

Universidad de las Ciencias Informáticas

Facultad 4



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

Título:

Módulo Arrendamiento de contenedores para el Sistema de
Gestión Integral del Agente Transitario de Cargas

Autores:

Rafael León Calero
Marcos Adrián Díaz Silva

Tutores:

Ing. Leannys Rodriguez Moreno
Ing. Irina Ivis Santiesteban Pérez
Ing. Amet Martinez Verdecia

La Habana, Cuba.
Julio de 2016



No basta tener buen ingenio; lo principal es aplicarlo bien.

René Descartes

Declaración de Autoría

Declaramos ser los únicos autores de este trabajo y concedemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste se firma la presente a los ____ días del mes de _____ del año 2016.

Firma del autor

Rafael León Calero

Firma del autor

Marcos Adrián Díaz Silva

Firma del tutor

Ing. Irina Ivis Santiesteban Pérez

Firma del tutor

Ing. Leannys Rodriguez Moreno

Firma del tutor

Ing. Amet Martinez Verdecia

Dedicatoria

Este logro se los dedico a mi madre, a mi padre, a mis hermanas, a mis abuelas y abuelos por ser las personas más importantes que tengo en esta vida, por no dudar ni un segundo de lo que soy capaz.

De Marcos

A mi mamá por ser lo más grande de este mundo y siempre estar ahí en todas las decisiones de mi vida buenas o malas. Mi papá por estar presente siempre cuando lo necesite y a mis amigos por su apoyo y consejo.

De Rafael

Agradecimientos

Agradecer a todo aquel que de una forma u otra tuvo que ver en que hoy me convierta en ingeniero, especialmente a mi papá, mi mamá, a mis hermanas y mi segunda madre, mi abuela, a mi familia completa por brindarme su apoyo para que no me rindiera en algunos momentos en los que pensé podría ser una opción, a mi familia aquí en La Habana, en Las Tunas, y en Jatibonico, a mis amistades de ahora y a las de hace ya algún tiempo con las que compartí momentos inolvidables en esta universidad, a mis profesores que estuvieron ahí en el momento en que los necesitaba, a los tutores de la tesis, a mi compañero de tesis Rafael (Felo) por aguantarme en estos 4 años que estuvimos compartiendo la misma aula y la misma mesa, a mis compañeros de aula que a pesar de conocerme como soy siempre me apoyaron y ayudaron en todo momento. Y, por último, pero no menos importante un agradecimiento muy especial a esa persona que compartió conmigo estos dos últimos años de mi carrera, que se mantuvo ahí al lado mío a pesar de muchos errores que cometí, que me escucho cada vez que necesito hablar, que me enseñó a pensar de una forma diferente, que me mostró lo que es vivir la vida, te estaré eternamente agradecido por tu apoyo Chiquita mía, en fin, a todos aquellos que me apoyaron en los momentos que más los necesitaba. Hala Madrid

De Marcos

A mi mamá y mi papá por acompañarme en estos años de estudio. Mis amigos de años anteriores que no pude graduarme con ellos pero al final logre mi objetivo: Miguel, Eduardo, Moro, Edwin, Yunion, Gonce y muchos más. Los nuevos compañeros de aula por estos 5 años de estudio y compartir, aquellos que se quedaron en el camino como Yusniel, Danger. Los nuevos integrantes que se incorporaron en 2 año de Artemisa La Peketita, Yaritza, Katira y aquellos que me consideran el viejito del aula. Los socios de andadas Angel, Osuel, Camilo y los restantes compañeros del aula. A Kuki por esta siempre cuando la necesite para cualquier problema y escuchar muchas veces. Los tutores por su apoyo durante todo el transcurso de la realización de este trabajo, a Osmany que ya no se encuentra entre nosotros por confiar siempre en nosotros. Los compañeros de escalera del edificio Dinier, Roylan, Humberto, Yuniorkis, Dalianne y muchos más. Mi compañero de tesis Marco por los años de estudio y sobre todo por aguantar estos meses de tesis y ser campeones de los últimos mundialitos de fútbol de la Facultad 4 y a todos los profesores que durante toda la carrera me enseñaron y me convirtieron en la persona que soy hoy. Forza Milan

De Rafael

Resumen

Los Sistemas de Gestión, en estos tiempos, prometen un impulso en las estrategias encaminadas a lograr la optimización de los procesos de negocio. Permitir un desarrollo perdurable en la gestión de políticas, procedimientos y procesos es la principal meta de estos sistemas. El Agente Transitorio de Cargas TRANSCARGO se encuentra inmersa en estas transformaciones; en la misma no existe un espacio virtual donde se encuentre centralizada o fácilmente accesible información sobre el arrendamiento de contenedores y toda la documentación que este proceso genera. El objetivo de este trabajo es desarrollar un módulo para el Sistema de Gestión Integral del Agente Transitorio de Cargas que permita la gestión de información entorno al arrendamiento de contenedores en la entidad, como solución de este problema. Para el desarrollo de este módulo se empleó el framework OpenObject y el lenguaje Python en su versión 2.7. Se utilizó como IDE Pycharm 4.5 y para el modelado, Visual Paradigm 8.0 utilizando como lenguaje de modelado UML 2.0. El ciclo de desarrollo se realizó guiado por la metodología AUP en su variación UCI. Finalmente, se logró desarrollar un módulo para la gestión de la información relacionada con el arrendamiento de contenedores: entre sus principales funcionalidades permite arrendar y devolver uno o varios contenedores, obtener reportes según criterios y gestionar órdenes de liberación.

Palabras clave: arrendamiento de contenedor, contenedor, gestión de información, sistema gestión.

Índice de contenidos

| | |
|--|-----------|
| Introducción | 1 |
| Capítulo 1: Fundamentación teórica | 5 |
| 1.1 Conceptos fundamentales..... | 5 |
| 1.2 Sistema integral de gestión empresarial..... | 6 |
| 1.2.1 OpenERP | 7 |
| 1.3 Análisis de soluciones similares | 9 |
| 1.3.1 Internacionales | 9 |
| 1.3.2 Nacionales | 11 |
| 1.4 Metodologías de desarrollo de software, lenguaje y herramientas para el modelado | 12 |
| 1.4.1 Metodologías de desarrollo de software | 12 |
| 1.4.2 Lenguajes de modelado | 14 |
| 1.4.3 Herramienta para el modelado | 15 |
| 1.5 Lenguajes y tecnologías de programación | 15 |
| 1.5.1 Lenguaje del lado del cliente | 16 |
| 1.5.2 Lenguajes del lado del servidor | 16 |
| 1.5.3 Framework de desarrollo..... | 17 |
| 1.5.4 Sistema Gestor de Base de Datos..... | 18 |
| 1.5.5 Entorno de desarrollo | 19 |
| 1.6 Conclusiones parciales | 19 |
| Capítulo 2: Propuesta de solución | 21 |
| 2.1 Descripción del sistema | 21 |
| 2.2 Descripción de proceso de negocio..... | 21 |
| 2.3 Modelo de dominio..... | 24 |
| 2.4 Requisitos del sistema | 26 |
| 2.4.1 Requisitos funcionales..... | 26 |
| 2.4.2 Requisitos no funcionales..... | 27 |
| 2.5 Especificación de requisitos | 28 |
| 2.6 Descripción de requisitos por proceso..... | 30 |
| 2.7 Arquitectura de Odoo | 35 |
| 2.8 Patrones arquitectónicos..... | 37 |
| 2.9 Modelo de diseño..... | 38 |
| 2.9.1 Patrones de diseño | 38 |
| 2.9.2 Diagrama de clase del diseño | 40 |
| 2.9.3 Diagrama de secuencia del diseño..... | 40 |
| 2.10 Modelo de datos..... | 41 |
| 2.10.1 Descripción de las tablas de la base de datos | 42 |
| 2.11 Conclusiones parciales..... | 43 |
| Capítulo 3: Implementación y prueba..... | 44 |
| 3.1 Diagrama de componentes | 44 |
| 3.1.1 Descripción de los componentes..... | 45 |

| | |
|---|-----------|
| 3.2 Diagrama de despliegue | 47 |
| 3.3 Estándar de código | 47 |
| 3.3.1 Estructura del módulo..... | 47 |
| 3.3.2 Archivos XML | 48 |
| 3.3.3 Archivos Python..... | 49 |
| 3.4 Seguridad..... | 50 |
| 3.5 Pruebas a los sistemas informáticos | 51 |
| 3.5.1 Métodos de prueba | 52 |
| 3.5.2 Diseño de caso de prueba..... | 53 |
| 3.6 Resultados de las pruebas..... | 55 |
| 3.6.1 Resultados de las pruebas | 55 |
| 3.7 Conclusiones parciales | 56 |
| Conclusiones generales..... | 57 |
| Recomendaciones | 58 |
| Glosario de términos | 59 |
| Referencias bibliográficas | 61 |
| Bibliografía..... | 66 |

Índice de tablas

| | |
|---|----|
| Tabla 1: Descripción del proceso de negocio contratación de contenedores..... | 22 |
| Tabla 2: Requisitos funcionales..... | 26 |
| Tabla 3: Especificación de requisito Incluir contenedor arrendado | 29 |
| Tabla 4: Descripción del requisito por proceso Incluir contenedor arrendado | 30 |
| Tabla 5: Descripción de la tabla container_leasing_container | 42 |
| Tabla 6: CP Incluir contenedor arrendado | 53 |
| Tabla 7: Resultado de las pruebas por iteraciones | 55 |

Índice de ilustraciones

| | |
|---|----|
| Figura 1: Diagrama de proceso de negocio contratar contenedores | 24 |
| Figura 2: Diagrama de clases del Modelo de Dominio | 25 |
| Figura 3: Prototipo interfaz de usuario del requisito Incluir contenedor arrendado | 33 |
| Figura 4: Prototipo interfaz de usuario del requisito Incluir contenedor arrendado (Flujo alternativo información incompleta) | 34 |
| Figura 5: Prototipo interfaz de usuario del requisito Incluir contenedor arrendado (Flujo alternativo información errónea) | 35 |
| Figura 6: Arquitectura de Odoos (CALVO LOPEZ, 2010)..... | 37 |
| Figura 7: Modelo Vista Controlador..... | 38 |
| Figura 8: Diagrama clase de diseño Gestionar contenedor arrendado..... | 40 |
| Figura 9: Diagrama de secuencia de diseño del requisito Incluir contenedor arrendado | 41 |
| Figura 10: Diagrama modelo de datos del módulo Arrendamiento de contenedores..... | 42 |
| Figura 11: Diagrama de componentes del módulo Arrendamiento de contenedores | 44 |
| Figura 12: Paquete Vista..... | 45 |
| Figura 13: Paquete Odoos..... | 45 |
| Figura 14: Paquete Modelo..... | 45 |
| Figura 15: Paquete Reporte..... | 46 |
| Figura 16: Paquete Wizard | 46 |
| Figura 17: Diagrama de despliegue | 47 |
| Figura 18: Estructura del módulo de la propuesta de solución | 48 |
| Figura 19: Resultados de las pruebas al módulo..... | 56 |

Introducción

En la actualidad se ha ganado experiencia en la concepción de los sistemas de gestión, a la par del creciente desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC); imponiéndose llevar a cabo la transformación y el perfeccionamiento en la organización de la información de forma efectiva y con calidad, en correspondencia con los cambios del entorno. El uso de estas tecnologías en el mundo moderno ha incrementado considerablemente la capacidad de almacenamiento, gestión, procesamiento y distribución de datos e información de todo tipo lo que posibilita su aplicación en diferentes esferas de la sociedad tales como la educación, el transporte y salud.

En Cuba la informatización de las empresas ha ido en constante avance, surgiendo como reto el adaptar las organizaciones al cambio, con un mecanismo de mejoramiento continuo de los procesos. Se han desarrollado sistemas para la gestión de la información, con el objetivo de lograr mayor calidad y rapidez en el procesamiento de la información que se genera en los sectores productivos, industriales, educativos (TRINCHET SOLER, y otros, 2004).

Una de las empresas que se encuentran inmersas en la informatización de sus procesos es el Agente Transitario de Cargas TRANSCARGO, permitiéndole estar a la par con sus homólogos en el mundo. Esta entidad brinda servicios relacionados con el transporte nacional e internacional de cargas comerciales (CC), cargas no comerciales (CNC) y sus actividades conexas tanto en la esfera marítima como aérea, es el encargado de organizar y realizar la entrega de mercancías cumpliendo fielmente los intereses de sus clientes. La empresa está estructurada por Unidades Empresariales de Base (UEB) entre las que se encuentra la Dirección de Logística en la que se prestan los siguientes servicios:

- Agrupe y desagrupe de contenedores.
- Agenciamiento aduanal.
- Arrendamiento de contenedores.

En el proceso de arrendamiento de contenedores se debe realizar un control de las solicitudes de los clientes, los contenedores arrendados pueden ser de la misma empresa o de los proveedores; es decir que la empresa arrienda contenedores a otra para ella arrendárselos a sus clientes. También se lleva el control de los tipos de contenedores, cantidad de unidades averiadas, así como su devolución; seguimiento de los clientes y la cantidad de contenedores arrendados por cada uno y el depósito donde se encuentra él o los contenedores. Actualmente este proceso se realiza de forma manual por lo que se hace engorroso debido a la gran cantidad de contenedores arrendados y toda la documentación que se debe controlar para cada uno, teniendo como consecuencia el

retraso en la dinámica de la empresa e inconformidades tanto de los clientes como de los proveedores. Además, se han detectado las siguientes deficiencias:

- Pérdida de tiempo en la búsqueda de los contenedores que están disponibles.
- Información duplicada.
- Pérdida de información al realizarse los documentos manualmente.
- No se cuenta con un historial de arrendamiento de cada contenedor.
- Requiere de un seguimiento continuo para asegurarse de las fechas de caducidad de las solicitudes y proceder a la facturación.
- Al estar en formato manual se necesita un local para almacenar los registros de documentos por un tiempo.
- Se dificulta la realización de reportes solicitados por la dirección.

A partir de la problemática antes descrita, se plantea como **problema a resolver** la siguiente interrogante: ¿Cómo contribuir a la gestión de la información que se genera del arrendamiento de contenedores en el Agente Transitario de Cargas TRANSCARGO?

Se determina como **objeto de estudio**: la gestión de la información del arrendamiento de contenedores.

Para dar cumplimiento al problema planteado se propone como **objetivo general**: desarrollar un módulo para la gestión de la información que se genera del proceso arrendamiento de contenedores en el Agente Transitario de Cargas TRANSCARGO.

Se determina como **campo de acción**: la gestión de la información automatizada del proceso arrendamiento de contenedores.

Del objetivo general planteado se derivan los siguientes **objetivos específicos**:

- Construir los referentes teóricos relacionando los aspectos fundamentales que sustentan la investigación, mediante los cuales se consulta, recopila y extrae la información relevante sobre el problema a investigar.
- Caracterizar el proceso de gestión de información del arrendamiento de contenedores en la empresa.
- Diseñar la propuesta de solución que permita contribuir a la gestión de la información que se genera en el arrendamiento de contenedores.

- Implementar el módulo arrendamiento de contenedores utilizando las herramientas, tecnologías y lenguajes definidos.
- Validar la solución desarrollada mediante las pruebas.

Como **hipótesis** se define: el desarrollo de un módulo para la gestión del proceso arrendamiento de contenedores en el Agente Transitario de Cargas TRANSCARGO, permitirá contribuir a la gestión de la información del arrendamiento de contenedores.

Para la realización de este trabajo se utilizaron los siguientes métodos de investigación:

Métodos teóricos

Analítico-sintético: el empleo de este método se evidencia en el análisis, selección e identificación de los conceptos y definiciones más importantes relacionadas con el tema de investigación, lo que permitió elaborar una propuesta de solución enfocada en la situación planteada.

Histórico-Lógico: este método se utilizó para el estudio de las tendencias actuales de otras aplicaciones, así como las características de la metodología de desarrollo, lenguajes de programación y framework en el desarrollo del módulo.

