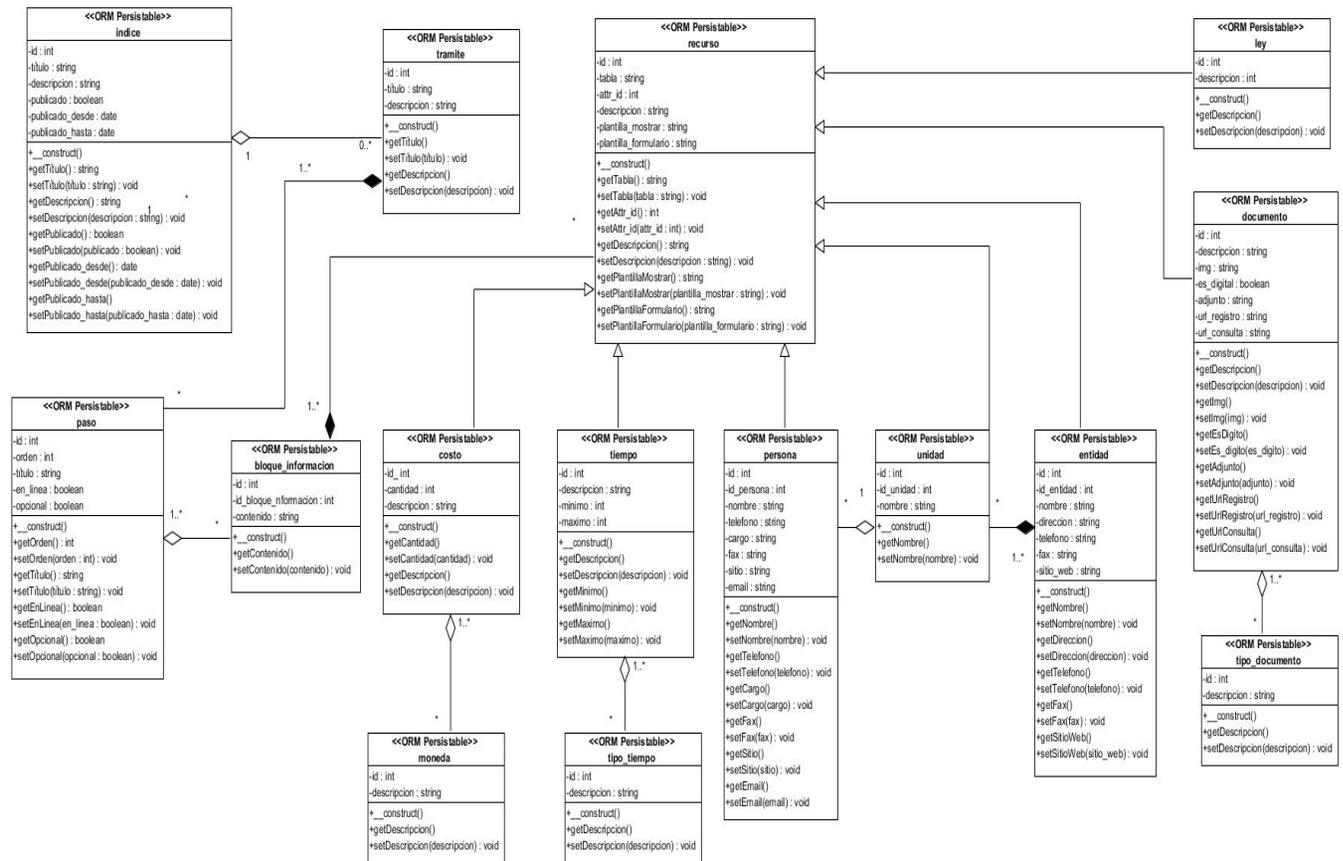


Figura 5: Diagrama de clases

2.6 Modelo de datos

Dispone de un diagrama que ayuda a entender los datos y como se relacionan entre ellos. Consiste en un conjunto de objetos básicos llamados entidades y relaciones entre estos ellos, implementándose de forma gráfica a través del Diagrama Entidad relación (44).

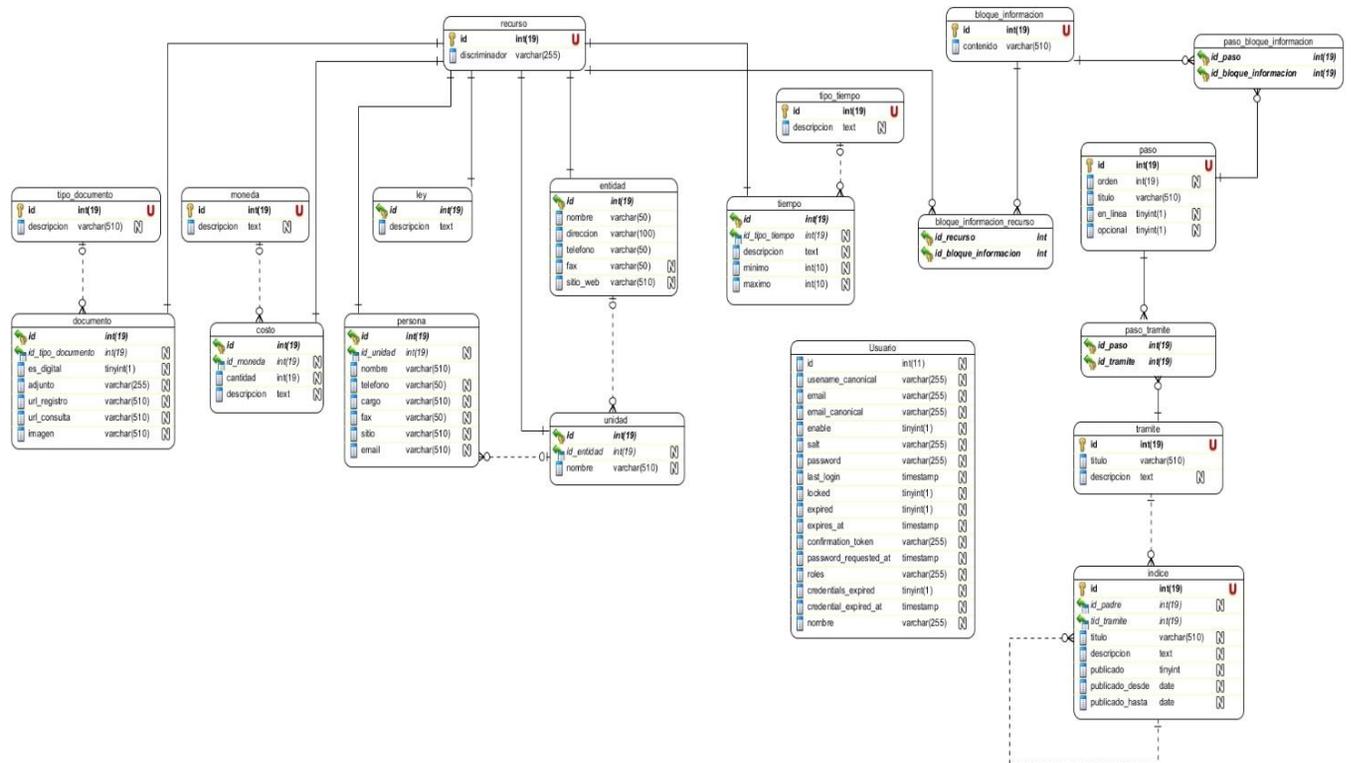


Figura 6: Modelo de Datos

2.7 Patrones de diseño

Un patrón es una descripción de un problema y la solución, a la que se le da un nombre y que se puede aplicar a nuevos contextos. Con sus características ayuda a obtener un software de alta calidad (45).

2.7.1 Patrones de diseño de la base de datos

En el diseño de la base de datos se emplearon los siguientes patrones:

Llaves subrogadas: el uso de este patrón es bastante generalizado pues la mayoría de las entidades contienen una llave única, lo que facilita reducir el costo de las búsquedas de la base de datos. Permite además, la protección antes los cambios debido a que la lógica del negocio no está en las llaves y evita a contención (bloqueo) ya que los mecanismos de generación secuencial son rápidos.

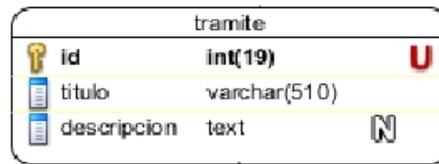


Figura 7: Uso del patrón Llaves subrogadas

Control de acceso basado en roles (RBAC¹⁴): este patrón se implementa mediante la asignación de roles a los diferentes usuarios del sitio, a través de los cuales dichos usuarios obtienen los permisos para realizar determinadas funciones en el mismo; lo que simplifica la gestión de los usuarios debido a que el manejo de los permisos se reduce al asignarle el rol apropiado.

2.7.2 Patrones de Diseño de Asignación de Responsabilidades o GRASP¹⁵

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

Estos patrones se describen a continuación:

- ✓ **Experto:** este patrón es aplicado en todas las clases debido a que cada una de ellas es experta pues contienen la información necesaria para cumplir con las responsabilidades que le fueron asignadas. Permite conservar el encapsulamiento ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece tener sistemas más robustos y de fácil mantenimiento (46).

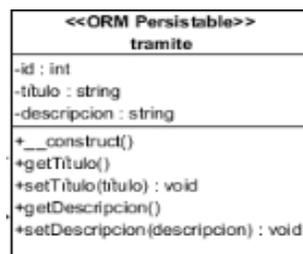


Figura 8: Ejemplo del patrón Experto en la clase trámite

¹⁴ RBAC. Del inglés: Role-based access control.

¹⁵ GRASP. Del inglés: General responsibility assignment software patterns.

- ✓ **Creador:** guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El patrón se evidencia en las clases controladoras que, para cada una de las funcionalidades de la aplicación, son las encargadas de crear las instancias de los objetos que manejan, favoreciendo así la reutilización y el bajo acoplamiento (46).

```
public function newAction()
{
    $entity = new Tramite();
    $form = $this->createCreateForm($entity);

    return array(
        'entity' => $entity,
        'form' => $form->createView(),
    );
}
```

Figura 9: Evidencia del patrón Creador en la clase TramiteController.php

- ✓ **Bajo acoplamiento:** El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, lo que trae como beneficio que no se afecten los cambios de otros componentes, la facilidad de entendimiento, así como una fácil reutilización. En el diseño se proporciona un bajo acoplamiento debido a que las clases existentes tienen asignadas responsabilidades de tal forma que estas no dependan en gran medida de otras (46).

```
<?php
namespace AppBundle\Entity;

use AppBundle\AppBundle;
use Doctrine\ORM\Mapping as ORM;
use Symfony\Component\Validator\Constraints as Assert;
use Gedmo\Mapping\Annotation\Uploadable;

/**
 * Entidad
 *
 * @ORM\Table(name="entidad")
 * @ORM\Entity
 */
class Entidad extends Recurso
{
    const DEFAULT_ADMIN_PAGE_SIZE=10;

    /**
```

Figura 10: Evidencia del patrón Bajo acoplamiento en la clase: Entidad

- ✓ **Alta cohesión:** La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. El patrón se evidencia en el diseño debido a que, a cada una

de las clases se asignaron responsabilidades de tal forma, que estén estrechamente relacionadas entre sí y no realicen un trabajo excesivo (46).

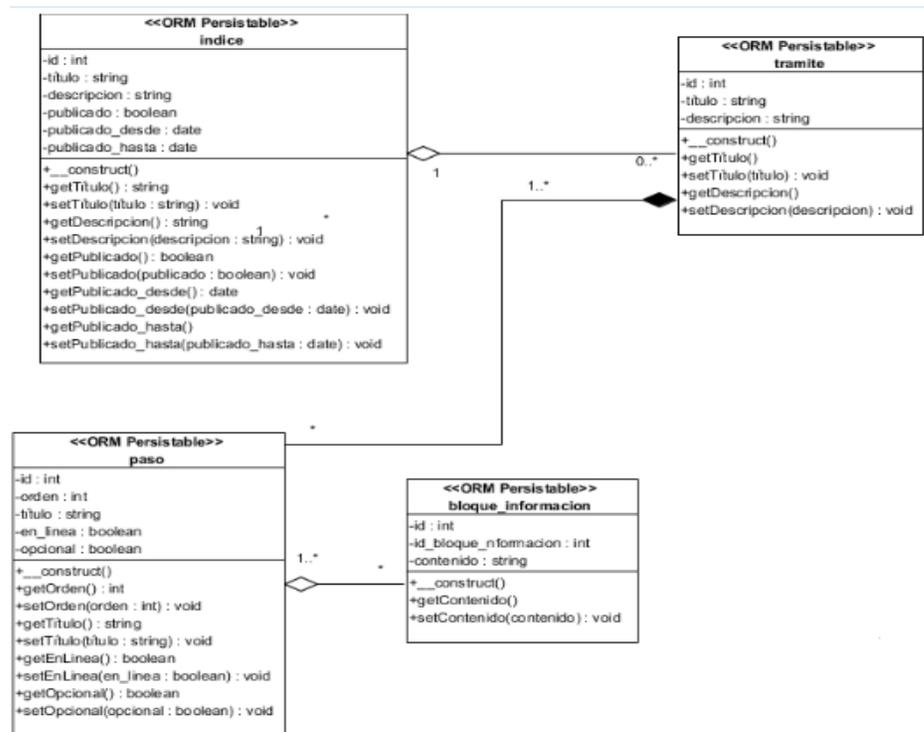


Figura 11: Evidencia del patrón Alta Cohesión en las clases

2.7.3 Patrones de Comportamiento (Patrones GOF¹⁶)

En la presente solución no se define el uso de un nuevo patrón de comportamiento, debido a que se hará uso de los patrones de GOF que utiliza el marco de trabajo Symfony los cuales son descritos en el Capítulo 1.

¹⁶ GOF. Del inglés: Gang of Four.

2.8 Conclusiones parciales

En el presente capítulo se realizó la descripción del sistema mediante el uso de la metodología seleccionada.

Se detallaron las características fundamentales de la propuesta de solución, logrando una mayor claridad de las necesidades del cliente y aplicando una correcta captura de requisitos. Además se generó el artefacto LRP, en el cual se agruparon los RF según la estimación de su complejidad por parte del desarrollador.

Se realizó un estudio sobre todas las HU definidas, lo que se concretó en las plantillas de historias de usuario, en las que se especificaron: la descripción, junto al prototipo y los requisitos que responden a las mismas. Luego fueron definidas las Tareas de la ingeniería para así sentar las bases para las restantes fases de la metodología.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

En la realización de este capítulo se muestran los resultados obtenidos a través de la aplicación de las métricas y técnicas empleadas para validar los requisitos y el diseño del sistema; además de los resultados alcanzados luego de la realización de las pruebas de funcionalidad y las unitarias. Se generan los principales artefactos definidos por la metodología, correspondientes a la fase de Desarrollo, en sus dos flujos de trabajo (Implementación y Prueba).

3.1 Técnicas de validación de requisitos

Con el objetivo de ratificar que los requisitos de software obtenidos definen el sistema que el cliente desea, se llevó a cabo un proceso de validación de los mismos, para el cual se emplearon las siguientes técnicas:

- **Construcción de prototipos:** mediante los prototipos se mostró al cliente un modelo ejecutable del sistema que permitió tener una visión preliminar de cómo sería el sistema. A través de la interacción con los mismos se comprobó si satisfacía o no las necesidades, Los cuales fueron aprobados.
- **Generación de casos de prueba:** como parte del proceso de validación de los requisitos funcionales del sistema, fueron diseñados casos de pruebas para cada historia de usuario.

3.2 Validación del diseño de la aplicación web

Para comprobar la calidad y fiabilidad del diseño del sistema, se emplearon las métricas basadas en clases: Tamaño Operacional de Clase (TOC) y Relaciones entre Clases (RC), permitiendo validar el diseño propuesto en la investigación. La aplicación detallada de las métricas se encuentra recogida en el resultado obtenido. A continuación se muestra un resumen del mismo.

3.2.1 Tamaño Operacional de Clases (TOC)

La métrica TOC permite medir la responsabilidad, la complejidad de implementación y la reutilización de las clases del diseño. Es importante destacar que para esta métrica la responsabilidad y la complejidad son inversamente proporcionales a la reutilización, por lo que a mayor responsabilidad y complejidad de implementación de una clase, menor será su nivel de reutilización. El tamaño

operacional de una clase se puede determinar empleando medidas para saber el número total de operaciones que contiene (47).

A continuación se muestra el resultado de la aplicación de la métrica TOC en las 17 clases que juegan un papel fundamental en los procesos principales del sistema informático.