Análisis documental: el uso de este método se muestra en la búsqueda y las consultas de la literatura especializada en el tema abordado en la investigación.

Métodos empíricos:

Observación: el uso de este método permite obtener información relacionada con la gestión de contenedores que se realiza actualmente en la empresa.

Entrevista en profundidad: para identificar las funcionalidades a implementar y establecer criterios en vista del desarrollo del módulo se entrevistó a los trabajadores encargados del arrendamiento de contenedores para así tener una mejor visión de lo que necesita el cliente.

El documento consta de tres capítulos en los que se abordan diferentes temáticas. Estos están estructurados de la siguiente forma:

Capítulo 1 Fundamentación teórica

Se exponen los conceptos y fundamentos generales que sirven de soporte teórico a la propuesta de solución que se plantea. Se realiza un análisis de los sistemas similares tanto internacionales como nacionales, donde se explica su funcionamiento y características. Se analizan las herramientas y lenguajes de programación definidos para el desarrollo del módulo Arrendamiento

de contenedores para el Agente Transitario de Cargas TRANSCARGO, así como la metodología a emplear en el desarrollo del módulo.

Capítulo 2 Descripción de la solución propuesta

Se identifican las funcionalidades y características del sistema. Se elaboran los artefactos correspondientes al análisis y el diseño del software según la metodología de desarrollo de software definida para guiar el desarrollo de la solución de la presente investigación.

Capítulo 3 Implementación y pruebas

Se describen los elementos necesarios para la implementación del módulo. Se identifica el tipo de prueba a efectuar y se describe su aplicación durante la comprobación del sistema para determinar si cumple con las funcionalidades identificadas. Se describe el análisis de los resultados alcanzados en las pruebas y se mencionan las acciones realizadas para resolverlas.

Capítulo 1: Fundamentación teórica

Introducción

En el presente capítulo se expone una visión general de los aspectos relacionados con el arrendamiento de contenedores, los conceptos y aspectos teóricos asociados al problema. En la primera parte se muestra una sección teórica donde se tratan los temas relacionados a la conceptualización de la gestión de contenedores y sus definiciones asociadas. Por último, se analizan y explican detalladamente las herramientas, tecnologías, lenguajes y metodología de software definidas en la arquitectura del proyecto a utilizar para el desarrollo de la propuesta de solución.

1.1 Conceptos fundamentales

El proceso de arrendamiento de contenedores se puede encontrar de diferentes formas en dependencia del tipo de empresa que lleve a cabo el alquiler. Para ello se hace necesario expresar las palabras que conforman el arrendamiento de contenedores, llevándolo al contexto de los agentes transitorios y por último relacionándolo con los sistemas de gestión utilizados en empresas.

Contenedor

Contenedor, “del inglés *container*, es un recipiente que se utiliza para depositar residuos o un embalaje grande, de dimensiones y tipos normalizados internacionalmente, que se utiliza para el traslado de mercancías” (Definición, 2010).

Otra definición de contenedor, es el elemento de transporte o carga que consiste en un recipiente especialmente construido para facilitar el traslado de mercancías, como unidad de carga en cualquier tipo de transporte. Cuenta con accesorios que permiten su manejo rápido en la carga, descarga y trasbordo (Innova, 2013).

Por lo tanto, después de analizar los conceptos antes expuestos, el equipo de trabajo entiende por contenedor un dispositivo para almacenar o transportar mercancías.

Arrendamiento

Arrendamiento, se le llama así al préstamo, adquisición del uso o aprovechamiento temporal, ya sea de cosas, obras o servicios a cambio de un valor (DefinicionABC, 2010).

Arrendamiento de contenedores, es el proceso que se realiza mediante contratos de arrendamiento donde las partes acuerdan el valor por el uso de los contenedores. Esto queda plasmado en un documento válido por el tiempo acordado entre las dos partes, el arrendador y el cliente

(DefinicionABC, 2010).

Analizando los conceptos anteriores se entiende por arrendamiento de contenedores a la acción de obtener un contenedor por un tiempo definido.

Agentes transitarios

Transitario, según Diccionario de la Real Academia Española, “*empresa o entidad que se ocupa de las gestiones administrativas y logísticas necesarias para el transporte de mercancías, especialmente en puertos y aeropuertos*” (Academia, Real).

El transitario es una persona física o jurídica que presta servicios para el transporte internacional de mercancías. Es un intermediario entre el exportador e importador y las compañías de transporte ya sean terrestre, marítimo, aéreo o multimodal (IBERTRANSIT, 2013).

El equipo de trabajo entiende por agente transitario al encargado de los trámites legales para el transporte de mercancías nacionales como internacionales.

Sistema de gestión

Sistema de gestión es aquel que permite lograr los objetivos y metas de una organización, a través de una serie de estrategias, entre las que se encuentran optimización de procesos y un enfoque basado en la gestión y la disciplina (Implementacion SIG, 2015).

El equipo de trabajo entiende por sistema de gestión a un conjunto de etapas integradas en un proceso con el objetivo de lograr un objetivo único.

Gestión empresarial

La gestión empresarial conocida también como administración de empresas, es una ciencia social que estudia la organización de las empresas y la manera como se gestionan los recursos, procesos y resultados de sus actividades (ClubEnsayos, 2011).

Actualmente las empresas de todo tipo del mundo se encuentran inmersas en la gestión empresarial para agilizar todos sus procesos. Por esto se prosigue analizar que es un sistema para la gestión de los procesos en las empresas y cuáles son sus principales características.

1.2 Sistema integral de gestión empresarial

Sistema de gestión empresarial (por sus siglas en inglés ERP) es un sistema integral de gestión empresarial diseñado para modelar y automatizar la mayoría de los procesos en una empresa (área contabilidad y finanzas, comercial, logística y producción). Además, unifica y ordena la información

en un solo lugar posibilitando la toma de decisiones de forma rápida y segura. Los ERP son el núcleo de otras aplicaciones como CRM (Customer Relationship Management), Data Mining (Conversión de datos en información útil) y BI (Business Intelligence) (EXXIS, 2010).

ERP se define según dos principios básicos (Prodware, 2016):

- Aplicaciones informáticas como módulos independientes, pero perfectamente compatibles con una única base de datos común.
- El uso de un motor de flujos de trabajo debe permitir definir todas las tareas de un proceso y gestionar su aplicación en todos los módulos del sistema.

Principales ventajas de los sistemas ERP (NUBISTALIA, 2011):

- Permiten agilizar los procesos empresariales gracias a la capacidad de obtener información y procesarla de forma inmediata.
- Permiten disminuir los tiempos y los costos de los procesos.
- Eliminan la posible duplicidad de datos y operaciones innecesarias.
- Proporcionan información fiable e íntegra.
- Crean una visión unificada de la empresa, común a todos los departamentos y trabajadores.

Implantar un ERP y automatizar los procesos en cualquier empresa reduce costos en la gestión de la información y aumenta la productividad de la empresa, permitiéndoles a las mismas contar con un sistema donde toda la información se encuentra centralizada. A continuación, se muestran las características de unos de sus principales exponentes dentro de los ERP.

1.2.1 *OpenERP*

OpenERP (cambia su nombre a Odoo desde la versión 8) es un ERP completo de código abierto y sin costo de licencias que cubre las necesidades de las áreas: contabilidad y finanzas, ventas, recursos humanos, compras, proyectos, almacenes, CRM (Customer Relationship Management) y fabricación (Domatrix, 2012).

Odoo cuenta con varios módulos base como son (ODOO):

- Gestión de relaciones con el cliente (CRM).
- Gestión de proyectos.
- Gestión de almacenes.

- Gestión contable y financiera.
- Gestión de compras.
- Gestión de ventas.
- Recursos humanos.
- Marketing.
- Fabricación.
- Gestión del conocimiento.

Además de los módulos oficiales actualmente existen más de 500 módulos que completan el programa y que permiten adaptar Odoo a las necesidades del entorno de la empresa. También dispone de soluciones verticales para sectores como sanidad, hostelería, ingeniería civil e industria alimentaria (NUBISTALIA, 2011).

Este ERP posee muchas facilidades o fortalezas que lo hacen estar por encima de sus homólogos (Domatrix, 2012):

- Libertad: sin dependencia del proveedor, como producto no pertenece a ninguno de sus distribuidores, es decir, tiene la opción de elegir el proveedor que más les convenga a sus necesidades.
- Filosofía Open: puede contratar únicamente lo que necesite. Desarrollo de algún módulo en específico o formación/soporte técnico de algún módulo oficial.
- Código abierto: puede disponer del código para realizar cualquier mejora a algún módulo existente o crear uno nuevo de acuerdo a las necesidades existentes.
- Conectividad: visualización de informes en formato estándar PDF, importación/exportación con MS Office, Excel o CVS¹ y la posibilidad de conexión con casi cualquier *software* utilizando *webservices*.
- Gratuito: es un producto que no tiene coste de licencias.
- Multiplataforma: la interfaz web de Odoo le permite acceder desde cualquier ordenador independiente del sistema operativo (GNU/Linux, Mac OS X o Windows) incluso tablets y

¹ CVS: Los archivos **CSV** (del inglés *comma-separated values*) son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas y las filas por saltos de línea. Los campos que contengan una coma, un salto de línea o una comilla doble deben ser encerrados entre comillas dobles.

smartphone con Android o IOS.

- OpenObject: el *framework* de OpenERP permite un desarrollo rápido de funcionalidades o conectividad con otras plataformas.
- Variedad: cuenta actualmente con una gran cantidad de módulos liberados que se pueden combinar y/o servir como base para construir casi cualquier solución vertical.
- Integración: módulos de gestión y aplicaciones web propias para construir portales web, tiendas online, foros o eventos; todo dentro de una plataforma única y control unificado.
- Fácil migración: la herramienta oficial importa y exporta datos maestros en formato .csv para que resulte sencillo seguir trabajando con los datos desde la aplicación actual.

Con el objetivo de identificar las experiencias efectivas de las empresas que cuenten con este ERP y así obtener una solución al problema existente, es necesario realizar un estudio sobre los módulos y sistemas actuales en la gestión de información en el área de los contenedores.

1.3 Análisis de soluciones similares

Para desarrollar una aplicación informática con un alto nivel de calidad es necesario analizar las soluciones similares en cuanto a módulos desarrollados en Odoo y empresas que brinden servicios de alquiler, para detectar las funcionalidades que se puedan aplicar a la propuesta de solución de la presente investigación.

1.3.1 Internacionales

Product Container es un módulo desarrollado por la comunidad de Odoo S.A para la versión 7.0 en adelante de OpenERP útil para gestionar y dar seguimiento a los contenedores en los cuales los productos son entregados al cliente y su retorno de los mismos (Odoo S.A, 2012).

Algunas de las características principales de este módulo (Odoo S.A, 2012):

- Definir el producto a guardar en el contenedor.
- Seleccionar el contenedor apropiado para cada producto.
- Devolución de los contenedores a los clientes.

Funcionalidades del módulo (Odoo S.A, 2012):

- Manejo de contenedores: permite el manejo de contenedores según el producto que almacenen y administrar fácilmente el stock de contenedores; permitiendo así un mejor

manejo a la hora de realizar un seguimiento y gestión de los contenedores.

- Definir productos por contenedores: permite crear un nuevo producto y seleccionar el contenedor al que se va asignar.
- Contenedor para cantidades múltiples: permite elegir el producto a guardar en el contenedor, seleccionar el tipo de contenedor a utilizar y la cantidad de productos y envases en la pestaña de ventas.

Product Container Tracking es un módulo desarrollado por la comunidad de Odoo S.A para la versión 7.0 o superior de OpenERP útil para gestionar y realizar un seguimiento de los contenedores en los que se reciben o entregan productos, también para gestionar almacén, ubicación actual de envases y productos de diferentes números de serie. Además con el uso de este módulo se pueden crear movimientos de entrada o salida del recipiente junto con el producto, por lo que se puede dar seguimiento fácilmente al contenedor en el almacén (Odoo S.A, 2012).

Algunas de las características principales de este módulo (Odoo S.A, 2012):

- Obtener la locación exacta de los contenedores según su número de serie.
- Obtener cantidad de contenedores exactos por ubicación.
- Devolver contenedores a sus proveedores o readmisión como contenedores de clientes.

Funcionalidades del módulo (Odoo S.A, 2012):

- Contenedor de un solo producto: define el producto a guardar en el contenedor, también permite seleccionar el contenedor a utilizar en cualquier movimiento. Asigna números de series aleatorios para los productos entrantes y OpenERP crea automáticamente un movimiento de entrada con ese número de serie.
- Contenedor para más de un producto: precisa los productos a guardar en el contenedor, también permite elegir un contenedor para cada producto que desee. Permite elegir la cantidad de productos y asignarlos a números de serie y paquetes, OpenERP automáticamente crea un número de serie con el número del paquete seleccionado y lo asigna como número de identificador al serial original.

IMB Container es un programa desarrollado por la compañía Info Mirben S.L para el control y gestión específico de empresas dedicadas a la venta o alquiler de contenedores de obra. Con esta aplicación se puede gestionar todo lo relacionado al arrendamiento de contenedores y su venta, disponiendo para ello un sistema de control y facturación (RedAccenir, S.L, 2009).

Algunas características principales de esta aplicación (RedAccenir, S.L, 2009):

- Creación de tres tipos de contenedores, cada uno con sus especificaciones.
- Configurar precios de venta por contenedores.
- Desarrollada para sistema operativo Windows.

Funcionalidades de la aplicación (RedAccenir, S.L, 2009):

- Almacén: dar entrada a los contenedores que se encuentren en stock, en qué obra se encuentra, cliente que lo ha contratado y tiempo de contratación.
- Informes: emitir facturas por cada albarán² de entrega, elaborar informes de almacén, clientes o ventas.
- Albaranes: elaborar albaranes de entrega de los contenedores, elaborar hojas de ruta con su tarifa correspondiente y kilometrajes.

1.3.2 Nacionales

No se encontraron evidencias de módulos realizados en Odoó para la gestión de la información generada en el servicio de arrendamiento de contenedores de una empresa transitaria por lo que se analizaron empresas y entidades que prestan servicios de alquiler o arrendamiento para analizar sus funciones y servicios prestados que puedan servir de ayuda en la solución propuesta.

Empresa Central de Equipos (CUBIZA) es una organización líder nacional en el servicio de elevación y colocación de cargas. Ofrece soluciones relacionadas con el izaje, la transportación de carga especializada y la cimentación indirecta. Dentro de los servicios que presta se encuentran (CUBIZA, 2011):

- Alquiler de equipos con operarios.
- Brinda asesoría técnica y elabora el proyecto para cualquier operación de reparación, mantenimiento y construcción de nuevas obras.

Empresa de Servicio de la Agroindustria Azucarera (ESAZUCAR) perteneciente al Grupo Azucarero AZCUBA presta servicios de gastronomía, alojamiento y otros a las organizaciones del mismo grupo y a terceros. Dentro de los servicios que presta esta empresa se encuentra (Grupo

² Albarán: El albarán, también conocido como nota de remisión o nota de entrega, es un documento mercantil que acredita la entrega de algún producto o la prestación de un servicio.

Azucarero AZCUBA, 2014):

- Alquiler de almacenes, locales y de equipos especializados de transporte agrícola.
- Arrendamiento de oficinas, incluyendo los servicios generales asociados a las entidades del Ministerio del Azúcar.
- Alquiler de locales, de recreación y alimentación asociados a los trabajadores del Ministerio del Azúcar.

Tras los estudios de los módulos y sistemas similares expuestos, se hace necesario para la propuesta de solución tomar como apoyo las tendencias actuales sobre los servicios y las funcionalidades de los mismos. Algunas de estas funcionalidades analizadas y que están presente en la propuesta de solución son alquiler y devolución de contenedores, el trabajo con los identificadores de los contenedores; es decir que todas las soluciones vistas solo manejan los contenedores, sus movimientos y la mercancía que transportan. Otras de las funcionalidades analizadas es el alquiler de locales y equipos, que permiten analizar los datos recogidos para comprender. Pero ninguno realiza en su totalidad las funciones del sistema propuesto porque no cuentan con una opción de generar reportes por diferentes criterios, no manejan los contenedores averiados, no cuentan con un sistema de notificaciones para la facturación de los contenedores.