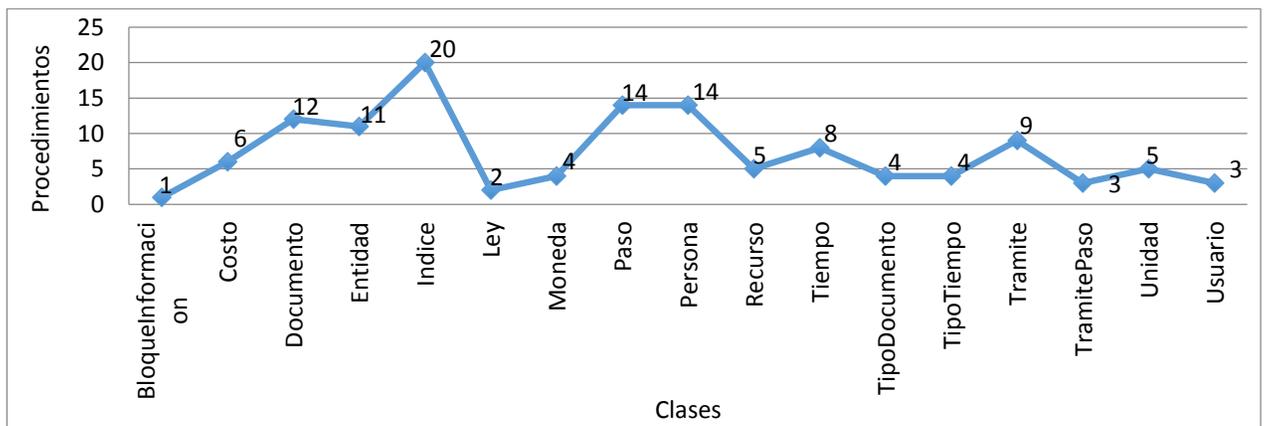


Figura 12: Cantidad de procedimientos por clases

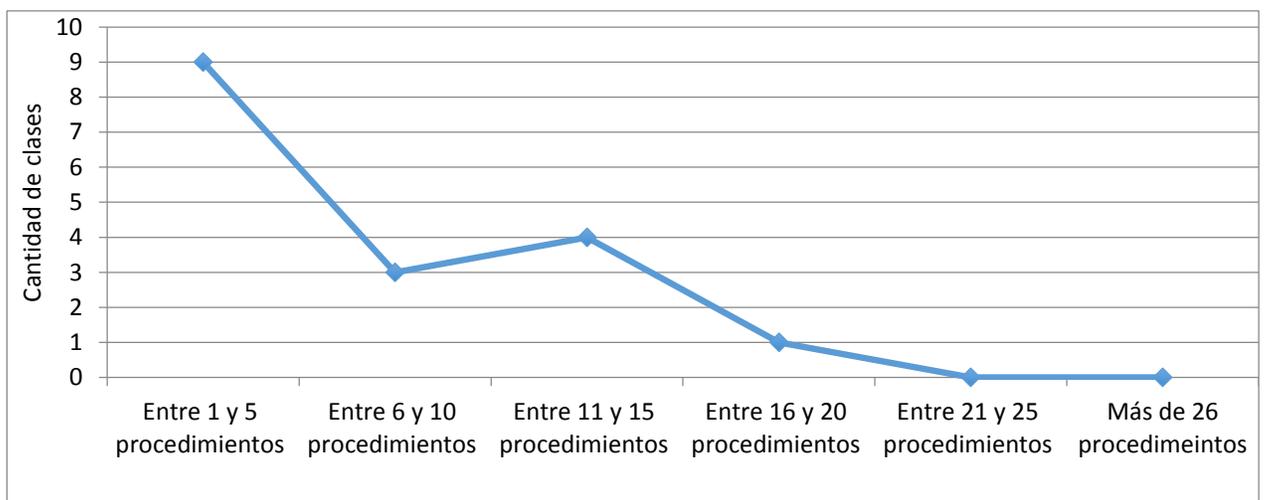


Figura 13: Resumen de cantidad de procedimiento por clases

Aplicando como umbral un promedio de 7 procedimientos (125 procedimientos en total / 17 clases existentes), se obtienen los valores de los atributos de calidad: responsabilidad de las clases, complejidad al implementar las mismas, así como sus niveles de reutilización.

Tabla 10: Valores de las variables de calidad: Responsabilidad, Complejidad y Reutilización

Nivel	Responsabilidad		Complejidad		Reutilización	
	Cantidad de clases	Promedio	Cantidad de clases	Promedio	Cantidad de clases	Promedio
Baja	12	71	12	71	0	0
Media	5	29	5	29	5	29
Alta	0	0	0	0	12	71



Figura 14: Resultado de las variables: Responsabilidad, Complejidad y Reutilización

Luego de aplicada la métrica TOC se observa que las clases del diseño de la aplicación web no se encuentran sobrecargada en cuanto a responsabilidades y el nivel de complejidad de las mismas no es muy alto, lo que favorece en gran medida la reutilización de estas.

3.2.2 Relaciones entre Clases (RC)

La métrica RC permite evaluar el acoplamiento, la complejidad de mantenimiento, la reutilización y la cantidad de pruebas de unidad necesarias para probar una clase teniendo en cuenta las relaciones existentes entre ellas (47). Luego de aplicar la métrica RC se arrojaron los siguientes resultados:

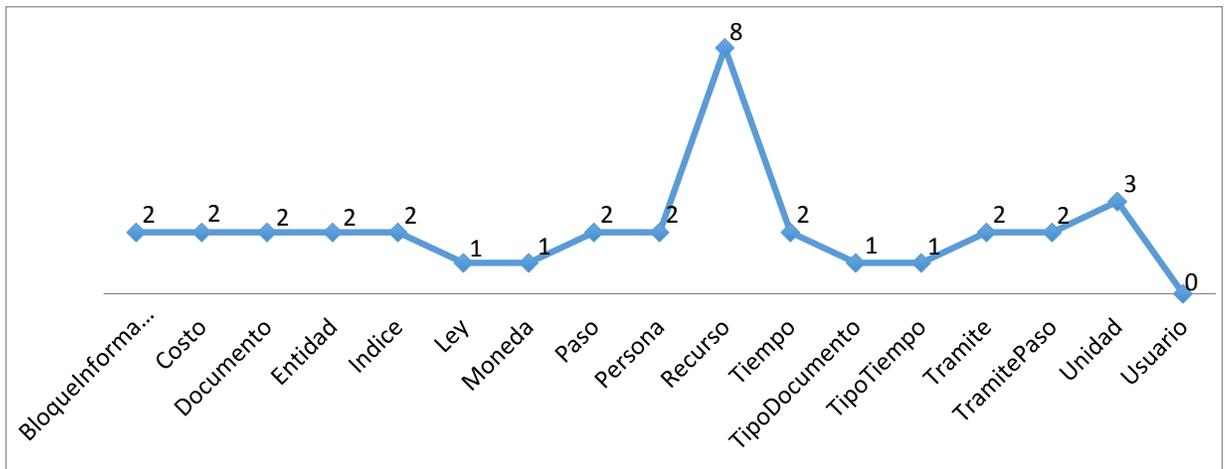


Figura 15: Cantidad de relaciones de uso por clases

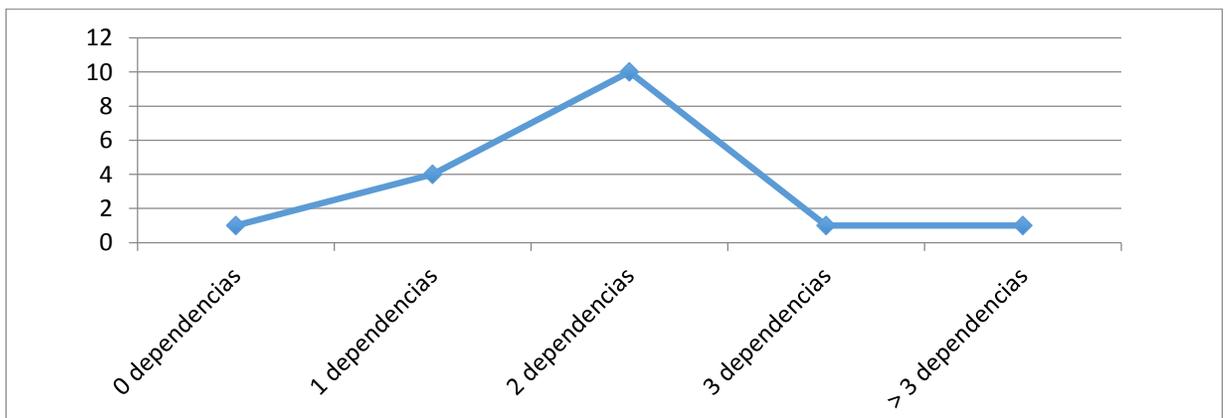


Figura 16: Resumen de relaciones de uso

Luego de calcular los valores de las variables de calidad: acoplamiento, complejidad de mantenimiento, reutilización y cantidad de pruebas, así como un umbral promedio de 2 asociaciones de uso por clase, se obtiene los siguientes resultados:

Tabla 11: Valores de la variable: Acoplamiento, Complejidad, Reutilización y Cantidad de pruebas

Nivel	Acoplamiento		Complejidad		Reutilización		Cant. Pruebas	
	Cantidad de clases	Promedio						
Ninguno	1	6	-	-	-	-	-	-
Baja	14	82	16	94	1	6	16	94
Media	1	6	0	0	0	0	0	0
Alta	1	6	1	6	16	94	1	6

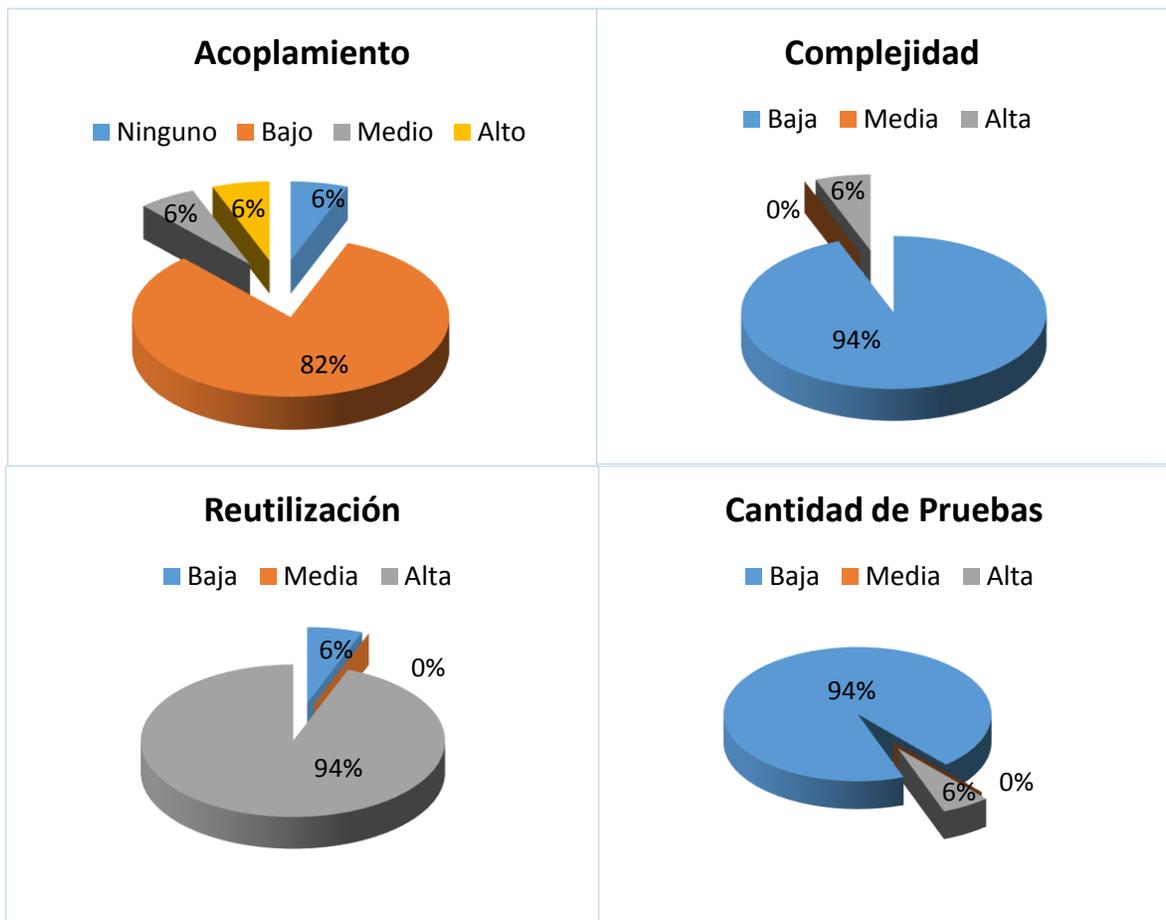


Figura 17: Resultado de las variables: Acoplamiento, Complejidad, Reutilización y Cantidad de pruebas

Una vez aplicada la métrica y teniendo en cuenta el umbral definido para validar el diseño, se obtiene como resultado que las clases promueven el bajo acoplamiento, la complejidad de mantenimiento y la cantidad de pruebas no son altas. En consecuencia el grado de reutilización es mayor.

En sentido general, los resultados obtenidos de la aplicación de las métricas TOC y RC demuestran que el diseño no es complejo, que las clases presentan bajo acoplamiento y un alto grado de reutilización.

3.3 Pruebas

Las pruebas son un instrumento adecuado para verificar el estado de la calidad de un producto, en las cuales un sistema o componente es ejecutado bajo condiciones o requisitos especificados (48). Una de las principales fortalezas de la metodología de desarrollo SXP es el proceso de pruebas, el cual permite asegurar el éxito del producto al realizarse de manera continua, proporcionando la obtención de un producto de mayor calidad, ya que los errores son detectados en un corto plazo de tiempo y se corrigen de una manera más sencilla.

3.3.1 Pruebas de funcionalidad

Las pruebas de caja negra, es el propio cliente quien las realiza en compañía de uno de los representantes del equipo de desarrollo y se orientan a las funcionalidades del sistema. Su objetivo es comprobar, desde la perspectiva del usuario final, el cumplimiento de las especificaciones definidas para el producto. Una HU solo se considerará terminada cuando haya pasado correctamente todas las pruebas de aceptación.

A continuación se muestran las pruebas de aceptación correspondiente a las HU_1 e HU_3:

Tabla 12: A-1-1 correspondiente a la HU_1: Autenticar usuario

Caso de prueba de aceptación	
Código caso de prueba: A-1-1	Nombre historia de usuario: Autenticar usuario
Nombre de la persona que realiza la prueba: Araí Nieves Delgado	
Descripción de la prueba: Se ejecuta la prueba y se verifica que el usuario se autentique correctamente.	
Condiciones de ejecución: La página de autenticación de la aplicación debe estar abierta.	
Entrada / pasos de ejecución:	
<ul style="list-style-type: none"> 1- Introducir los datos en los campos (usuario y contraseña) 2- Clic en el botón Autenticar 	
Resultado esperado: El usuario accede al sistema y se activan las funcionalidades correspondientes,	
Evaluación de la prueba:	

Tabla 13: A-3-1 correspondiente a la HU_3: Crear trámite

Caso de prueba de aceptación	
Código caso de prueba: A-3-1	Nombre historia de usuario: Crear trámite
Nombre de la persona que realiza la prueba: Araí Nieves Delgado	
Descripción de la prueba: Se realizará la prueba a la funcionalidad Crear trámite.	
Condiciones de ejecución: El usuario debe:	
<ul style="list-style-type: none"> - Estar autenticado. - Ser administrador del sitio. -Estar situado en la interfaz correspondiente a Gestionar Trámite. 	
Entrada / pasos de ejecución:	
Acción:	Resultado esperado:
Se selecciona la opción "Añadir	El sistema debe mostrar un formulario solicitando los

trámite”.	datos necesarios para añadir trámite
Se introducen los datos necesarios para añadir el trámite y se da clic en el botón “Añadir”	<p>El sistema verifica que estén llenos todos los campos requeridos.</p> <p>Si existen datos incorrectos o campos vacíos el sistema le muestra un mensaje informando cuál es error.</p> <p>Si los datos son correctos el sistema guarda el nuevo trámite, se actualiza el listado de los mismos y se le notifica al usuario el resultado de la operación.</p>
Evaluación de la prueba:	

Para comprobar que las funcionalidades del sistema se realizan correctamente y responden a las necesidades del cliente, se llevó a cabo un total de 3 iteraciones, realizadas por el Ingeniero Yordanis Cruz Segura, Jefe del Departamento de Aplicaciones Financieras y Aduanales, del centro CEIGE, teniendo en cuenta los siguientes parámetros: ortografía, redacción, validación, funcionalidad y recomendación, como se muestra en la figura 18.