1.4 Metodologías de desarrollo de software, lenguaje y herramientas para el modelado

Las metodologías, lenguajes y herramientas no son más que un conjunto de técnicas y recursos de software utilizados para lograr un exitoso desarrollo de aplicaciones informáticas. A continuación, se brindan características de la metodología, lenguajes y herramientas para el modelado definidas en el proyecto.

1.4.1 Metodologías de desarrollo de software

Una metodología es definida como *“la serie de métodos y técnicas de rigor científico que se aplican sistemáticamente durante un proceso de investigación para alcanzar un resultado teóricamente válido. En este sentido, la metodología funciona como el soporte conceptual que rige la manera en que aplicamos los procedimientos en una investigación”* (Significados, 2013).

AUP (Agile Unified Process) es una versión simplificada del Proceso Unificado de Rational (RUP) desarrollada por Scott Ambler. AUP aplica técnicas ágiles incluyendo desarrollo orientado a pruebas, modelado ágil, gestión de cambios ágil y refactorización de base de datos para mejorar la productividad (Ambysoft Inc, 2005).

Esta metodología pasa por diferentes fases en su ciclo de vida como son (Ambysoft Inc, 2005):

- Inicio: se encarga de definir el alcance del proyecto, los costos y plazos, los riesgos y la factibilidad del mismo.
- Elaboración: encargada de identificar y validar la arquitectura.
- Construcción: tiene como objetivo la creación de la documentación de apoyo, así como elaboración del software de forma incremental.
- Transición: el propósito de realizar las pruebas al proyecto para su posterior liberación funcional.

AUP (Agile Unified Process) variación UCI surge con la necesidad de establecer una metodología capaz de adaptarse a las características específicas de cada proyecto y que se adapta el ciclo productivo de la UCI. Mantiene los aspectos fundamentales del AUP en las fases de la misma (SANCHEZ RODRIGUEZ, 2015).

Fases del ciclo de vida de la metodología AUP-UCI (SANCHEZ RODRIGUEZ, 2015):

- Inicio: planeación del proyecto, es decir, el alcance proyecto, costos, estimaciones de tiempo.
- Ejecución: en esta se unifican las fases de AUP (elaboración, construcción, transición), es decir, todo lo relacionado con la arquitectura, requisitos, documentación de apoyo.
- Cierre: se analizan los resultados finales del proyecto, ejecución y el cierre del mismo.

Para mejor comprensión de las fases de AUP-UCI ver [Anexo #1](#).

Para el modelado del negocio esta metodología propone tres variantes a utilizar:

- Casos de Uso del Negocio (CUN).
- Descripción de Proceso de Negocio (DPN).
- Modelo Conceptual (MC).

Formas de recoger los requisitos del proyecto:

- Caso de Uso del Sistema (CUS).
- Historia de Usuario (HU).
- Descripción de Requisitos por Proceso (DRP).

Con esto aparecen cuatro escenarios para modelar el sistema en el proyecto (SANCHEZ RODRIGUEZ, 2015):

- Escenario No 1: proyectos que modelen el negocio con CUN solo pueden modelar el sistema mediante CUS.
- Escenario No 2: proyectos que modelen el negocio con MC solo modelan el sistema mediante CUS.
- Escenario No 3: proyectos que modelen el negocio con DPN solo pueden modelar el sistema mediante DRP.
- Escenario No 4: proyectos que no modelen negocio solo modelan el sistema mediante HU.

La metodología AUP-UCI fue seleccionada para guiar el proceso ingenieril, porque facilita un modelado ágil y se adecua a requisitos que pueden ser cambiantes. Al ser una versión ágil de RUP es apropiada para proyectos medios y garantiza la documentación necesaria en el desarrollo del software. Dentro de los escenarios recogidos en esta metodología se escogió el escenario No 3 porque permite representar una gran cantidad de niveles de detalles y relaciones entre los procesos especificados. Este se aplica a negocios a informatizar y como resultado se obtengan procesos muy complejos independientes de las personas que los manejan y ejecutan. Además, es la metodología y el escenario utilizado en el proyecto para guiar el proceso ingenieril.

1.4.2 Lenguajes de modelado

Lenguaje Unificado de Modelado (UML) es un lenguaje que permite modelar cualquier sistema a través de diagramas, permite realizar presentaciones gráficas que contienen la información relevante del sistema. Se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar y controlar la información sobre los sistemas (CASTRO, 2008).

Sus principales funciones son:

- Visualizar: permite representar cualquier sistema de forma gráfica de manera que cualquier persona lo pueda entender.
- Especificar: facilita especificar las características de un sistema.
- Construir: mediante los modelos permite construir el sistema.
- Documentar: documenta los elementos gráficos lo que se puede utilizar como documentación, permitiendo una futura revisión.

Para la realización de este trabajo se requiere de un lenguaje gráfico, con el objetivo de especificar el sistema de software de forma estándar. Después de realizar varios estudios sobre las

características del lenguaje UML se decide utilizarlo para el diseño de los artefactos en la fase de ejecución de la metodología seleccionada. Este lenguaje permite que todo software de diseño, se visualice, especifique y documente con un lenguaje de forma común. Además, cuenta con notaciones estándar y semánticas esenciales para el modelado de un sistema y se ajusta a la metodología seleccionada.

1.4.3 Herramienta para el modelado

Visual Paradigm es una herramienta que utiliza como lenguaje de modelado UML. Contribuye a una rápida construcción de aplicaciones de mayor calidad y con un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde los diagramas y su respectiva información. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue (PRESSMAN, 2005).

Esta herramienta es integrable con varios entornos de desarrollo como Eclipse, NetBeans, Visual Studio. Permite aumentar la velocidad en el análisis, captura, plan, desarrollo, comprobación y despliegue de los requisitos; es multiplataforma y cuenta con una versión libre para la comunidad (Community Edition). Genera la documentación en los formatos PDF, HTML y en formato de documento de Microsoft Word y permite importar proyectos de otras herramientas de modelado como Rational Rose, Erwin y Microsoft Visio (Visual Paradigm International, 2014).

Entre sus principales características se destacan:

- Modelado de caso de uso.
- Generación de reportes.
- Ingeniería inversa de código Java, .NET, PHP5, Python.
- Ingeniería inversa - Código a modelo, código a diagrama.

La herramienta escogida por los autores fue Visual Paradigm 8.0, la cual es multiplataforma y posee una excelente integración con los IDE (Integrated Development Environment). Brinda grandes facilidades para la creación de diagramas y cuenta con la posibilidad de realizar control de versiones durante todo el ciclo de trabajo.

1.5 Lenguajes y tecnologías de programación

Con el objetivo de analizar los lenguajes y las tecnologías de programación establecidas en el proyecto para el desarrollo del módulo fue necesario hacer un estudio de las características de las mismas, donde se analizan como lenguaje del lado del cliente XML; del lado servidor Python. Como *framework* de desarrollo OpenObject y como sistema gestor de base de datos PostgreSQL.

1.5.1 Lenguaje del lado del cliente

XML (Extensible Markup Language) es una especificación/lenguaje de programación desarrollada por el W3C. Es una versión de SGML (Standard Generalized Markup Language) diseñado especialmente para los documentos de la web. Permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones (O'Reilly Media, Inc., 2010).

Algunas de las ventajas de XML (UDIMA, 2009):

- Fácilmente procesable.
- Separa radicalmente el contenido y el formato de presentación.
- Diseñado para cualquier lenguaje y alfabeto.

Características de este lenguaje (UDIMA, 2009):

- Es un subconjunto de SGML que incorpora las tres características más importantes de este:
 - Extensibilidad, estructura y validación.
- Basado en texto.
- Orientado a los contenidos no a presentación.
- Las etiquetas se definen para crear los documentos, no tienen un significado preestablecido.
- No existe un visor genérico de XML.

Después de analizar las características del lenguaje, se decide su uso para la propuesta de solución. Es fácil de entender tanto por personas como software. Se puede utilizar en cualquier lenguaje. Además, es el lenguaje que utiliza Odoos para mediante su *framework* interpretar y elaborar las vistas.

1.5.2 Lenguajes del lado del servidor

Python es un lenguaje de *scripting* independiente de plataforma y orientado a objeto, preparado para realizar cualquier tipo de programa. Es un lenguaje interpretado, lo que trae consigo que no necesita compilar los programas para ejecutarlos, por lo que ofrece como ventajas la rapidez de desarrollo y su curva de aprendizaje es corta (ALEGSA, 2014).

En los últimos años este lenguaje se ha hecho muy popular, algunas de las principales razones son:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje.
- La sencillez y velocidad con que se crean los programas.
- La cantidad de plataformas en las que es posible desarrollar, como Unix, Windows, Mac.
- Además, el lenguaje Python es gratuito hasta para propósitos empresariales.

Después de analizar las características del lenguaje, se decide utilizar para la realización de la solución. Este posee una curva de aprendizaje muy suave, se ejecuta con *bytecode* por lo que es muy rápido. Su distribución es gratuita, es libre y multiplataforma. Posee además librerías que facilitan su uso a los desarrolladores. Además de ser el lenguaje en que se basa Odoo para la implementación de sus módulos.

1.5.3 Framework de desarrollo

OpenObject es un framework de código abierto, rápido e inteligente para el desarrollo de aplicaciones en Python. Es el framework utilizado para la construcción y desarrollo de módulos de Odoo como son: CRM (Customer Relationship Management), servicio de asistencia, producción, proyectos, eventos, acciones, recursos humanos.

Algunas características de este framework (ADINS, 2011):

- Capacidad de un módulo para llevar la modificación a los objetos y las vistas existentes. El concepto de módulo OpenObject hace posible utilizar la misma base de código para la instalación múltiple - por ejemplo, diferentes filiales, sin embargo, permite la personalización y funcionalidades en una base por base de datos.
- Basado en la web y cliente-servidor.
- Las vistas se definen en lenguaje basado en XML y OpenObject, sobre la marcha convierte estas en una aplicación cliente-servidor o el cliente basado en Web.
- Permite al desarrollador construir rápidamente aplicaciones con diferentes tipos de vista: árbol lista, calendario y Kanban, Gantt.
- OpenObject está construido alrededor de Python:
 - Moderno, orientado a objetos.
 - Portátil (Linux, Windows, Mac).

OpenObject ofrece, en un solo paquete el componente básico para la construcción de una aplicación de negocios (ADINS, 2011):

- Mapeo objeto-relacional (ORM): simplifica el acceso a la base de datos.
- Sistema de Permiso: completo sistema de permisos, menús, campos, acceso a datos, acciones comerciales.
- Multi-idioma / internacionalización: unicode UTF-8, varios idiomas por base de datos (cambio de idioma), los datos temporales almacenados en UTC.
- Campos traducibles: base de datos de los campos de texto pueden ser marcados como traducibles. Por ejemplo, el nombre del producto puede tener los siguientes nombres en español y francés "coche, voiture".
- Motor de informes: OpenObject incorpora un motor de informes con la generación de PDF.
- Los servicios Web: OpenObject ofrece sus servicios a través de XML-RPC y JSON (Javascript).
- Incorpora un motor con una completa integración al correo electrónico.
- Trabaja con el sistema de gestión de base de datos de código abierto PostgreSQL:
 - Tamaño máximo de base de datos ilimitado.
 - Máximo Tabla Size 32 TB.
 - Máximo Fila Size 1.6 TB.
 - Máximo Campo Tamaño 1 GB.

Después de analizar las características de este framework se decide utilizar para la elaboración de la solución ya que está escrito sobre Python, permite el almacenamiento de gran cantidad de información en la base de datos mediante PostgreSQL, además que incluye una serie de funcionalidades primarias como generación de PDF, motor de correo entre otros. Este entorno de trabajo es el utilizado por Odoon para mostrar y utilizar todos los módulos desarrollados.

1.5.4 Sistema Gestor de Base de Datos

PostgreSQL es un sistema de gestión de base de datos (SGBD) objeto-relacional, distribuido bajo la licencia BSD y con su 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 (MARTINEZ, 2010).

Dispone de una gran cantidad de características.(MARTINEZ, 2010) Algunas de ellas:

- Es una base de datos 100% ACID.
- Integridad referencial.
- Completa documentación.
- Acceso encriptado vía SSL.
- Disponible para Linux y Unix en todas sus variantes (AIX, BSD, SGI, Mac OS X) y Windows 32/64 bit.

Después de analizar cada una de las ventajas de este SGBD se decide utilizar para el manejo de los datos de la solución. Este es de libre acceso y utilización, además posee características como SGBD que lo hacen destacarse por encima de otros gestores de base de datos como es la manipulación de multiprocesos en vez de multihilos y es el gestor de base de datos que utiliza Odoo para su manejo con los datos.

1.5.5 Entorno de desarrollo

PyCharm es un IDE (Integrated Development Environment) desarrollado por la compañía JetBrains, está basado en IntelliJ IDEA, el IDE de la misma compañía, pero enfocado hacia Java y la base de Android Studio. Pycharm tiene cientos de funciones que lo pueden ver como una herramienta muy pesada, pero que valen la pena ya que ayuda con el desarrollo del día a día (JetBrains, 2010).

Características de PyCharm (JetBrains, 2010):

- Autocompletado, resaltador de sintaxis, herramienta de análisis y refactorización.
- Integración con framework web como: Django, Flask, Pyramid, Web2Py.
- Soporta entornos virtuales e intérpretes de Python 2.x, 3.x, PyPy, Iron Python y Jython.
- Sistemas de control de versiones: Git, CVS, Mercurial.
- Frameworks javascripts: jQuery, AngularJS.
- Debugger avanzado de Python y Javascript.

Después de realizar el estudio de las características de este IDE el equipo de trabajo se decide por Pycharm. Este posee características como es la integración con varios *framework* web como Django y control de versiones Git. Además, que se adapta perfectamente al trabajo con Odoo.

1.6 Conclusiones parciales

A partir de la revisión del estado del arte sobre los módulos y los sistemas para el arrendamiento de contenedores, se comprobó que no se encontró una aplicación que pueda ser utilizada en el

entorno de la empresa TRANSCARGO que resuelva la problemática planteada en su totalidad. Estos sistemas se van a tener en cuenta para el desarrollo de un módulo capaz de satisfacer las necesidades que actualmente presenta esta área en cada uno del proceso que realiza. En la investigación se realizará el diseño e implementación utilizando como metodología de desarrollo **AUP-UCI**, como lenguaje de programación **Python 2.7**, herramienta para el modelado **Visual Paradigm 8.0** utilizando como lenguaje de modelado **UML 2.0**. Como IDE **Pycharm 4.5** y como framework de desarrollo **OpenObject**.

Capítulo 2: Propuesta de solución

Introducción

En el presente capítulo se describe la solución propuesta y el proceso realizado para el desarrollo del módulo Arrendamiento de contenedores. Se representa el diagrama del modelo de dominio. Se detallan además los requisitos funcionales y no funcionales, plasmados en el artefacto y se presenta un ejemplo de artefactos propios de la metodología AUP-UCI. Se especifica la arquitectura del sistema y los patrones de diseño utilizados. A continuación, se muestra el modelo de diseño generando todos los artefactos correspondientes, así como el modelo de datos (importante en los sistemas en los que se gestiona información).

2.1 Descripción del sistema

El módulo propuesto será una herramienta útil para la gestión de la información en el arrendamiento de contenedores en TRANSCARGO. Pretende brindar información pertinente en la toma de decisiones para la gestión de la información en el arrendamiento de contenedores.

El módulo permitirá resolver problemas como:

- Incluir contenedores arrendados.
- Incluir órdenes de liberación.
- Generar reportes de los contenedores en el sistema.
- Gestionar los contenedores.
- Arrendar y devolver contenedores.

2.2 Descripción de proceso de negocio

Un proceso de negocio (Business Process) no es más que un conjunto de actividades que se realizan en coordinación en entorno organizacional y técnico. Estas actividades, en su conjunto, ayudan a alcanzar un determinado objetivo de negocio. Cada proceso de negocio es realizado por una única organización, pero puede interactuar con procesos de otras organizaciones (WESKE, 2007).