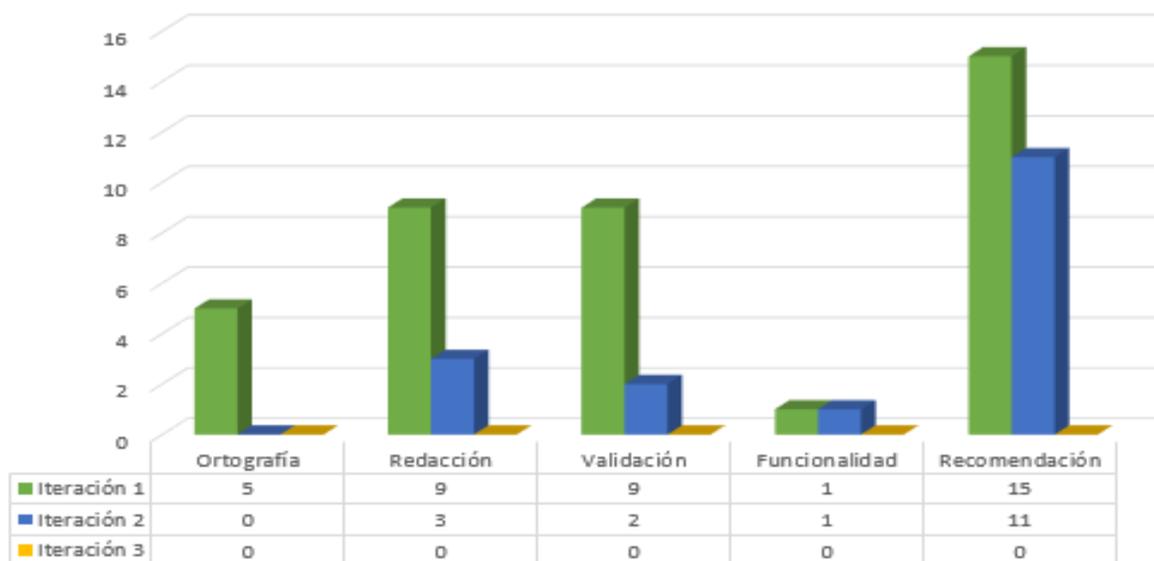


Figura 18: No conformidades detectadas a través de las pruebas de Caja Negra

En una primera iteración se detectaron un total de 39 no conformidades, predominando entre ellas las de tipo recomendación. En la segunda iteración se obtuvieron 17 no conformidades donde aún existían errores de validación, realizándose además varias recomendaciones. Finalmente, en una tercera iteración se obtuvieron resultados satisfactorios debido a que no se encontraron no conformidades.

3.3.2 Pruebas unitarias

Para comprobar que cada sentencia de código se ejecuta al menos una vez, se realizaron pruebas al código de las funcionalidades más complejas desde el punto de vista de la programación. Para ello se empleó el método de caja blanca con la técnica del Camino Básico; a partir de la obtención de la complejidad de un procedimiento y la obtención de un conjunto básico de caminos, se realizaron los casos de prueba (49).

Seguidamente se muestra el proceso de pruebas realizado a la funcionalidad `updateAction()` perteneciente a la controladora `TramiteController`.

```
public function updateAction(Request $request, $id)
{
    $em = $this->getDoctrine()->getManager();//1

    $entity = $em->getRepository('AppBundle:Tramite')->find($id);//1

    if (!$entity) {//2
        throw $this->createNotFoundException('Unable to find Tramite entity.');//3
    }

    $deleteForm = $this->createDeleteForm($id);//4
    $editForm = $this->createEditForm($entity);//4
    $editForm->handleRequest($request);//4

    if ($editForm->isValid()) {//5
        $em->flush();//6

        return $this->redirect($this->generateUrl('admin_tramite_show', array('id' => $id)));//6
    }

    return array(
        'entity' => $entity,
        'edit_form' => $editForm->createView(),
        'delete_form' => $deleteForm->createView(),
    );//7
}
```

Figura 19: Funcionalidad `updateAction()`

Para obtener los casos de prueba a partir de la técnica seleccionada se debe construir el grafo de flujo correspondiente al código de la funcionalidad. Luego se determina la complejidad ciclomática $V(G)$ del grafo resultante, la cual es un indicador del número de caminos independientes que existen en un grafo, es decir, es cualquier camino dentro del código que introduce por lo menos un nuevo conjunto de sentencias de proceso o una nueva condición (49).

La complejidad se puede calcular de tres formas:

- 1- El número de regiones del grafo del flujo
- 2- Aristas – Nodos + 2
- 3- Nodos predicado + 1 (Un nodo predicado es el que representa una condicional if o case, es decir, salen varios caminos)

Realizando los cálculos correspondientes se obtiene por cualquiera de las variantes el siguiente resultado: $V(G) = 3$.

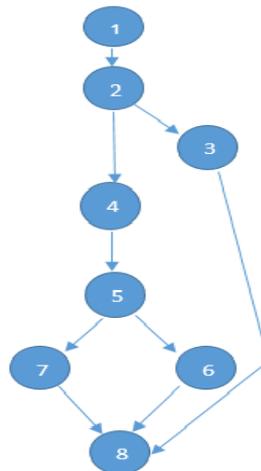


Figura 20: Grafo del flujo del código de la función updateAction

Por lo que el conjunto de caminos básico sería:

Camino Básico 1: 1-2-3-8

Camino Básico 2: 1-2-4-5-7-8

Camino Básico 3: 1-2-4-5-6-8

Luego se definen los casos de prueba para cada uno de los caminos básicos obtenidos. A continuación se muestra el resultado de las pruebas aplicadas.

Tabla 14: Caso de prueba para el camino básico 1

Descripción: Se verificará que el sistema no permita modificar un trámite que no existe en la base de datos.		
Condición de ejecución:	Entrada:	Resultados esperados:
El trámite a modificar no debe existir en la base de datos de la aplicación.	Se alteran los datos del formulario modificar del trámite cambiando el id, el cual es un campo oculto.	El sistema debe mostrar un mensaje notificando que se debe seleccionar un trámite para poder modificar.
Resultado:		

Tabla 15: Caso de prueba para el camino básico 2

Descripción: Se verificará que se realice la modificación de un trámite de forma correcta.		
Condición de ejecución:	Entrada:	Resultados esperados:
El trámite debe existir en la base de datos de la aplicación y los datos introducidos por el usuario deben ser válidos.	El identificador del trámite y los datos introducidos	El sistema debe modificar el trámite.
Resultado:		

Tabla 16: Caso de prueba para el Camino básico 3

Descripción: Se verificará que se realice la modificación de un trámite de forma correcta.		
Condición de ejecución:	Entrada:	Resultados esperados:
El trámite debe existir en la base de datos de la aplicación y los datos introducidos por el usuario contienen errores.	El identificador del trámite y los datos introducidos	El sistema debe mostrar la interfaz para modificar, señalando los errores

Resultado:		

De forma general se le realizaron las pruebas de caja blanca a 6 funcionalidades, seleccionadas a partir de su complejidad de implementación. Se realizaron dos iteraciones, las cuales arrojaron los siguientes resultados: en una primera iteración se detectaron 12 no conformidades de forma general, las cuales fueron corregidas en su totalidad. Por otro lado, una segunda iteración arrojó resultados satisfactorios con cero no conformidades.

Con los resultados obtenidos luego de realizar las pruebas de Caja Negra y Caja Blanca, se demuestra que el sistema es estable y que su funcionamiento interno y externo es el correcto.

3.4 Impacto de la solución propuesta

El sistema desarrollado constituye la solución de las principales deficiencias presentadas en la problemática respecto a la realización de un trámite, así como la distribución de la información paso a paso y una correcta gestión de la misma.

Actualmente, la persona interesada en realizar una operación en el país, debe dirigirse al MINCEX para que le oriente los lugares a los que debe asistir, debido a que no existe un documento rector que explique paso a paso dicho proceso. El individuo debe presentarse físicamente con la persona requerida y ésta a su vez, facilitar los documentos necesarios en formato duro. En el caso de que la persona o entidad no se sienta satisfecha, puede auxiliarse de las regulaciones y leyes existentes o realizar su búsqueda haciendo uso de las TIC.

La solución propuesta brinda un conjunto de facilidades al usuario, tales como: la búsqueda fácil y rápida de contenidos relacionados a los procedimientos para la realización de trámites comerciales, así como visualizarlos de una manera sencilla y amigable para el usuario. También posee varias estrategias de navegabilidad como la existencia de migas de pan, así como la accesibilidad a través del logo a la página de inicio del sistema.

A partir de la realización de talleres, en conjunto con los funcionarios de la AGR y con especialistas del proyecto VUCEC, se evaluó su satisfacción con respecto al uso del sistema, arrojando como

resultado que satisface las necesidades, logrando un mejor acceso a los procedimientos para realizar trámites comerciales en Cuba. Teniendo en cuenta la forma en que se realiza actualmente la búsqueda de dicha información y el modo en que se gestiona en la propuesta desarrollada se generó el Acta de Aceptación del producto que refleja la conformidad del cliente.

3.5 Conclusiones parciales

La aplicación de técnicas de validación de requisitos permitió ratificar que los obtenidos estaban en correspondencia con las solicitudes del cliente. Por su parte, la validación del diseño mediante las métricas TOC y RC permitió obtener, de forma general, el grado de complejidad de implementación y mantenimiento, de responsabilidad, reutilización, acoplamiento y la cantidad de pruebas necesarias para realizar a las clases. Favoreció además, la creación de un diseño lo más sencillo posible, de fácil mantenimiento e implementación y que promueve la reutilización. Además, las pruebas de aceptación permitieron comprobar que las funciones son operativas a través de la interfaz del software, que la entrada se acepta de forma adecuada y se produce un resultado correcto, manteniendo así la integridad de la información externa.

CONCLUSIONES

«Sistema para documentar y publicar los procedimientos para realizar trámites comerciales en Cuba»

CONCLUSIONES

Con la culminación del presente trabajo se obtuvo el diseño e implementación del sistema para publicar y actualizar la información referente a los procedimientos de las entidades y organismos del país que intervienen en las operaciones del comercio exterior. Los resultados obtenidos durante la investigación arrojaron las conclusiones siguientes:

- ✓ El estudio sobre los principales sistemas de gestión de la información para la realización de trámites en Cuba, demostró que no arrojaron elementos relevantes para el desarrollo del sistema, puesto a que ninguno brinda información necesaria sobre cómo realizar los procedimientos para desarrollar un trámite comercial.
- ✓ La fundamentación teórica realizada en la investigación permitió justificar la selección de la metodología de desarrollo, los patrones de diseño, de arquitectura y las herramientas a utilizar para el desarrollo del sistema.
- ✓ Las pruebas realizadas arrojaron resultados satisfactorios, demostrando la calidad del sistema desarrollado, además fue validado con el uso de métricas y técnicas aplicables al desarrollo de la solución.
- ✓ El sistema desarrollado cumple satisfactoriamente los requisitos definidos por el cliente y posee una interfaz amigable e intuitiva

RECOMENDACIONES

«Sistema para documentar y publicar los procedimientos para realizar trámites comerciales en Cuba»

RECOMENDACIONES

Una vez cumplidos los objetivos de la presente investigación se recomienda:

- ✓ Establecer un mecanismo de retroalimentación que permita a los usuarios emitir propuestas de simplificación de los trámites.
- ✓ Extender el sistema para que permita agregar contenidos informativos de manera dinámica, con el objetivo de brindar información oportuna a los usuarios.

REFERENCIAS BIBLIOGRÁFICAS

«Sistema para documentar y publicar los procedimientos para realizar trámites comerciales en Cuba»

REFERENCIAS BIBLIOGRÁFICAS

1. DEFINICIÓNABC. Definición de Comercio Exterior » Concepto en Definición ABC. [online]. [Accessed 1 June 2015]. Available from: <http://www.definicionabc.com/economia/comercio-exterior.php>
2. *Panorama Digital de los Trámites de Comercio Exterior en América Latina y el Caribe. Propuestas de acciones a nivel regional.* [online]. Santa Fé de Bogotá, Colombia, 2010. Available from: http://sela.org/attach/258/EDOCs/SRed/2010/04/T023600004015-0-Panorama_Digital_-_Tramites_de_Comercio_Exterior.pdf Encuentro Regional Latinoamericano y del Caribe sobre Ventanillas Únicas de Comercio Exterior.
3. GLADYS MARSÍ PEÑALVER ROMERO. *MA-GMPR-UR2 Metodología ágil para proyectos de software libre.* Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas : Universidad de las Ciencias Informáticas, 2008.
4. CUBAINDUSTRIA. Comercio Exterior. [online]. 23 January 2014. [Accessed 1 June 2015]. Available from: <http://www.cubaindustria.cu/index.php/comercio-exterior/estructura-del-comercio-exterior>
5. *Sommerville_Parte_I_Vision_General.pdf* [online]. [Accessed 1 June 2015]. Available from: http://eva.uci.cu/file.php/161/Documentos/Materiales_basicos/Materiales_basicos_de_la_Unidad_1/Sommerville/Sommerville_Parte_I_Vision_General.pdf
6. UNCTAD. unctad.org | About Us. [online]. 2012. [Accessed 1 June 2015]. Available from: <http://unctad.org/es/Paginas/AboutUs.aspx>
7. CESAR ARTURO GUERRA. Obtención de Requerimientos. Técnicas y Estrategia | SG. [online]. October 2007. [Accessed 1 June 2015]. Available from: <http://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategia>
8. Conferencia de las Naciones Unidas sobre Comercio y Desarrollo - EcuRed. [online]. [Accessed 5 June 2015]. Available from: http://www.ecured.cu/index.php/Conferencia_de_las_Naciones_Unidas_sobre_Comercio_y_Desarrollo
9. Capítulo 2. Introducción a UML. [online]. [Accessed 1 June 2015]. Available from: <https://docs.kde.org/stable4/es/kdesdk/umbrello/uml-basics.html>
10. *Informe sobre el comercio y desarrollo, 2014* [online]. Nueva York y Ginebra, 2014. Available from: http://unctad.org/es/PublicationsLibrary/tdr2014_es.pdf Informe se la secretaría de la Conferencia de las Naciones Unidad sobre Comercio y Desarrollo.
11. Visual Paradigm for UML Modeler Edition – Freecode. [online]. [Accessed 1 June 2015]. Available from: <http://freecode.com/projects/vpuml-me>

REFERENCIAS BIBLIOGRÁFICAS

«Sistema para documentar y publicar los procedimientos para realizar trámites comerciales en Cuba»

12. Ficha_Mincex. [online]. [Accessed 1 June 2015]. Available from: <http://www.cubagob.cu/gobierno/fichas/fmincex.htm>
13. Interactive Wireframe Software & Mockup Tool | Axure. [online]. [Accessed 6 June 2015]. Available from: <http://www.axure.com/>
14. CARMEN LIENG MENA LOMBILLO. Aniversario del INDER y del MINCEX. [online]. 23 February 2014. [Accessed 1 June 2015]. Available from: <http://www.ariguanaboradioweb.icrt.cu/index.php/comentarios/7666-aniversario-del-inder-y-del-mincex>
15. *Microsoft Word - Framework.doc - Framework.pdf* [online]. [Accessed 6 June 2015]. Available from: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
16. Gobierno Electrónico. [online]. [Accessed 5 June 2015]. Available from: <http://www.gobiernoelectronico.gob.ec/>
17. 1.1. Symfony en pocas palabras (Symfony 1.4, la guía definitiva). [online]. [Accessed 6 June 2015]. Available from: http://librosweb.es/libro/symfony_1_4/capitulo_1/symfony_en_pocas_palabras.html
18. Business Facilitation Program. [online]. [Accessed 1 June 2015]. Available from: <http://www.businessfacilitation.org/>
19. *Identifying Design Patterns in the Symfony Framework*. 2014. Istanbul, Turkey.
20. SITIO DEL GOBIERNO Y LA REPÚBLICA DE CUBA. Sitio del Gobierno de la República de Cuba. [online]. [Accessed 1 June 2015]. Available from: <http://www.cubagob.cu/>
21. Home — Doctrine Project. [online]. [Accessed 6 June 2015]. Available from: <http://www.doctrine-project.org/>
22. Cámara de Comercio. [online]. [Accessed 1 June 2015]. Available from: <http://www.camaracuba.cu/index.php/es/>
23. TWIG. Twig - The flexible, fast, and secure PHP template engine. [online]. [Accessed 6 June 2015]. Available from: <http://twig.sensiolabs.org/>
24. CEPEC. CEPEC. [online]. [Accessed 1 June 2015]. Available from: <http://www.cepec.cu/es/servicioscepec>
25. *WHITEPAPER - twig-plantillas-wp-acens.pdf* [online]. [Accessed 6 June 2015]. Available from: <http://www.acens.com/wp-content/images/2014/06/twig-plantillas-wp-acens.pdf>
26. FRANK JIBAJA. Metodologías para el desarrollo de software | Frank Jibaja Ramírez - Academia.edu. [online]. [Accessed 1 June 2015]. Available from: http://www.academia.edu/9953322/Metodologias_para_el_desarrollo_de_software

REFERENCIAS BIBLIOGRÁFICAS

«Sistema para documentar y publicar los procedimientos para realizar trámites comerciales en Cuba»