Business Process Management (BPM) incluye conceptos, métodos y técnicas para dar soporte al diseño, administración, configuración, realización y análisis de procesos de negocio. La base de todo es la representación explícita de los procesos con sus actividades y restricciones de ejecución.

Una vez definido (modelado), un proceso de negocio puede ser sujeto a análisis, mejora o realización (WESKE, 2007).

La siguiente tabla muestra la descripción del proceso de negocio contratación de contenedores

Tabla 1: Descripción del proceso de negocio contratación de contenedores

| | |
|-----------------------------------|--|
| Objetivo | Contratar contenedores |
| Evento(s) que lo genera(n) | Solicitud de arrendamiento de contenedores |
| Pre condiciones | El cliente debe tener un contrato de arrendamiento de contenedores con TRANSCARGO |
| Marco legal | |
| Reglas de negocio | |
| Responsable | UEB Logística |
| Clientes internos | UEB Comercial |
| Clientes externos | Proveedor SERVICONT |
| Entradas | Solicitud de contratación |
| Flujo de eventos | |
| Flujo básico | |
| 1. | Solicitud de arrendamiento de contenedores. El cliente realiza una solicitud para arrendar contenedores de diversos tipos según su necesidad. |
| 2. | Atiende solicitud de contratación. La UEB_Logística en el área de arrendamiento de contenedores atiende la solicitud del cliente. |
| 3. | Verifica contrato. La UEB_Logística en el área de arrendamiento de contenedores verifica si el cliente cuenta con un contrato con la UEB_Comercial de la empresa TRANSCARGO. |
| 4. | Chequea disponibilidad interna de contenedores. La UEB_Logística en el área de arrendamiento de contenedores chequea la disponibilidad de contenedores propios. |
| 5. | Entregar contenedor. La UEB_Logística en el área de arrendamiento de contenedores entrega el/los contenedor/es solicitados. |
| 6. | Extraer contenedor. El cliente extrae el o los contenedores solicitados y cuenta con 72h para extraerlos del depósito, vencido ese tiempo la empresa comienza a facturar por retención. Se entrega el Interchange de salida con la descripción del o de los contenedores arrendados. |
| 7. | Devuelve contenedor. El cliente devuelve el o los contenedores arrendados pasado un determinado tiempo, por averías o porque ya no los necesita más. |
| 8. | Inspecciona contenedor. El inspector de SERVICONT revisa el o los contenedores devueltos y emite el Interchange de entrada. |
| 9. | Factura contratación. La UEB_Lógista en el área de arrendamiento de contenedores factura la contratación si el o los contenedores devueltos no presentan daños. Este se determina comparando el Interchange de salida con el de entrada. |

| | |
|--|---|
| 10. | Fin del proceso. |
| Pos-condiciones | |
| 1. | Se arrendaron uno o varios contenedores a un cliente. |
| 2. | Se devolvieron uno o varios contenedores arrendados. |
| Salidas | |
| 1. | Contrato de solicitud. |
| 2. | Modelo de liberación. |
| 3. | Modelo de interchange. |
| Flujos alternos | |
| 3.a No tiene contrato con UEB_Comercial | |
| 1. | Firma contrato. El cliente firma un contrato con la UEB para poder realizar la solicitud de arrendamiento de contenedores en el UEB_ Logística. |
| 2. | Regresa al paso 1 del flujo básico |
| 4.a No existe disponibilidad interna de contenedores. | |
| 1. | Solicita contenedor. La UEB_ Logística en el área de arrendamiento de contenedores solicita el o los contenedores solicitados a algún proveedor. |
| 2. | Entrega contenedores solicitados. El proveedor entrega el o los contenedores solicitados a la UEB_ Logística en el área de arrendamiento de contenedores. |
| 3. | Va al paso 6 del flujo básico. |
| 8.a El contenedor presenta daños y es propio de TRANSCARGO | |
| 1. | Va al paso 9 del flujo básico. |
| 8.b El contenedor presenta daños y pertenece a un proveedor | |
| 1. | Factura daños. El proveedor realiza una factura a la UEB_ Logística por los daños que presenta el contenedor. |
| 2. | Transfiere factura de daños al cliente. La UEB_ Logística transfiere al cliente la factura por los daños ocasionados al o a los contenedores. |
| 3. | Va al paso 10 del flujo básico. |
| Asuntos pendientes | |
| N/A | |

A continuación, se muestra el diagrama de procesos de negocio realizado con Visual Paradigm 8.0

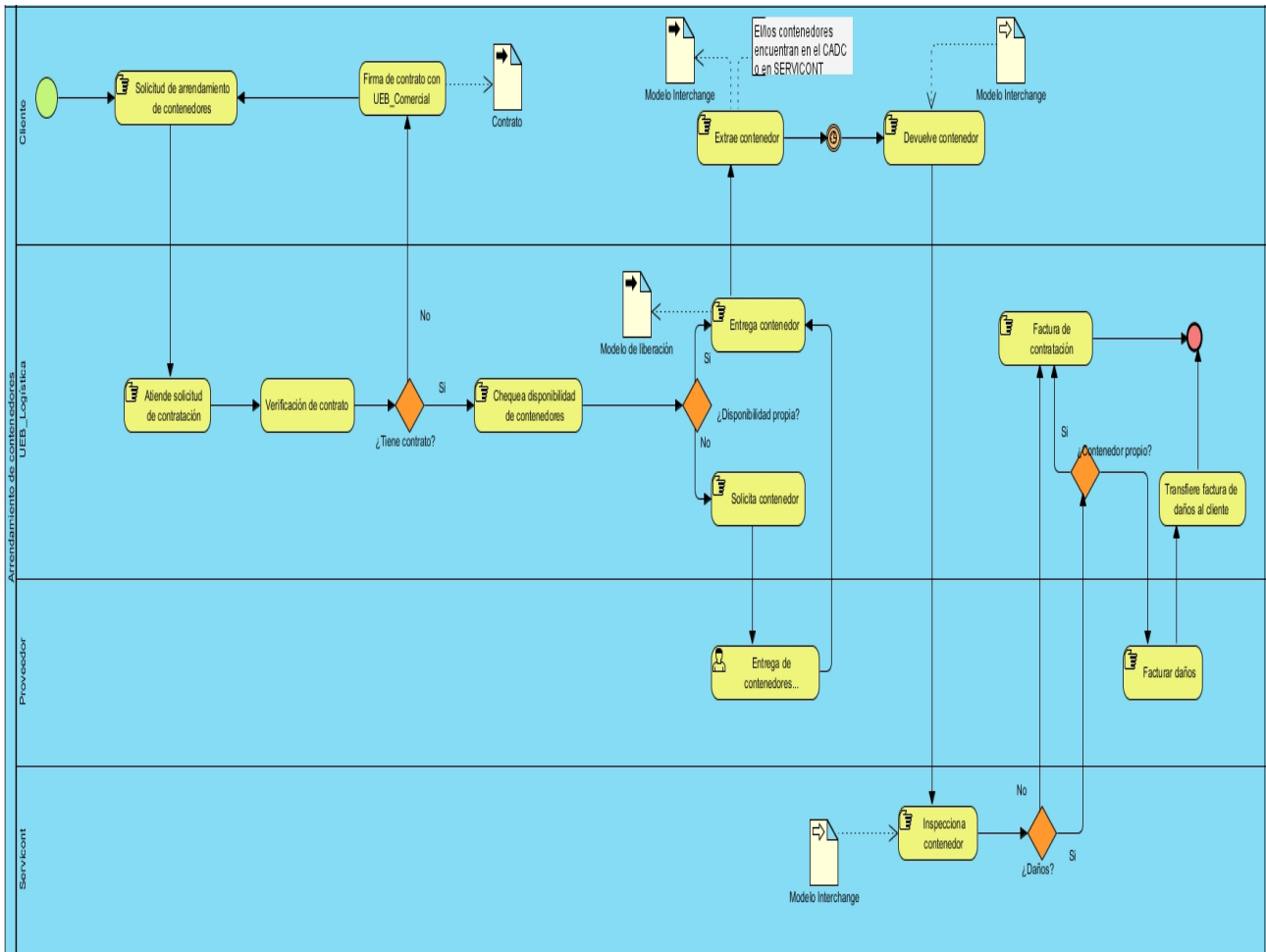


Figura 1: Diagrama de proceso de negocio contratar contenedores

2.3 Modelo de dominio

El modelo de dominio es utilizado por el analista como un medio para comprender el sistema debido a que es una representación de los conceptos y elementos de la vida real que serán usados en el mismo. Se crea con el fin de representar los conceptos claves del dominio del sistema, las propiedades más importantes y las relaciones entre los conceptos facilitando una mejor comunicación entre desarrolladores y el cliente (LARMAN, y otros, 2003). A continuación, se muestra el diagrama de clases del modelo de dominio del módulo Arrendamiento de contenedores.

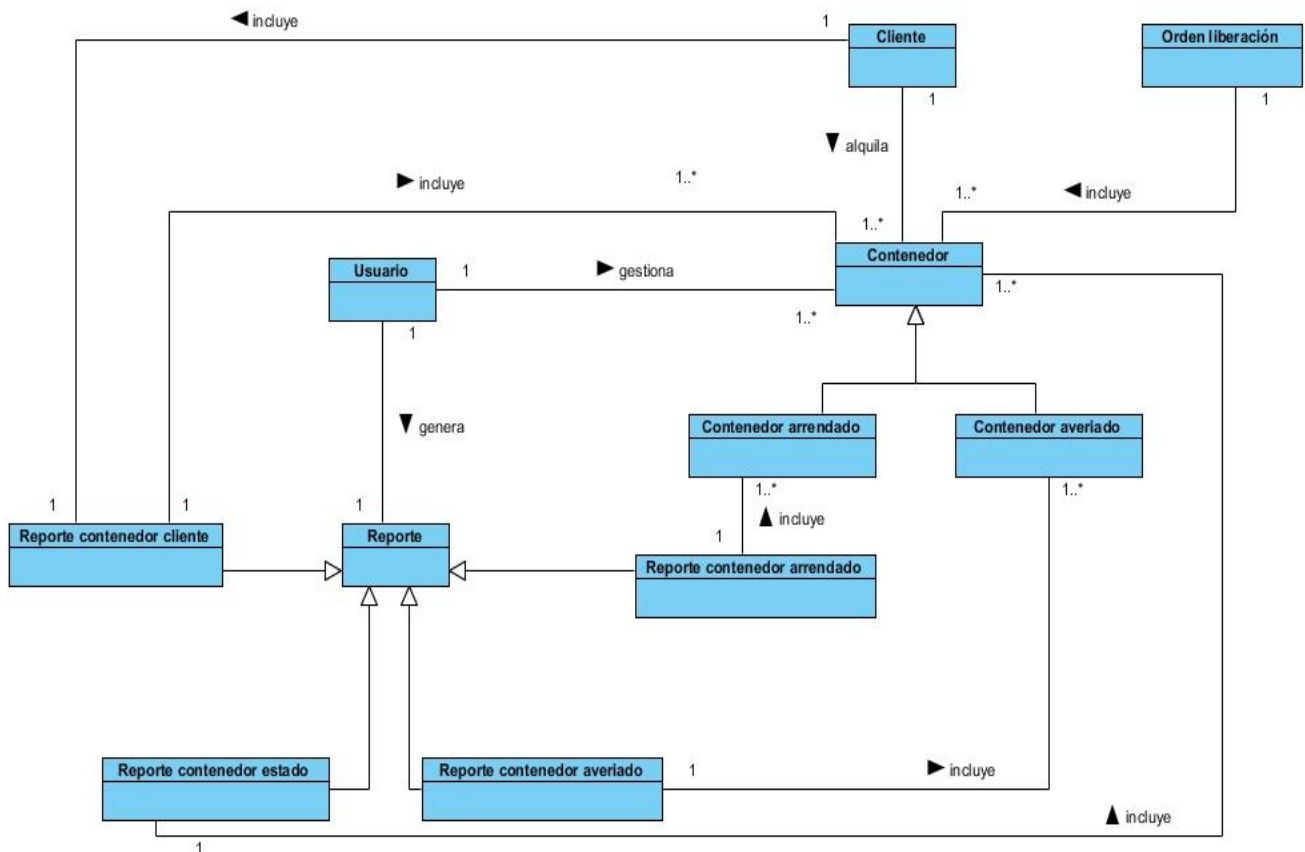


Figura 2: Diagrama de clases del Modelo de Dominio

A continuación, se muestran una descripción de cada una de las clases del modelo de dominio:

Usuario: identifica a las personas que tendrán acceso al sistema con el objetivo de gestionar los contenedores y generar los reportes.

Contenedor: contiene los valores de los contenedores.

Cliente: persona o empresa que alquila contenedores.

Orden liberación: información referente a las órdenes de liberación teniendo en cuenta los contenedores.

Contenedor arrendado: identifica los contenedores arrendados con que cuenta la empresa teniendo en cuenta los contenedores.

Contenedor averiado: identifica los contenedores que tengan alguna avería con que cuenta la empresa teniendo en cuenta los contenedores.

Reporte: identifica los tipos de reportes que se pueden generar en el sistema.

Reporte contenedor arrendado: contiene los reportes generados de los contenedores arrendados por un rango de fecha teniendo en cuenta el estándar de los reportes.

Reporte contenedor cliente: contiene los reportes generados de los contenedores por clientes en específico teniendo en cuenta el estándar de los reportes

Reporte contenedor estado: contiene los reportes generados de los contenedores por estado en específico teniendo en cuenta el estándar de los reportes

Reporte contenedor averiado: contiene los reportes generados de los contenedores averiados teniendo en cuenta el estándar de los reportes

2.4 Requisitos del sistema

La definición de los requisitos del sistema permite definir las capacidades del mismo y sobre qué restricciones o condiciones debe trabajar. Son planteados para lograr un entendimiento claro entre el cliente y el equipo de desarrollo, en aras de lograr satisfacer sus verdaderas necesidades (Departamento de Ciencias de la Computación e Inteligencia Artificial, 2008).

2.4.1 Requisitos funcionales

Los requisitos funcionales definen las funcionalidades del sistema. Estos son los que dictan lo que debe realizar el sistema (Departamento de Ciencias de la Computación e Inteligencia Artificial, 2008).

Tabla 2: Requisitos funcionales

| No. | Descripción | Prioridad |
|------|---|-----------|
| RF1 | Incluir contenedor arrendado | Alta |
| RF2 | Modificar datos del contenedor arrendado | Alta |
| RF3 | Eliminar contenedor arrendado | Media |
| RF4 | Ver datos del contenedor arrendado | Media |
| RF5 | Listar contenedores arrendados existentes | Media |
| RF6 | Filtrar contenedores | Media |
| RF7 | Arrendar contenedores | Media |
| RF8 | Registrar devolución de contenedores | Alta |
| RF9 | Ver historial del contenedor arrendado | Media |
| RF10 | Generar reportes de contenedores arrendados | Alta |
| RF11 | Visualizar el reporte de contenedores arrendados | Alta |
| RF12 | Generar reporte de contenedores por estado | Alta |
| RF13 | Visualizar el reporte de contenedores por estado | Alta |
| RF14 | Generar reporte de contenedores por cliente | Alta |
| RF15 | Visualizar el reporte de contenedores por cliente | Alta |

| | | |
|-------------|---|-------|
| RF16 | Incluir reporte de contenedores averiados | Alta |
| RF17 | Modificar datos del reporte de contenedores averiados | Alta |
| RF18 | Eliminar reporte de contenedores averiados | Media |
| RF19 | Ver el reporte de contenedores averiados | Media |
| RF20 | Listar reporte de contenedores averiados | Media |
| RF21 | Filtrar reporte de contenedores averiados | Media |
| RF22 | Visualizar reporte de contenedores averiados | Media |
| RF23 | Incluir orden de liberación | Alta |
| RF24 | Modificar datos de la orden de liberación | Alta |
| RF25 | Eliminar orden de liberación | Media |
| RF26 | Visualizar orden de liberación | Media |
| RF27 | Listar orden de liberación | Media |
| RF28 | Filtrar listado de las órdenes de liberación | Media |
| RF29 | Ver historial de una orden de liberación | Media |

2.4.2 Requisitos no funcionales

Los requisitos no funcionales definen propiedades o cualidades que el sistema debe tener. Estas son las que dictan que el producto sea atractivo, usable, rápido o confiable (Departamento de Ciencias de la Computación e Inteligencia Artificial, 2008).