27. Bootstrap · The world's most popular mobile-first and responsive front-end framework. [online]. [Accessed 6 June 2015]. Available from: <http://getbootstrap.com/>
28. El Proceso Unificado de Desarrollo de Software (RUP). [online]. [Accessed 1 June 2015]. Available from: <http://yaqui.mxl.uabc.mx/~molguin/as/RUP.htm>
29. Introducción a FOSUserBundle — Documentación FOSUserBundle. [online]. [Accessed 6 June 2015]. Available from: <http://sf2-es.net16.net/paquetes/fosuserbundle/index.html>
30. PATRICIO LETELIER and MARÍA DEL CARMEN PENADÉS. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). .
31. Lenguaje de Programación JavaScript. [online]. [Accessed 1 June 2015]. Available from: <http://www.larevistainformatica.com/JavaScript.htm>
32. ROBERTH G. FIGUEROA, CAMILO J. SOLÍS and ARMANDO A. CABRERA. METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES. .
33. ¿Que es PHP, cómo surgió y para qué se utiliza? [online]. [Accessed 1 June 2015]. Available from: <http://www.nociondigital.com/webmasters/php-tutorial-que-es-php-como-surgio-y-para-que-se-utiliza-detalle-191.html>
34. MARÍA JESÚS LAMARCA LAPUENTE. Lenguajes hipertextuales. [online]. [Accessed 1 June 2015]. Available from: http://www.hipertexto.info/documentos/lenguajes_h.htm
35. MARÍA JESÚS LAMARCA LAPUENTE. Hojas de Estilo (CSS). [online]. [Accessed 1 June 2015]. Available from: <http://www.hipertexto.info/documentos/css.htm>
36. MANUEL SIERRA. Qué es un servidor y cuáles son los principales tipos de servidores (proxy,dns, web,ftp,pop3 y smtp, dhcp...). [online]. [Accessed 1 June 2015]. Available from: http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-smtp-dhcp&catid=57:herramientas-informaticas&Itemid=179
37. CIBERAULA. Una Introducción a Apache. [online]. [Accessed 1 June 2015]. Available from: http://linux.ciberaula.com/articulo/linux_apache_intro/
38. RAFAEL MENÉNDEZ-BARZANALLANA ASENSIO. Sistemas de Bases de Datos - Definición y Características de un SBD - Wikilearning. [online]. [Accessed 1 June 2015]. Available from: http://www.wikilearning.com/curso_gratis/sistemas_de_bases_de_datos-definicion_y_caracteristicas_de_un_sbd/3621-1
39. postgresql. [online]. [Accessed 6 June 2015]. Available from: <http://www.postgresql.org.es/>
40. *untitled - P06_M2109_02151.pdf* [online]. [Accessed 1 June 2015]. Available from: http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02151.pdf

REFERENCIAS BIBLIOGRÁFICAS

«Sistema para documentar y publicar los procedimientos para realizar trámites comerciales en Cuba»

41. XP - Extreme Programming Ingeniería de Software. [online]. [Accessed 1 June 2015]. Available from: http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html
42. *Desarrollo web ágil con Symfony2 - desarrollo-agil-symfony-2.1.pdf* [online]. [Accessed 8 June 2015]. Available from: <http://sunshine.prod.uci.cu/gridfs/sunshine/books/desarrollo-agil-symfony-2.1.pdf>
43. Patrón Modelo Vista Controlador - EcuRed. [online]. [Accessed 6 June 2015]. Available from: http://www.ecured.cu/index.php/Patr%C3%B3n_Modelo_Vista_Controlador
44. PEDRO GUTIÉRREZ. Fundamento de las bases de datos: Modelo entidad-relación. [online]. 5 November 2013. [Accessed 1 June 2015]. Available from: <http://www.genbetadev.com/bases-de-datos/fundamento-de-las-bases-de-datos-modelo-entidad-relacion>
45. Patrones Grasp | El Mundo Informático. [online]. [Accessed 6 June 2015]. Available from: <https://jorgesaavedra.wordpress.com/category/patrones-grasp/>
46. MARCELLO VISCONTI and HERNÁN ASTUDILLO. *Fundamentos de Ingeniería de Software*.
47. Métrica de diseño - EcuRed. [online]. [Accessed 3 June 2015]. Available from: http://www.ecured.cu/index.php/M%C3%A9trica_de_dise%C3%B1o
48. EDUARDO ZALAZAR MARTÍNEZ. Propuesta de Procedimiento para realizar pruebas de Caja Blanca a las aplicaciones que se desarrollan en lenguaje Python | eGov UFSC. [online]. [Accessed 6 June 2015]. Available from: <http://www.egov.ufsc.br/portal/conteudo/propuesta-de-procedimiento-para-realizar-pruebas-de-caja-blanca-las-aplicaciones-que-se-des>
49. Pruebas de caja blanca - EcuRed. [online]. [Accessed 5 June 2015]. Available from: http://www.ecured.cu/index.php/Pruebas_de_caja_blanca

ANEXOS

Anexo 1 Plantilla de Concepción Inicial del Sistema

1. Polo productivo

Centro de Informatización de Entidades (CEIGE)

2. Clasificación del proyecto

Desarrollo de aplicación web

3. Tipo de proyecto

Nacional

4. ¿Qué es?

Sistema que permitirá a los funcionarios del MINCEX de Cuba registrar y actualizar los datos referentes a los trámites establecidos para la realización de las operaciones de importación, exportación y tránsito de mercancías. El mismo permitirá una fácil consulta por parte de los interesados quienes podrán encontrar datos como: organismos que intervienen en los trámites, documentos necesarios, costos aproximados, leyes que avalan los pasos establecidos en los procesos además de otros datos que agilizan la realización de las operaciones comerciales.

5. Metodología a utilizar

SXP es la metodología a utilizar, esta consiste en la programación rápida y extrema. Es la unión de las metodologías XP y SCRUM y se centra en que con la utilización de SCRUM para la gestión se logra una correcta planificación y organización, mientras que XP respalda el proceso de desarrollo. Para el logro del éxito del producto SXP se tiene como particularidad tener como parte del equipo al usuario final.

6. Involucrados

Maylevis Morejón Valdés, Eddie Nelson Beltrán González, Luis Manuel Betancourt Oliva.

7. Roles

Rol	Responsabilidad	Nombre
Líder del Proyecto	Asegurar que el proyecto se esté llevando a cabo de acuerdo con las prácticas y que todo funciones.	Maylevis Morejón Valdés
Gerente	Es el responsable de tomar las decisiones finales. Participa en la definición de los objetivos y requerimientos, así como en la selección de los miembros del Equipo del Proyecto. Controla el progreso del software y trabaja junto con el Líder del proyecto,	Maylevis Morejón Valdés
Cliente	Contribuye a definir las historias de usuario y los casos de prueba de aceptación, para validar su implementación. Asigna la prioridad a las historias de usuario y decide cuál implementar en cada iteración.	Maylevis Morejón Valdés, Eddie Nelson Beltrán González, Luis Manuel Betancourt Oliva
Programador	Define las tareas de ingeniería y produce el código del sistema. Además selecciona el estándar de programación a utilizar.	Araí Nieves Delgado
Analista	Escribe la concepción del sistema y las historias de usuario. Asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración.	Araí Nieves Delgado
Diseñadores	Encargado del diseño del sistema, así como de los prototipos de	Araí Nieves Delgado

	interfaces. Responsables del diseño de metáforas y supervisan el proceso de construcción.	
Encargado de pruebas	Escribe los casos de pruebas de aceptación. Ejecuta las pruebas regularmente y es el responsable de las herramientas de soporte para pruebas.	Araí Nieves Delgado
Arquitecto	Se vincula directamente con el analista y el diseñador debido a que su trabajo tiene que ver con la estructura y diseño del sistema.	Araí Nieves Delgado
Consultor	Miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Aporta ideas y experiencias para el beneficio del sistema	Yorlen Guirado Más, Leonardo Darell Antúnez Naranjo

8. Misión

Brindar una aplicación web que permita al MINCEX una fácil gestión y publicación de los procedimientos para la realización de trámites comerciales y a su vez sirva de guía a las personas jurídicas o naturales interesadas en consultar dicha información, manteniendo en todo momento la transparencia de los procedimientos.

9. Alcance

El desarrollo de las funcionalidades necesarias para registrar y publicar información referente a los procedimientos para la realización de trámites comerciales en Cuba.

Anexo 2 Requisitos no funcionales

Requisitos No Funcionales		
Tipo	No	Descripción
Usabilidad	Usuario Final	RNF#1 El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora.
	Aplicación Informática	RNF#2 El software tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario, lo que le permitirá un avance considerable en la explotación de la aplicación en todas sus funcionalidades.
Ambiente	RNF#3	El sistema debe presentar una interfaz de usuario fácil de entender y usar.
	RNF#4	El sistema se desarrollará con tecnología PHP versión 5.4.
	RNF#5	Se empleará como Gestor de Base de Datos Postgres en su versión 9.1.
	RNF#6	Se empleará como Servidor de Aplicaciones Web Apache en su versión 2.2.22
	RNF#7	Como Entorno de Desarrollo Integrado (IDE) se empleará NetBeans en su versión 7.2.
	RNF#8	Como marco de trabajo se empleará Symfony2 en su versión 2.3.7.
	RNF#9	Las computadoras de los clientes solo requerirán de un navegador Internet Explorer o Firefox en sus versiones 6 y 16.0.2, respectivamente o superior.
	RNF#10	El idioma de todas las interfaces de la aplicación será el español.
	RNF#11	El sistema mostrará el nombre del usuario que está autenticado en el sistema.
	RNF#12	El sistema contará con un menú que permite acceder

		a todas las funcionalidades.
	RNF#13	El sistema no permite la entrada de datos incorrectos.
	RNF#14	Se mostrarán con un * los campos obligatorios.
Portabilidad	RNF#15	El sistema debe ser multiplataforma, haciendo énfasis en Linux y Windows.
Soporte	RNF#16	La aplicación contará antes de su puesta en marcha con un período de pruebas, se le dará mantenimiento, configuración y se brindará el servicio de instalación.