Para especificar los requerimientos fueron definidas categorías encargadas de recoger las cualidades que debe tener un sistema para su correcto funcionamiento y la prestación adecuada de sus servicios.

Confiabilidad

- Permitir el acceso a la información solo a los usuarios autorizados y denegar el acceso a aquellos que no lo están.
- Establecer un sistema de roles, donde cada rol solo tiene acceso a un conjunto de funcionalidades específicas.

Apariencia

- El módulo debe tener una interfaz con colores adecuados a la línea de proyectos desarrollados con Odo.

Usabilidad

- El módulo utiliza la ayuda de los campos en correspondencia con las líneas de proyectos desarrollados con Odo.
- Permite resaltar los campos obligatorios destacándolos con un color distintivo a la interfaz del módulo.

Portabilidad

- Podrá ser utilizado en cualquier sistema operativo y ejecutarse sobre cualquier navegador, siendo como mínimo compatible con:
 - Google Chrome 10.0 o superior.
 - Firefox 10.0 o superior.
 - Safari 4.1 o superior.

Hardware

- Servidor como mínimo 8Gb de RAM y 200Gb de disco duro.
- Cliente como mínimo 512Mb de RAM y un navegador web.

Operabilidad

- Permitir al administrador del sistema el control total del mismo, además de realizar todas las configuraciones necesarias para que pueda ser utilizado por los restantes usuarios que tendrán acceso a la herramienta.

2.5 Especificación de requisitos

La especificación de requisitos muestra la respuesta que debe tener el sistema, la prioridad y complejidad las cuales son clasificadas como alta, baja y media dependiendo del orden de prioridad y su complejidad.

En la siguiente tabla se muestra la especificación de requisitos para el módulo Arrendamiento de contenedores perteneciente al requisito Incluir contenedor arrendado. Para tener más información sobre las demás especificaciones de requisitos consultar [Anexo #2](#).

Leyenda utilizada en la tabla prioridad para el cliente (PRC), complejidad (C).

Tabla 3: Especificación de requisito Incluir contenedor arrendado

| No. | Nombre | Descripción | PRC | C |
|------|------------------------------|--|------|------|
| RF 1 | Incluir contenedor arrendado | <p>El sistema debe permitir incluir un contenedor, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> - (*) No. del contenedor: identificador único de un contenedor. Campo de texto, admite el siguiente formato cuatro letras seguidas de un espacio, seis números, guion medio (-) y un número. Ejemplo: zscu 859568-8. - (*) Longitud del contenedor: tamaño del contenedor que el corresponsal solicita enviar. Campo de selección simple que incluye: <ul style="list-style-type: none"> • 20 • 40 - (*) Tipo de contenedor: hace referencia a los tipos de contenedores que el cliente solicita. Campo de selección que incluye <ul style="list-style-type: none"> • DV • HC • OP • OS • BLK • FLT • RF • RH • ISO - (*) Proveedor: entidad que arrenda contenedores a TRANSCARGO. Campo de selección. - Estado: situación actual en la que se encuentra el contenedor. Campo de selección que incluye: <ul style="list-style-type: none"> • Alquilado • En reparación • Disponible • Fuera de servicio • Baja • Otros - (*) Ubicación: lugar en el que se encuentra ubicado actualmente el contenedor. Campo de selección que incluye: <ul style="list-style-type: none"> • SERVICOM • Cliente • CADC • Santiago de Cuba • Isla de la Juventud • Otros - Fecha de alquiler: día en el que se realiza | Alta | Alta |

| | | | | |
|--|--|---|--|--|
| | | <p>el alquiler de los contenedores. Campo de selección de fecha, el formato DD/MM/AAAA. Ejemplo 06/11/2015.</p> <ul style="list-style-type: none"> - Fecha de devolución: día en que tiene previsto la devolución de los contenedores. Campo de selección de fecha, el formato DD/MM/AAAA. Ejemplo 09/12/2015 - Cliente: nombre de la entidad que realiza la solicitud para el alquiler de contenedores. Campo de selección. - Observaciones: se describe alguna situación específica relacionada con el contenedor y sus datos. Campo de texto. | | |
|--|--|---|--|--|

2.6 Descripción de requisitos por proceso

A continuación, se presentan la descripción de requisitos por proceso correspondiente al requisito Incluir contenedor arrendado. Para tener información sobre las demás descripciones de requisitos consultar [Anexo #3](#).

Tabla 4: Descripción del requisito por proceso Incluir contenedor arrendado

| | |
|--|--|
| Precondiciones | <p>El usuario debe estar autenticado en el sistema. El usuario debe poseer los permisos para incluir contenedor arrendado.</p> |
| Flujo de eventos | |
| Flujo básico Incluir contenedor arrendado | |
| 1. | El usuario selecciona la opción Crear para Incluir contenedor. |
| 2. | <p>El sistema debe permitir introducir y/o seleccionar los siguientes datos para Incluir Contenedor arrendado:</p> <ul style="list-style-type: none"> - (*) No. del contenedor: identificador único de un contenedor. Campo de texto, admite el siguiente formato cuatro letras seguidas de un espacio, seis números, guion medio (-) y un número. Ejemplo: zscu 859568-8. - (*) Longitud del contenedor: tamaño del contenedor que el corresponsal solicita enviar. Campo de selección simple que incluye: <ul style="list-style-type: none"> • 20 • 40 - (*) Tipo de contenedor: hace referencia a los tipos de contenedores que el cliente solicita. Campo de selección que incluye <ul style="list-style-type: none"> • DV • HC • OP • OS • BLK • FLT • RF • RH • ISO - (*) Proveedor: entidad que arrenda contenedores a TRANSCARGO. Campo de selección. |

| | |
|---|--|
| | <ul style="list-style-type: none"> - Estado: situación actual en la que se encuentra el contenedor. Campo de selección que incluye: <ul style="list-style-type: none"> • Alquilado • En reparación • Disponible • Fuera de servicio • Baja • Otros - (*) Ubicación: lugar en el que se encuentra ubicado actualmente el contenedor. Campo de selección que incluye: <ul style="list-style-type: none"> • SERVICOM • Cliente • CADC • Santiago de Cuba • Isla de la Juventud • Otros - Fecha de alquiler: día en el que se realiza el alquiler de los contenedores. Campo de selección de fecha, el formato DD/MM/AAAA. Ejemplo 06/11/2015. - Fecha de devolución: día en que tiene previsto la devolución de los contenedores. Campo de selección de fecha, el formato DD/MM/AAAA. Ejemplo 09/12/2015 - Cliente: nombre de la entidad que realiza la solicitud para el alquiler de contenedores. Campo de selección. - Observaciones: se describe alguna situación específica relacionada con el contenedor y sus datos. Campo de texto. |
| 3. | Se introducen y/o seleccionan los datos para incluir un contenedor. |
| 4. | El sistema brinda las opciones: <ul style="list-style-type: none"> • Guardar. • Descartar |
| 5. | El usuario selecciona el opción Guardar. |
| 6. | El sistema <i>valida los datos</i> |
| 7. | El sistema <i>crea</i> el contenedor y muestra el siguiente mensaje de información: <i>El contenedor se ha incluido satisfactoriamente.</i> |
| 8. | Concluye así el requisito. |
| Pos-condiciones | |
| 1. | Se incluyó un nuevo contenedor satisfactoriamente. |
| Flujos alternativos | |
| Flujo alternativo 6.a Información incompleta | |
| 1. | El sistema señala el o los campos obligatorios que no hayan sido introducidos y/o seleccionados y muestra debajo del campo el mensaje de error: <i>Campo obligatorio</i> |
| 2. | El usuario introduce y/o selecciona los datos. |
| 3. | Volver al paso 6 del flujo básico |
| Pos-condiciones | |
| 1. | NA |
| Flujos alternativo 6.b Información errónea | |
| 1. | El sistema señala el o los campos incorrectos y muestra debajo del campo el mensaje de error: <i>Campo incorrecto</i> |
| 2. | El usuario introduce y/o selecciona correctamente los datos. |
| 3. | Volver al paso 6 del flujo básico. |
| Pos-condiciones | |
| | N/A |

| Flujo alternativo 4.a Descartar cuando se hayan introducido datos | |
|--|---|
| 1. | El usuario selecciona la opción <i>Descartar</i> |
| 2. | El sistema muestra un mensaje de información: <i>Se ha modificado la entidad, sus cambios se perderán</i> y solicita confirmación <i>¿Está seguro que quiere abandonar la página?</i> |
| 3. | El sistema brinda las siguientes opciones: <ul style="list-style-type: none"> • <i>Aceptar</i> • <i>Cancelar</i> |
| 4. | El usuario selecciona la opción <i>Aceptar</i> |
| 5. | El sistema elimina los datos introducidos y regresa al listado de contenedores. |
| 6. | Concluye así el requisito. |
| Pos-condiciones | |
| 1. | No se crea la entidad contenedor. |
| Flujo alternativo 4.b Descartar cuando se hayan introducido datos opción Cancelar | |
| 1. | El usuario selecciona la opción <i>Cancelar</i> |
| 2. | El sistema mantiene los cambios realizados. |
| 3. | Volver al paso 4 del flujo básico. |
| Pos-condiciones | |
| 1. | N/A |
| Flujo alternativo *Descartar | |
| 1. | El usuario selecciona el botón <i>Descartar</i> |
| 2. | El sistema elimina los datos creados, regresa a la interfaz anterior y muestra el mensaje de información: <i>La acción ha sido cancelada.</i> |
| 3. | Concluye así el requisito. |
| Pos-condiciones | |
| 1. | No se crea el contenedor arrendado |
| Validaciones | |
| 1. | NA |
| Conceptos | NA NA |
| Requisitos especiales | |
| Asuntos pendientes | NA |

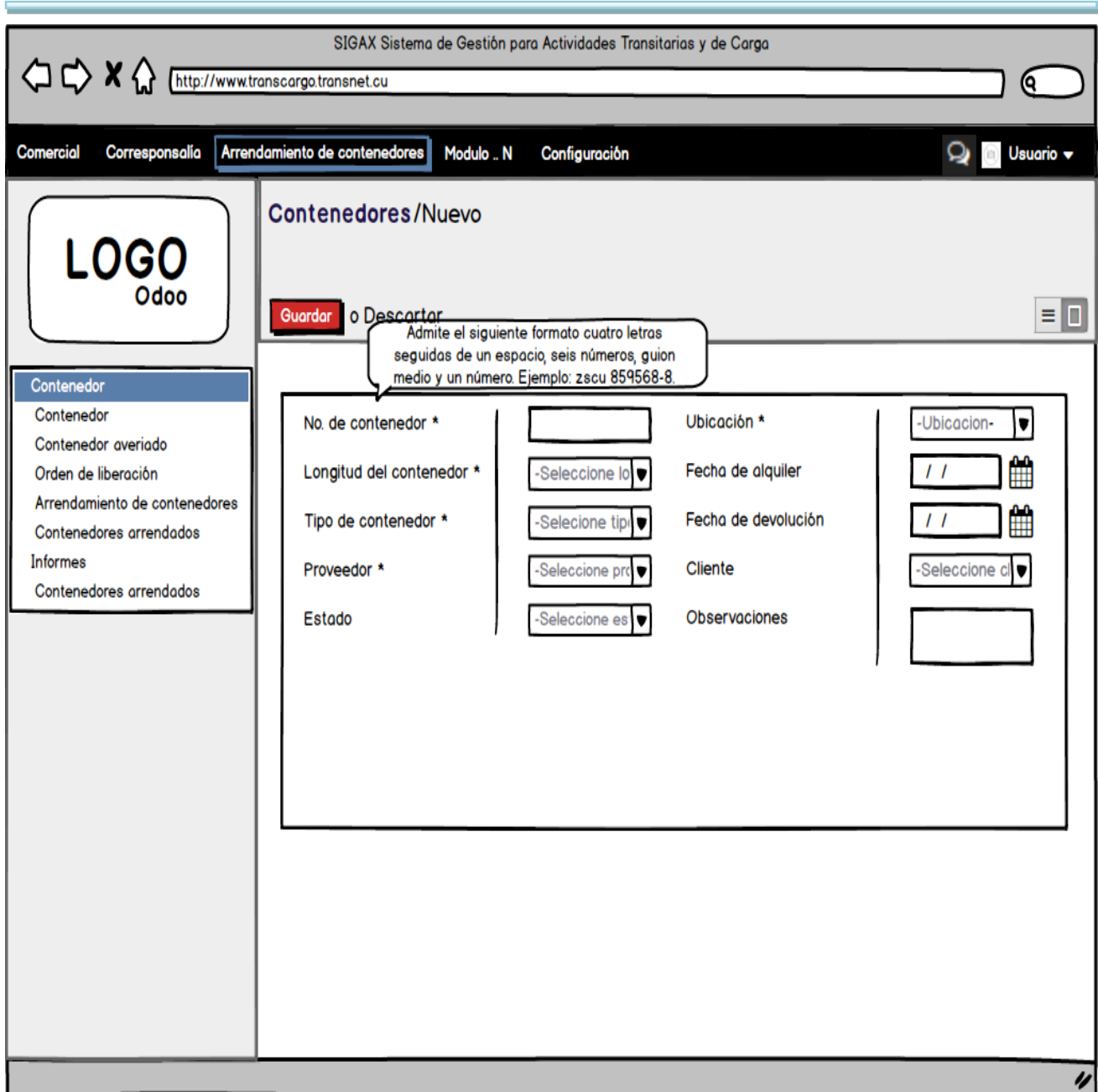


Figura 3: Prototipo interfaz de usuario del requisito Incluir contenedor arrendado

SIGAX Sistema de Gestión para Actividades Transitorias y de Carga

http://www.transcargo.transnet.cu

Comercial Corresponsalia **Arrendamiento de contenedores** Modulo .. N Configuración

Usuario

Los siguientes campos son obligatorios:
No. de contenedor
Proveedor

Contenedores/Nuevo

Guardar o Descartar

Contenedor

- Contenedor
- Contenedor averiado
- Orden de liberación
- Arrendamiento de contenedores
- Contenedores arrendados
- Informes
- Contenedores arrendados

| | | | |
|---------------------------|----------------------|---------------------|----------------------|
| No. de contenedor * | <input type="text"/> | Ubicación * | SERVICOM |
| Longitud del contenedor * | 40 | Fecha de alquiler | 16/11/2015 |
| Tipo de contenedor * | RH | Fecha de devolución | 25/11/2015 |
| Proveedor * | -Seleccione pro | Cliente | ETECSA |
| Estado | Disponible | Observaciones | <input type="text"/> |

Figura 4: Prototipo interfaz de usuario del requisito Incluir contenedor arrendado (Flujo alterno información incompleta)