Anexo 3: Lista de reserva del producto

Prioridad				
Ítems *	Requisito	Descripción	Estimación (semanas)	Estimado por:
Requisitos Funcionales				
Muy Alta				
RF#67	Autenticar Usuario.	Permite autenticar un usuario en la aplicación con su usuario y contraseña.	1	Programador
RF#68	Añadir Usuario	Permite añadir un usuario al sistema	1	Programador
RF#73	Asignar rol	Permite asignar rol a un usuario en específico.	1	Programador
RF#1	Añadir Trámite.	Permite añadir un trámite comercial.	2	Programador
RF#11	Añadir Índice.	Permite añadir un índice.	2	Programador
RF# 6	Añadir Pasos.	Permite añadir un paso.	2	Programador
RF#16	Añadir Bloques de información.	Permite añadir un bloque de información.	2	Programador
RF#77	Mostrar resumen del trámite.	Muestra un resume del trámite.	2	Programador

RF#78	Mostrar listado de trámites.	Muestra un listado, agrupado de manera jerárquica, de los trámites.	1	Programador
RF#79	Mostrar desglose de un paso.	Muestra un listado de los bloques de información de un paso.	2	Programador
RF#80	Establecer orden de pasos para un trámite	Establece el orden de los pasos para un trámite.	1	Programador
RF#81	Asignar pasos a un trámite	Permite asignar un paso a un trámite.	2	Programador
Alta				
RF#2	Modificar Trámite.	Permite modificar los datos de los trámites existentes en el sistema.	2	Programador
RF#4	Mostrar Trámite.	Permite mostrar los datos de los trámites existentes en el sistema.	2	Programador
RF#5	Buscar Trámite.	Permite buscar todo lo relacionado con un trámite.	1	Programador
RF#69	Modificar usuario	Permite modificar los datos del usuario existente en el sistema.	1	Programador
RF#72	Buscar usuario	Permite buscar todo lo relacionado con un usuario.	1	Programador
RF#75	Bloquear rol	Permite bloquear un rol q haya sido asignado a un usuario.	1	Programador
RF#76	Desbloquear rol	Permite desbloquear un rol q haya sido asignado a un usuario.	1	Programador
RF#12	Modificar Índice.	Permite modificar los datos de los índices existentes en el sistema.	2	Programador
RF#14	Mostrar Índice.	Permite mostrar los datos de los índices existentes en el sistema.	1	Programador
RF#14	Buscar Índice.	Permite buscar todo lo relacionado con un índice.	1	Programador
RF#7	Modificar Pasos.	Permite modificar los datos de los	2	Programador

		pasos existentes en el sistema.		
RF#9	Mostrar Pasos.	Permite mostrar los datos de los pasos existentes en el sistema.	1	Programador
RF#10	Buscar Pasos.	Permite buscar todo lo relacionado con un paso.	1	Programador
RF#17	Modificar Bloques de información.	Permite modificar los datos de los bloques de información existentes en el sistema.	2	Programador
RF#82	Listar Bloques de información.	Permite listar los bloques de información existentes en el sistema.	2	Programador
Media				
RF#70	Eliminar Usuario.	Permite eliminar un usuario del sistema.	1	Programador
RF#71	Mostrar Usuario.	Permite mostrar un usuario en sistema.	1	Programador
RF#3	Eliminar Trámite.	Permite eliminar un trámite del sistema.	1	Programador
RF#13	Eliminar Índice.	Permite eliminar un índice del sistema.	1	Programador
RF#8	Eliminar Paso.	Permite eliminar un paso del sistema.	1	Programador
RF#18	Eliminar Bloque de información.	Permite eliminar un bloque de información del sistema.	1	Programador
RF#31	Añadir Persona.	Permite añadir una persona en un bloque de información.	1	Programador
RF#34	Mostrar Persona.	Permite mostrar los datos de la persona. existente en el sistema.	1	Programador
RF#33	Eliminar Persona.	Permite eliminar una persona de un bloque de información.	1	Programador

RF#21	Añadir Entidad.	Permite añadir una entidad en un bloque de información	1	Programador
RF#24	Mostrar Entidad.	Permite mostrar los datos de la entidad existente en el sistema.	1	Programador
RF#23	Eliminar Entidad.	Permite eliminar una entidad de un bloque de información.	1	Programador
RF#26	Añadir Unidad.	Permite añadir una unidad a un bloque de información.	1	Programador
RF#29	Mostrar Unidad.	Permite mostrar los datos de la unidad existente en el sistema.	1	Programador
RF#28	Eliminar Unidad.	Permite eliminar una unidad de un bloque de información.	1	Programador
RF#41	Añadir Documento	Permite añadir un documento a un bloque de información.	1	Programador
RF#44	Mostrar Documento	Permite mostrar los datos del documento existente en el sistema.	1	Programador
RF#43	Eliminar Documento	Permite eliminar un documento de un bloque de información.	1	Programador
RF#36	Añadir tipo de documento	Permite añadir un tipo de documento a un bloque de información.	1	Programador
RF#39	Mostrar tipo de documento	Permite mostrar los datos del tipo de documento existente en el sistema.	1	Programador
RF#38	Eliminar tipo de documento	Permite eliminar un tipo de documento de un bloque de información.	1	Programador
RF#56	Añadir ley	Permite añadir una ley a un bloque de información.	1	Programador
RF#58	Eliminar ley	Permite eliminar una ley de un bloque de información.	1	Programador

RF#59	Mostrar ley	Permite mostrar los datos de la ley existente en el sistema.	1	Programador
RF#61	Añadir tiempo	Permite añadir un tiempo a un bloque de información.	1	Programador
RF#64	Mostrar tiempo	Permite mostrar los datos del recurso tiempo	1	Programador
RF#63	Eliminar tiempo	Permite eliminar un tiempo de un bloque de información.	1	Programador
RF#51	Añadir costo	Permite añadir un costo a un bloque de información.	1	Programador
RF#54	Mostrar costo	Permite mostrar los datos del costo existente en el sistema.	1	Programador
RF#53	Eliminar costo	Permite eliminar un costo de un bloque de información.	1	Programador
RF#46	Añadir moneda	Permite añadir una moneda a un bloque de información.	1	Programador
RF#49	Mostrar moneda	Permite mostrar los datos de la moneda existente en el sistema.	1	Programador
RF#48	Eliminar moneda	Permite eliminar una moneda de un bloque de información.	1	Programador
RF#48	Eliminar rol	Permite eliminar un rol a un usuario	1	Programador
RF#83	Asignar bloques de información	Permite asignar un bloque de información a un paso.	1	Programador
RF#19	Mostrar bloques de información	Permite mostrar los datos de bloque de información existente en el sistema.	1	Programador
Baja				
RF#32	Modificar Persona.	Permite modificar los datos de una persona existente en un bloque de	1	Programador

		información.		
RF#35	Buscar Persona.	Permite buscar una persona existente en un bloque de información.	1	Programador
RF#22	Modificar Entidad.	Permite modificar los datos de una entidad existente en un bloque de información.	1	Programador
RF#25	Buscar Entidad.	Permite buscar los datos de una entidad existente en un bloque de información.	1	Programador
RF#27	Modificar Unidad.	Permite modificar los datos de una unidad existente en un bloque de información.	1	Programador
RF#30	Buscar Unidad.	Permite buscar los datos de una unidad existente en un bloque de información.	1	Programador
RF#57	Modificar ley.	Permite modificar los datos de una ley existente en un bloque de información.	1	Programador
RF#60	Buscar ley.	Permite buscar los datos de una ley existente en un bloque de información.	1	Programador
RF#47	Modificar moneda.	Permite modificar los datos de una moneda existente en un bloque de información.	1	Programador
RF#50	Buscar moneda.	Permite buscar los datos de una moneda existente en un bloque de información.	1	Programador
RF#52	Modificar costo.	Permite modificar los datos de un costo existente en un bloque de información.	1	Programador

RF#55	Buscar costo.	Permite buscar los datos de un costo existente en un bloque de información.	1	Programador
RF#62	Modificar tiempo.	Permite modificar los datos de un tiempo existente en un bloque de información.	1	Programador
RF#65	Buscar tiempo.	Permite buscar los datos de un tiempo existente en un bloque de información.	1	Programador
RF#37	Modificar tipo de documento	Permite modificar los datos de un tipo de documento existente en un bloque de información.	1	Programador
RF#40	Buscar tipo de documento	Permite buscar los datos de un documento existente en un bloque de información.	1	Programador
RF#49	Modificar Documento	Permite modificar los datos de un documento existente en un bloque de información.	1	Programador
RF#51	Buscar Documento	Permite buscar los datos de un documento existente en un bloque de información.	1	Programador