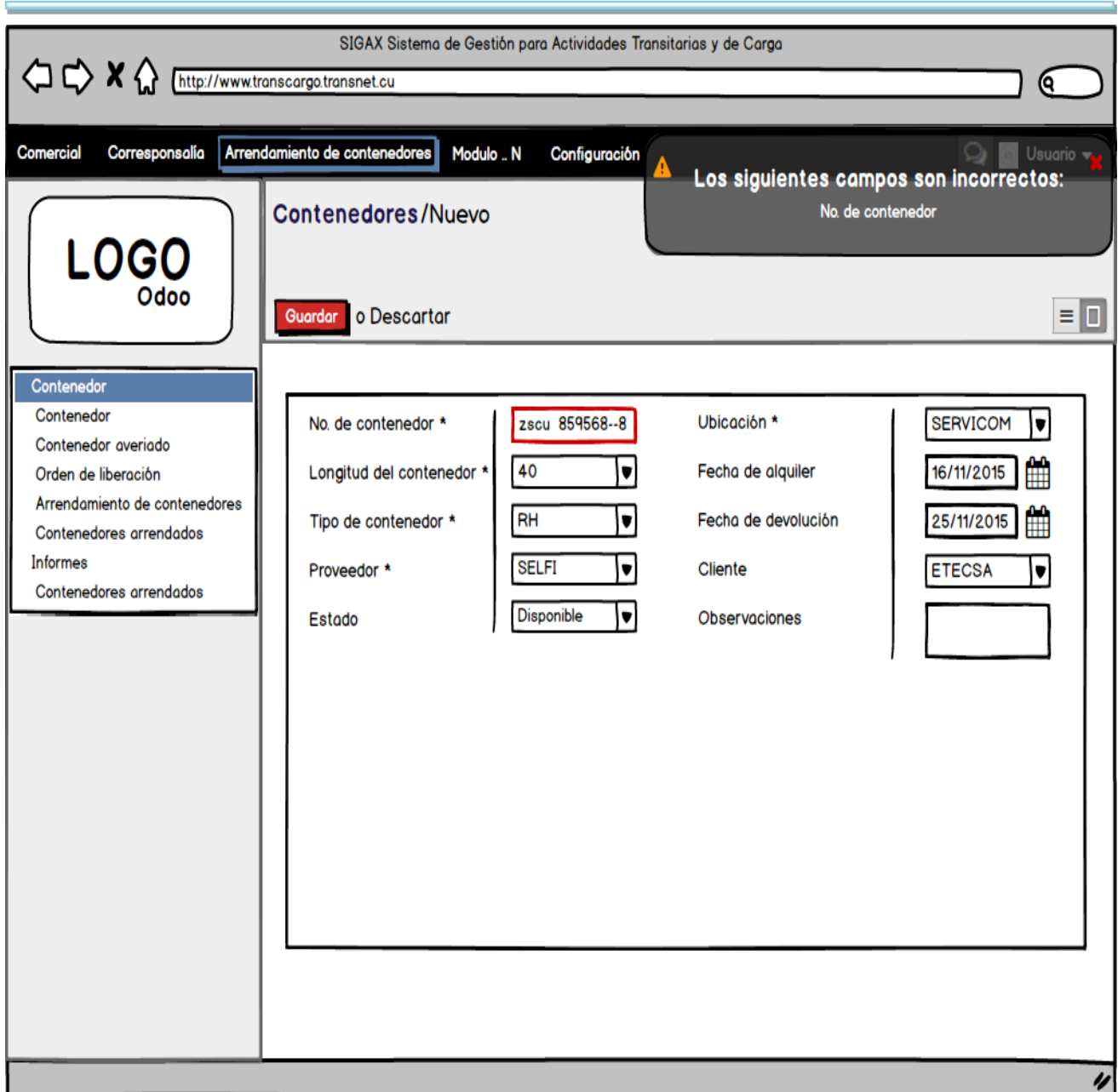


Figura 5: Prototipo interfaz de usuario del requisito Incluir contenedor arrendado (Flujo alterno información errónea)

2.7 Arquitectura de Odoo

Cliente/Servidor es una arquitectura distribuida que permite a los usuarios obtener acceso a la información aún en entornos multiplataforma. El esquema de funcionamiento de un sistema Cliente-Servidor es (Oposiciones TIC, 2011):

1. El cliente solicita información al servidor y este recibe la petición.
2. El servidor procesa dicha solicitud y envía el resultado al cliente.
3. El cliente recibe el resultado y lo procesa.

Los elementos principales de esta arquitectura son (Oposiciones TIC, 2011):

- Cliente: Es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor. Encargado de las funciones relacionadas con la manipulación y despliegue de datos, además de acceder a los servicios desde cualquier parte de la red.
- Servidor: Es el proceso que permite atender a múltiples clientes que hacen peticiones de algún recurso. Encargado de las funciones relacionados con la mayoría de las reglas del negocio y los recursos de datos.
- Middleware: Es un módulo intermedio que actúa como mediador entre sistemas. Es el intermediario entre el cliente y el servidor y se ejecuta en ambas partes.

La arquitectura de trabajo de Odoo está basada en Cliente/Servidor, esta tiene los componentes de la misma separados. El servidor se ejecuta separado del cliente y maneja la lógica del negocio y realiza el enlace entre la aplicación y la base de datos. El cliente es el que presenta la información a los usuarios y permite interactuar con el servidor (Bargaw, 2014).

- Servidor y módulos: la parte del servidor está escrita en Python y el cliente se comunica con él a través de interfaces XML-RPC. La parte del negocio se agrupa en los módulos que no son más que carpetas con estructura predefinida, con código Python y archivos XML. Estos definen la estructura de los datos, formularios, menús, procedimientos. Los módulos se conforman con una sintaxis independiente del cliente, lo que permite que al crear nuevos objetos lo realizas para todos los clientes.
- Clientes: estos son rápidos ya que no cuentan con la parte del negocio y muestra los datos en una aplicación web implementado como un servidor HTTP, permitiendo a los usuarios conectarse mediante un navegador web.

A continuación, se puede apreciar la representación de la arquitectura Cliente/Servidor en Odoo.

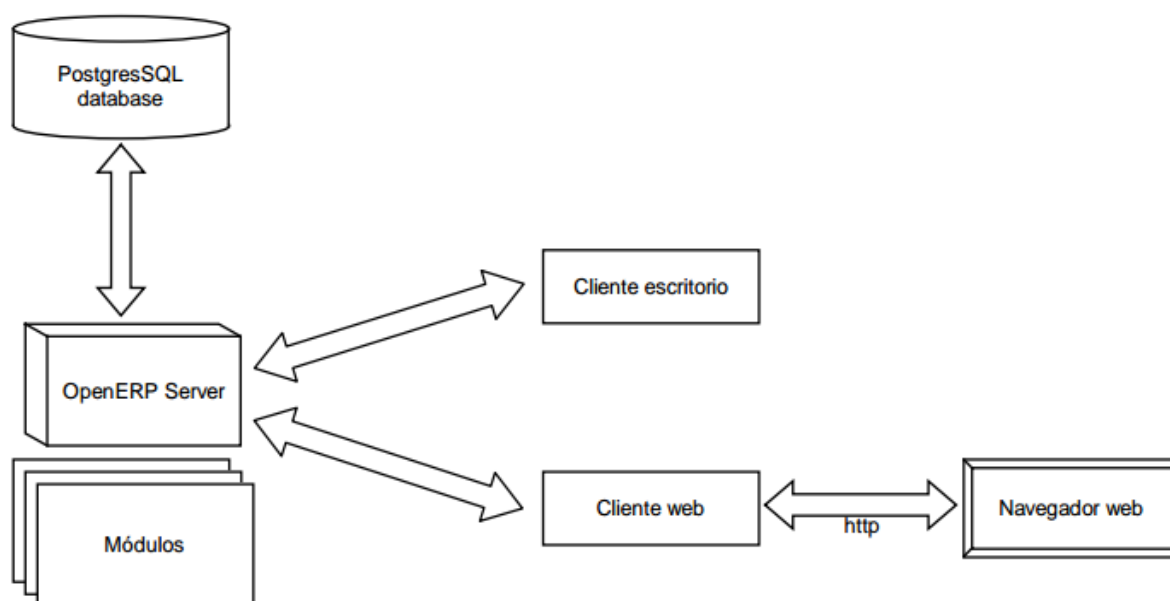


Figura 6: Arquitectura de Odoo (CALVO LOPEZ, 2010)

2.8 Patrones arquitectónicos

El Modelo-Vista-Controlador (MVC) es un patrón de arquitectura de software que separa la lógica de la aplicación de la lógica de la vista de una aplicación, garantizando así la actualización y mantenimiento de forma sencilla de la misma y en un corto espacio de tiempo (Patrón Modelo-Vista-Controlador, 2012).

Los elementos del patrón son los siguientes (Patrón Modelo-Vista-Controlador, 2012):

- Modelo: es el encargado de los datos, es decir, generalmente consultando la base de datos.
- Vista: es la representación visual de cómo se van a ver los datos, es decir, la información relacionada con las interfaces se incluye en este elemento.
- Controlador: es el encargado de controlar, es decir, recibe órdenes del usuario y es el encargado de pedir los datos al modelo y comunicárselos a la vista.

A continuación, se muestra una representación del funcionamiento del patrón MVC

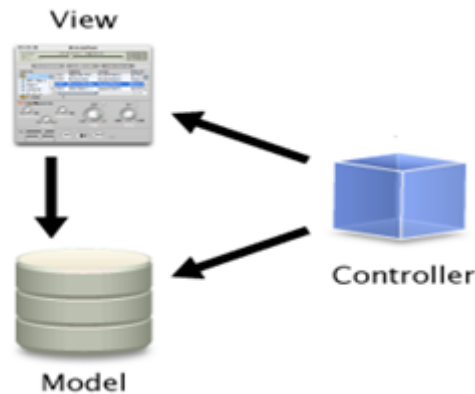


Figura 7: Modelo Vista Controlador

El *framework* OpenObject que se utiliza en Odoo utiliza este patrón aprovechando al máximo todas las características del mismo donde:

- Modelo (*model*): los objetos de Odoo con sus columnas que normalmente se guardan en las tablas de PostgreSQL con sus campos. Permite la creación/actualización automática de las tablas y acceder a las mismas sin utilizar SQL.
- Vista (*view*): listas, formularios, calendarios, gráficos, definidas en archivos XML. En estos archivos también se definen, menús, acciones, informes, asistentes.
- Controlador (*controller*): métodos Python definidos dentro de los objetos de Odoo que proporcionan la lógica: validación de datos, cálculos.

2.9 Modelo de diseño

El modelo de diseño crea una representación o modelo de *software* enfocado en la representación de los datos, funciones y el comportamiento, permite modelar el sistema que se va a construir posibilitando evaluar su calidad y mejorarlo antes de generar código. Mediante este se obtiene una representación arquitectónica y despliegue del sistema (LARMAN, y otros, 2003).

2.9.1 Patrones de diseño

Un patrón de diseño es una solución estándar para un problema común de programación. En un software contribuye a la reutilización de su diseño, ellos nombran, identifican los aspectos claves de su estructura (LARMAN, 1999). Estas características garantizan una reducción de tiempo en el desarrollo y mantenimiento. Además, mejora la seguridad, eficiencia y consistencia de los datos.

Para el desarrollo del módulo se tuvo en cuenta los patrones de asignación de responsabilidades, conocidos como GRASP acrónimo de “*General Responsibility Assignment Software Patterns*”, los cuales tienen como objetivo fundamental orientar al diseñador en cómo asignar las

responsabilidades a cada clase en diferentes circunstancias (LARMAN, 1999).

Además, se tuvo en cuenta los patrones GOF (Pandilla de los Cuatro) acrónimo de “Gang of Four”, los cuales constituyen un catálogo de 23 patrones de diseño publicados por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides en el libro *Design Patterns: Elements of Reusable Object Oriented Software* (BEDOYA, 2011) los cuales se clasifican en dependencia del propósito para los que hayan sido utilizados.

Patrones GRASP

A continuación, se muestran los patrones GRASP utilizados en la solución propuesta, con una descripción y ejemplo de donde fueron utilizados.

- Experto: sugiere asignar responsabilidades a las clases de modo que estas contengan la información necesaria para ejecutar una acción específica. Aplicando este patrón se mantiene el encapsulamiento, los objetos utilizan su propia información para realizar sus tareas. Hace que el sistema sea más fácil de entender y mantener. Se evidencia en la clase orden de liberación.
- Bajo acoplamiento: plantea que las clases deben estar lo menos ligadas entre sí que sea permisible. Permite que se pueda modificar alguna clase teniendo la menor repercusión posible en el resto del sistema. Se evidencia en la clase contenedor averiado.

Patrones GOF

Los patrones GOF utilizados son:

- Singleton: pertenece a la familia de patrones creacional, la cual refiere a la creación de objetos. Su propósito consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso a ella. Analizando los módulos de Odo como objetos, pueden ser pensados como una clase con una instancia única, permite identificar este patrón. Estos objetos no encapsulan datos, lo que separa un módulo de otro es el conjunto de funciones que contiene, permitiendo así la existencia de una única instancia para un módulo.
- Observer: pertenece a la familia de patrones de comportamiento, la cual refiere a definir la relación e iteración de los objetos del sistema. Su propósito consiste en definir una dependencia de unos a muchos entre objetos, permitiendo que se actualicen todos los objetos que dependen de él.

```
files = fields.One2many('ir.attachment', 'name',_('Annexes'), store=True,  
help=_('Evidences of the damage'))
```

2.9.2 Diagrama de clase del diseño

Un diagrama de clases del diseño (DCD) representa las especificaciones de las clases e interfaces de software en una aplicación. Contiene una mayor cantidad de detalles entre las que se encuentran: clases, atributos, operaciones, subsistemas y relaciones (LARMAN, y otros, 2003). Se muestra a continuación el diagrama de clases de diseño Gestionar contenedor arrendado perteneciente al módulo Arrendamiento de contenedores, los demás se muestran en el [Anexo #4](#). Donde el gestionar hace referencia a la integración de los requisitos: modificar, ver, listar y eliminar del contenedor arrendado.

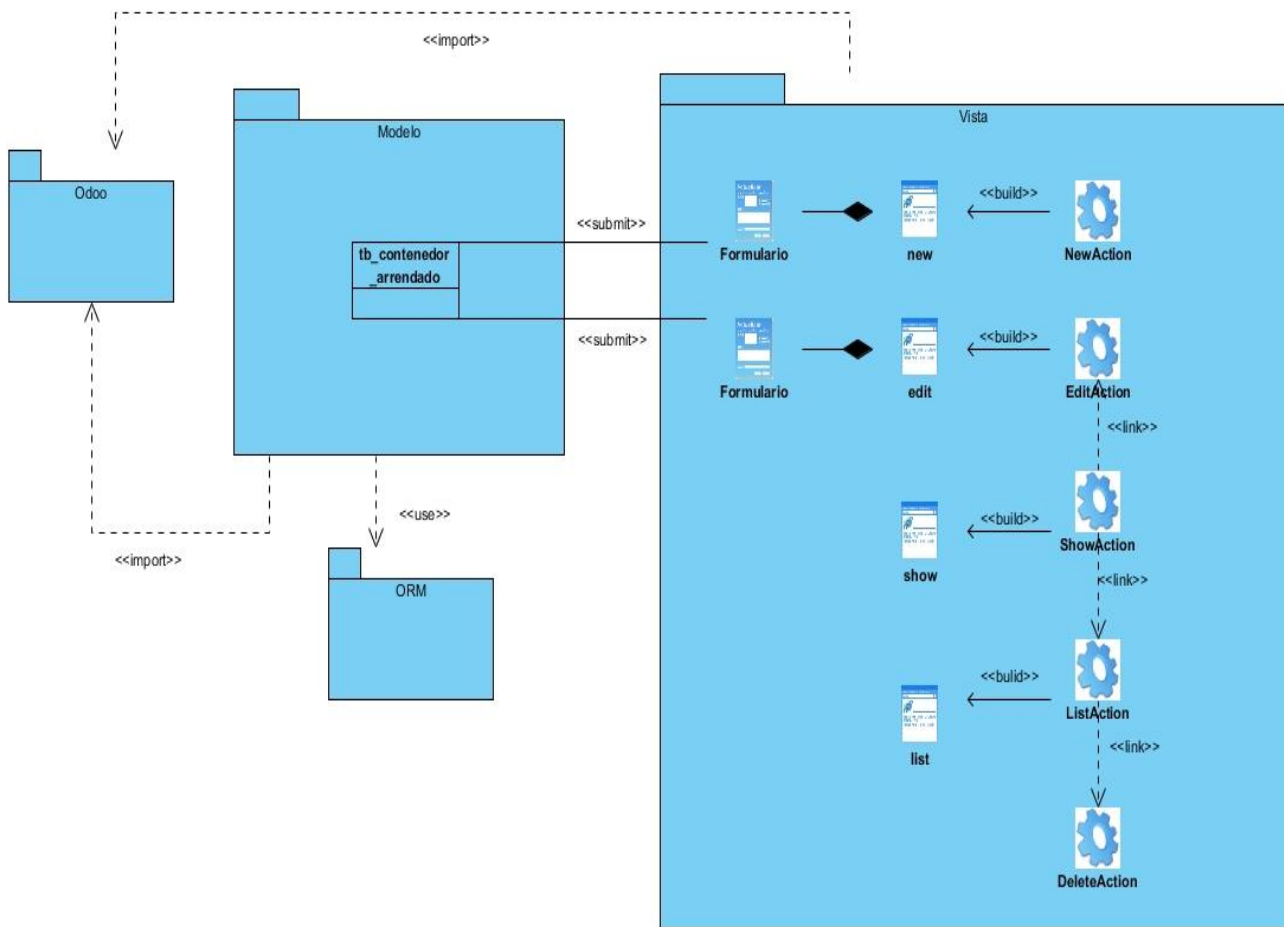


Figura 8: Diagrama clase de diseño Gestionar contenedor arrendado

2.9.3 Diagrama de secuencia del diseño

El diagrama de secuencia (DS) muestra cómo se pasa de un objeto a otro a medida que se realizan las operaciones relacionándose a través de mensajes. Muchas veces los desarrolladores utilizan textos para explicar cómo interactúan los objetos de diseño para llevar a cabo flujo de eventos (LARMAN, y otros, 2003). A continuación se muestra el diagrama de secuencia correspondiente a Incluir contenedor arrendado, los demás diagramas se muestran en el [Anexo #5](#).

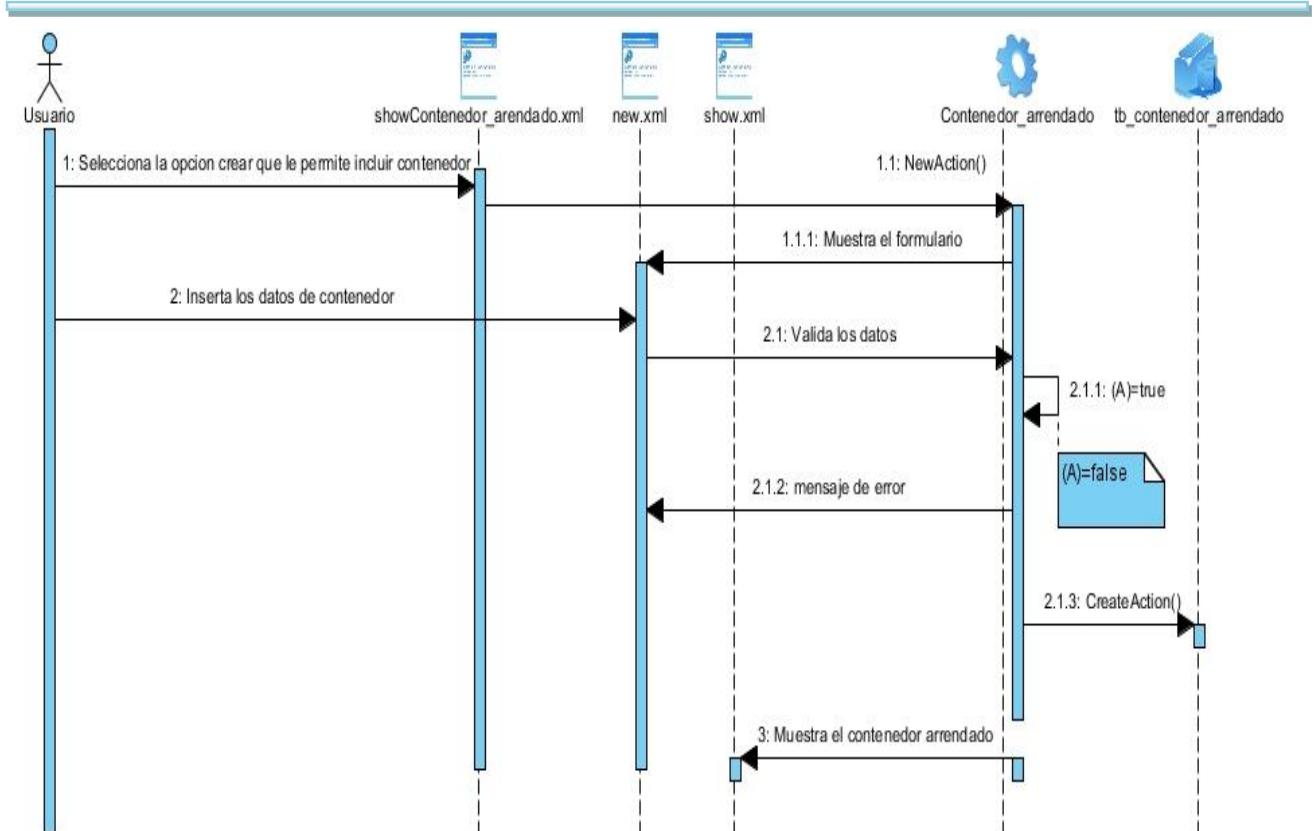


Figura 9: Diagrama de secuencia de diseño del requisito Incluir contenedor arrendado

2.10 Modelo de datos

Las bases de datos son de gran importancia en la concepción y el desarrollo del módulo, ya que estas permiten almacenar, organizar y manipular gran cantidad de información. A continuación, se muestra el modelo de datos de la propuesta de solución diseñado utilizando el Visual Paradigm 8.0.



Figura 10: Diagrama modelo de datos del módulo Arrendamiento de contenedores

2.10.1 Descripción de las tablas de la base de datos

A continuación, se muestra la descripción de la tabla container_leasing_container para almacenar los datos de los contenedores arrendados en el módulo. La descripción de las restantes tablas se muestra en el [Anexo #6](#).

Tabla 5: Descripción de la tabla container_leasing_container

| container_leasing_container | | |
|---|-------------|---|
| Descripción: Esta tabla es la encargada de almacenar los datos de los contenedores arrendados. | | |
| Atributo | Tipo | Descripción |
| id | Integer(4) | Campo que contiene el identificador del contenedor. |
| container_number | Varchar(13) | Campo que contiene el número del contenedor. |
| length | Integer(4) | Campo que contiene la longitud del contenedor. |
| type | Varchar(10) | Campo que contiene el tipo de contenedor. |

| | | |
|--------------|--------------|---|
| state | Varchar(255) | Campo que contiene el estado del contenedor. |
| location | Varchar(255) | Campo que contiene el lugar donde se encuentra el contenedor. |
| rental_date | Date | Campo que contiene la fecha de arrendamiento del contenedor. |
| return_date | Date | Campo que contiene la fecha de devolución del contenedor. |
| observations | Varchar(255) | Campo que contiene alguna observación sobre el contenedor. |
| deleted | Boolean | Campo que permite eliminar la visibilidad de un contenedor. |
| client_id | Integer(4) | Campo que contiene el identificador del cliente. |
| provider_id | Integer(4) | Campo que contiene el identificador del proveedor. |

2.11 Conclusiones parciales

El proceso de descripción y modelación del módulo mediante la metodología AUP variación UCI contribuyó a una mejor organización de los procesos y las tareas a realizar. Con el uso de esta metodología se generaron los siguientes artefactos, descripción del proceso de negocio, la descripción y especificación de requisitos, permitiendo conocer las funcionalidades y características de la propuesta de solución y el modelo de datos asociados al módulo. Se diseñaron los prototipos de interfaz de usuario que representan un punto de inicio para el proceso de implementación. La arquitectura y los patrones de diseño facilitaron la organización en el desarrollo, así como la optimización de tiempo y esfuerzo del desarrollo gracias a la reutilización de código. Toda esta documentación facilitará el proceso de implementación que se describe en el próximo capítulo.

Capítulo 3: Implementación y prueba

Introducción

Antes de comenzar el desarrollo de un sistema, es necesario que dicho proceso sea guiado por un conjunto de estándares y patrones con el objetivo de lograr una línea base, para que todos los desarrolladores que trabajen con el sistema puedan entender el funcionamiento del mismo. Es por eso que en el presente capítulo se desarrolla una descripción de la implementación del módulo. Una vez terminado el proceso de implementación se describen las pruebas realizadas al módulo Arrendamiento de contenedores, para validar la correcta implementación de los requisitos y verificar la calidad del mismo.

3.1 Diagrama de componentes

Los diagramas de componentes representan la estructura física del código, la ubicación de los archivos de código fuente. Se encargan de asignar la vista lógica de las clases del proyecto de archivos que contienen el código fuente en el que se implementa la lógica. Están compuestos por componentes (bloque de construcción física del sistema), interfaces (elemento que describe un grupo de operaciones usadas o creadas por componentes) y dependencias (relaciones entre componentes), que pueden ser organizados en una estructura de paquetes (BRESSANO, 2010). A continuación, se muestra la estructura de los componentes que forman parte de la solución propuesta.

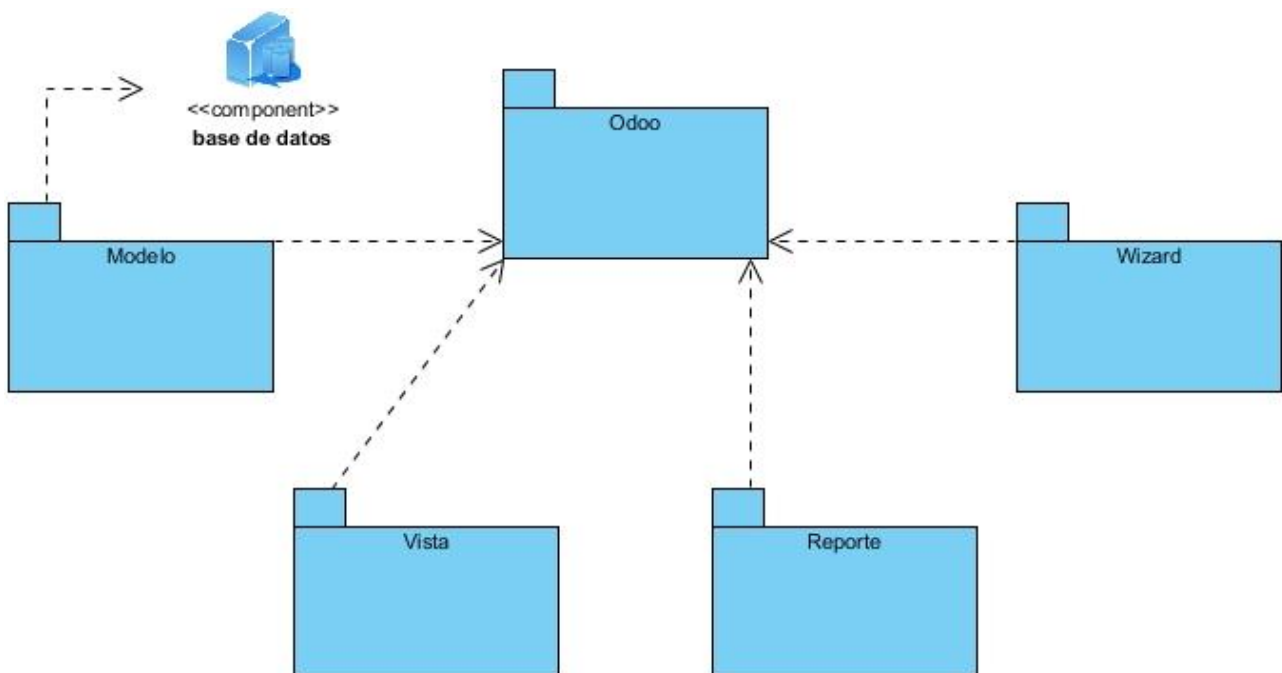


Figura 11: Diagrama de componentes del módulo Arrendamiento de contenedores

3.1.1 Descripción de los componentes

Paquete de componentes “Vistas”

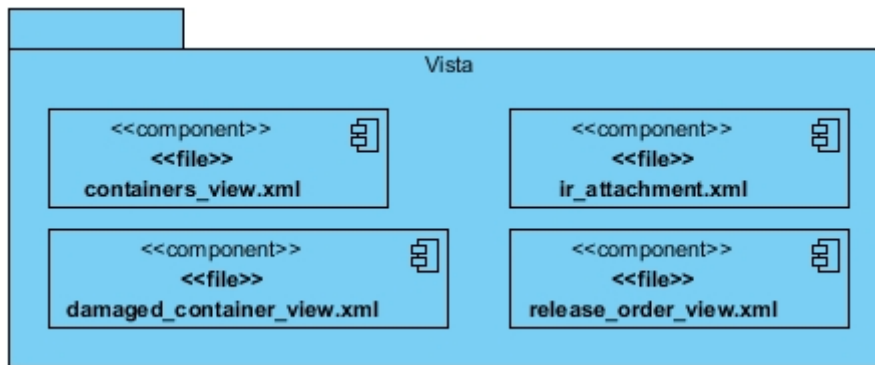


Figura 12: Paquete Vista

El paquete de componente “Vista” es una representación de las páginas con las cuales el usuario interactúa, permitiendo la entrada de datos al sistema y realizar consultas de dichos datos.

Paquete de componentes “Odoos”

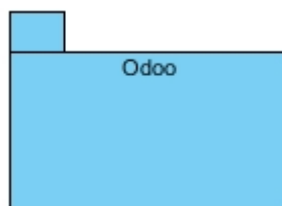


Figura 13: Paquete Odoos

El paquete de componentes “Odoos” es una representación de las operaciones que permiten el flujo y la comunicación entre las vistas, el modelo, reporte y wizard; es el encargado de gestionar todo el flujo de datos mediante direcciones HTTP y posteriormente actualizar la vista para que el usuario pueda ver cómo ha quedado el modelo de datos tras las actualizaciones.

Paquete de componentes “Modelo”

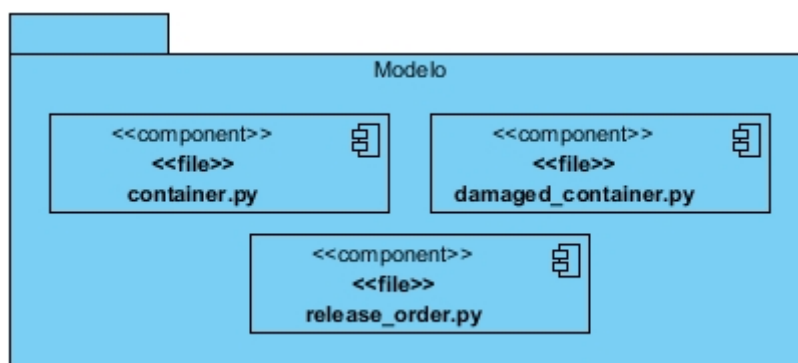


Figura 14: Paquete Modelo

Son las clases que representan todo el negocio del sistema. Estas clases constituyen tablas que son utilizadas en la base de datos mapeadas por el ORM, contiene todos los métodos y acciones relacionados con estas tablas.

Paquete de componentes “Reporte”

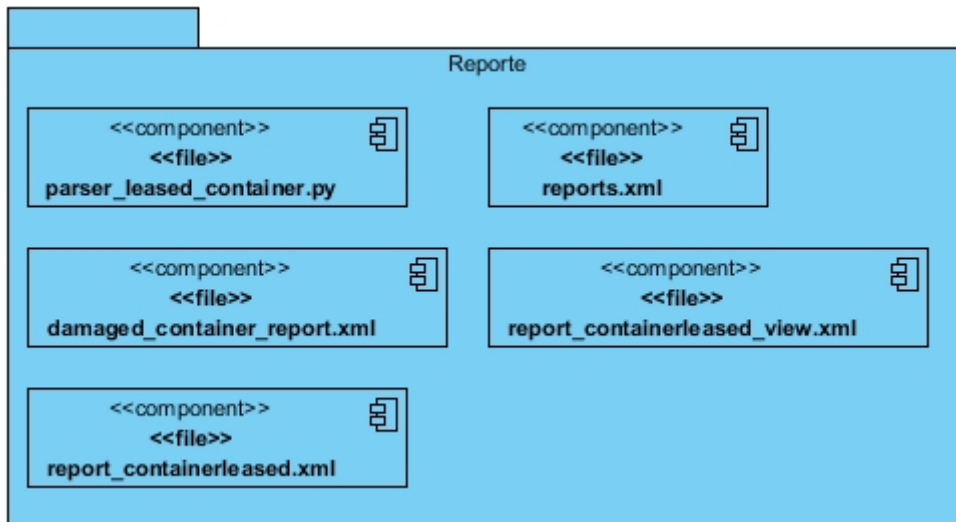


Figura 15: Paquete Reporte

Este paquete es utilizado para la creación de reportes. Contiene los modelos basados en vistas SQL y los archivos Python y XML.

Paquete de componentes “Wizard”

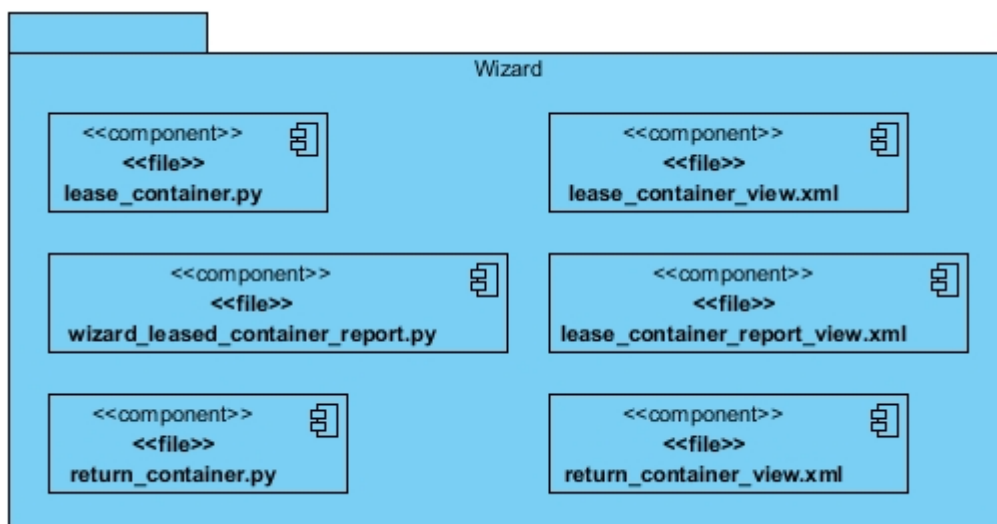


Figura 16: Paquete Wizard

Este paquete es utilizado para los datos que son transitorios. Contiene los modelos y vistas asociados a estos datos.

3.2 Diagrama de despliegue

El diagrama de despliegue se utiliza para modelar la disposición física de los componentes de hardware utilizando la implementación del sistema y la relación entre cada uno de ellos (PRESSMAN, 2006).

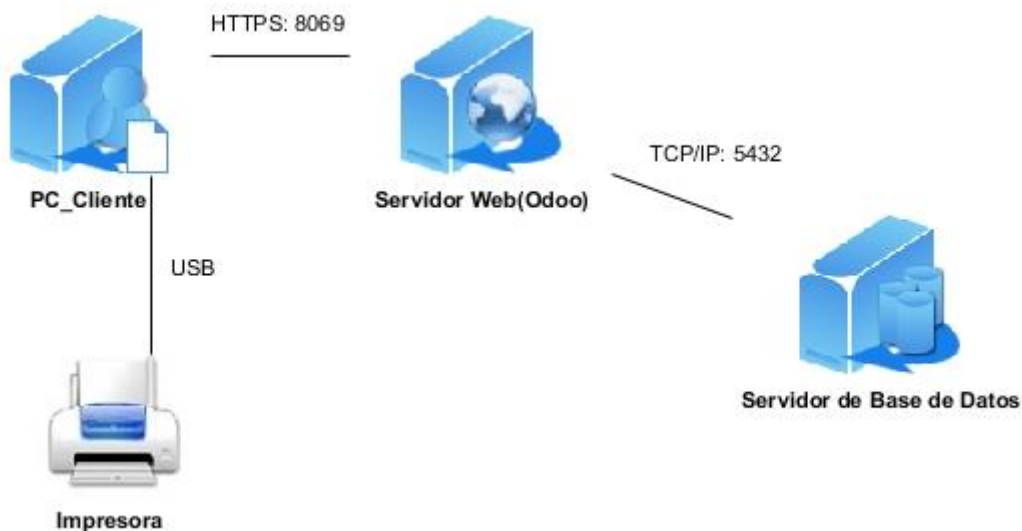


Figura 17: Diagrama de despliegue

3.3 Estándar de código

Con el objetivo de lograr que otros programadores comprendan el código (cómo identificar las variables, las funciones o métodos) se hace necesario el establecimiento de un estándar de codificación, para obtener un producto final con mayor calidad. En general, un estándar de codificación es un conjunto de reglas que se siguen para la escritura del código fuente. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones del código de rutinas. La creación de módulos en Odoo según su página oficial sigue una determina estructura a continuación se muestra la estructura de un módulo y los estándares de código en los lenguajes XML y Python.

3.3.1 Estructura del módulo

La estructura de un módulo de Odoo cuenta de un directorio con varias carpetas encargadas de guardar los archivos necesarios para su funcionamiento. A continuación se muestra el directorio de carpetas propuesto por los desarrolladores de Odoo para un módulo (Odoo):

- Data: contiene los datos de prueba del módulo.
- Models: contiene todos los archivos relacionados con los datos del módulo.

- **Controllers:** contiene los archivos de las rutas http.
- **Views:** contiene las vistas y las plantillas del módulo.
- **Static:** contiene los elementos web, separados en `css` /`js`, /`lib`, /`img` entre otros directorios.

Para la realización del módulo de la propuesta de solución se tuvo en cuenta este directorio de carpetas agregando una carpeta de `report`, `security`, `wizard` y `i18n` al directorio, primeramente, se expone el contenido de las nuevas carpetas como se muestra a continuación:

- **Wizard:** contiene los modelos transitorios (`osv_memory`) y sus vistas asociadas.
- **Report:** contiene los reportes, modelos basados en las vistas SQL (para presentar los reportes) y otros reportes complejos. Incluye los archivos Python y XML asociados.
- **Security:** contienen los archivos para agregar los grupos de usuarios y sus respectivos permisos a las diferentes funcionalidades del módulo.
- **I18n:** contiene el archivo de traducción al idioma español del módulo.

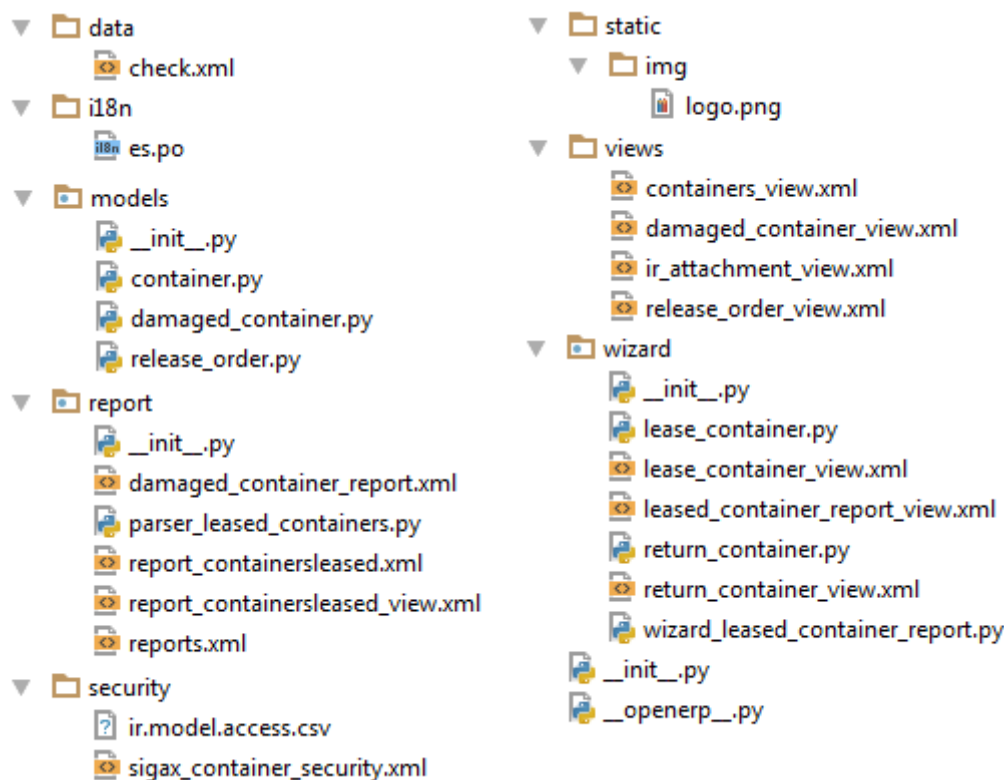


Figura 18: Estructura del módulo de la propuesta de solución

3.3.2 Archivos XML

Se define un estándar de código siguiendo las pautas establecidas por el proyecto para la

implementación de módulos:

- Para la declaración de <record> se debe colocar un id antes de colocar model. En la declaración de los *fields* comienza con el nombre del campo, seguido del valor de la etiqueta y otros atributos.
- En torno a las vistas y menús se declaran el nombre del menú seguido de underscore y la palabra menú y en las vistas en nombre de la vista seguido de la palabra *view*, underscore y el tipo de vista que puede ser tree, form.
- Las sentencias de código demasiado largas deben ser vistas sin necesidad de utilizar desplazamiento horizontal por lo que es necesario dividir las en tantas líneas como sean necesario.

Ejemplo de código:

```
<record id="view_damaged_container_tree" model="ir.ui.view">
  <field name="name">damaged_container_tree</field>
  <field name="model">sigax.containerleasing.damagedcontainer</field>
  <field name="type">tree</field>
  <field name="arch" type="xml">
    <tree string="Damaged container" colors="red:deleted == True" fonts="italic:deleted ==
True">
      <field name="container_number"/>
      <field name="date_of_issue"/>
      <field name="client_id"/>
      <field name="deleted" invisible="True"/>
    </tree>
  </field>
</record>
<menuitem name="Damaged container"
id="menu_damaged_container"
action="action_damaged_container"
parent="menu_containers"
sequence="20"/>
```

3.3.3 Archivos Python

Siguiendo las pautas de programación establecidas en el proyecto se define un estándar de codificación:

- No debe haber líneas muy largas sin la necesidad del desplazamiento horizontal y los comentarios de bloque debe comenzar con '#'.
- Importar archivos: primero las librerías de Python, luego los elementos de Odoo que se van utilizar y por último los módulos de Odoo si solo son necesarios.

- La declaración de clases y variables usan el estilo “encamellado” (CamelCase).
- Las cadenas de caracteres deben definirse utilizando comillas simples, siempre que sea necesario, para obtener un mejor rendimiento.
- Declaración de variables usan el estilo “encamellado” (CamelCase) para las del modelo y la notación en minúsculas para las variables comunes.

Ejemplo de código:

```
import re
from openerp import fields, models, api, exceptions
from openerp import SUPERUSER_ID
from datetime import date

class DamagedContainer(models.Model):
    _name = 'sigax.containerleasing.damagedcontainer'
    _rec_name = 'container_number'
    _inherit = 'mail.thread'

    container_number = fields.Char('Container number', size=80, required=True,
    help='It supports the following format: four letters followed by a space, '
    'six numbers, dash (-) and a number.')
    date_of_issue = fields.Date('Date of issue', required=True, default=fields.Date.today,
    help='Select the date the report was issued')
    client_id = fields.Many2one('sigax.commercial.entity', 'Client', required=True,
    help='Name of the entity that had leased the damaged container.')
    damages = fields.Text('Damages', size=80, required=True, help='Description of the damage
    that has the container.')
    conclusions = fields.Text('Conclusions', size=80, required=True,
    help='Result of decisions made on the container damaged.')
    files = fields.One2many('sigax.containerleasing.files', 'file', 'Annexes', store=True,
    help='Evidences of the damage')
    deleted = fields.Boolean('Deleted', default=False)
```

3.4 Seguridad

A diario, las empresas se encuentran amenazadas por riesgos que ponen en peligro la integridad de la información y con ello la viabilidad de los negocios. Riesgos que provienen no solo desde el exterior de las empresas, sino también desde el interior. Para poder trabajar de forma inequívoca es necesario aplicar políticas de seguridad informática que mantendrán los datos protegidos.

Para garantizar el funcionamiento correcto de la propuesta de solución se establecen como políticas de seguridad la autenticación de las personas autorizadas y previamente registradas en el sistema, así como la administración de la base de datos por parte del administrador de la aplicación (MENDOZA, 2015).

En cuanto a la programación del módulo, los mensajes de error que se le muestran al usuario, se

elaboran respetando el criterio de ofrecer la mínima información. Utilizando UseError que se define en `openerp.exceptions` en lugar de la advertencia de primer orden. Cuando se genera un módulo con el *framework* que utiliza Odoo, automáticamente se activan medidas de seguridad para hacerle frente algunas de las vulnerabilidades más extendidas en la web como: los ataques XSS³ y SQL Injection⁴. Para evitar los ataques tipo SQL Injection el *framework* de Odoo se basa en ORM que se abstrae de la generación de las consultas SQL ya que los desarrolladores no generan las secuencias SQL sino mediante el ORM, por lo que los pasan de manera apropiada evitando así por defecto este tipo de ataque. Por su parte, para evitar los ataques XSS el *framework* de Odoo mediante el uso de un sistema de plantillas de alto nivel que evita de forma automática todos los datos que queden.

3.5 Pruebas a los sistemas informáticos

La prueba de *software* es un elemento de un tema más amplio que suele llamarse verificación y validación. Donde la verificación es el conjunto de actividades que aseguran que el *software* implemente correctamente una función específica. Validación es un conjunto diferente de actividades que aseguran que el *software* construido corresponde con los requisitos del cliente (PRESSMAN, 2005).

Algunos de los objetivos de las pruebas de *software* según Pressman son:

- La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta el momento.

Las pruebas permiten ver hasta qué punto las funciones de *software* parecen funcionar de acuerdo con las especificaciones. Los datos que van generando las pruebas son un buen indicador de la fiabilidad del mismo, indicando la calidad en conjunto. De esta manera las pruebas que se llevan a cabo con el propósito de encontrar errores y fallas, su razón principal es que el sistema falle o tenga errores con el mínimo de tiempo y recursos.

La metodología AUP-UCI propone realizar una serie de pruebas a los productos realizados en la

³ XSS (Cross Site Scripting): es un ataque de inyección de código malicioso para su posterior ejecución que puede realizarse a sitios web, aplicaciones locales e incluso al propio navegador.

⁴ SQL Injection: se refiere a un ataque de inyección en el que un atacante puede ejecutar sentencias SQL maliciosos (también conocidos comúnmente como una carga maliciosa) que controlan el servidor de base de datos de una aplicación web.

Universidad de las Ciencias Informáticas (UCI), estas son pruebas internas, pruebas de liberación y pruebas de aceptación (SANCHEZ RODRIGUEZ, 2015). A continuación, se describen las pruebas realizadas al módulo Arrendamiento de contenedores para el Sistema de Gestión Integral del Agente Transitario de Cargas fueron las pruebas internas y las pruebas de aceptación:

- **Pruebas internas:** el objetivo de estas pruebas es verificar el resultado de la implementación probando cada construcción, incluyendo tanto las internas como las intermedias; así como las versiones finales a ser liberadas (SANCHEZ RODRIGUEZ, 2015).
- **Pruebas de aceptación:** el objetivo fundamental de estas pruebas es verificar que el sistema está listo y que puede ser usado por parte de los usuarios finales. Los criterios de aceptación deben establecerse en el momento en que las partes aceptan contraer contrato. Son ejecutadas normalmente por el cliente aunque en varias ocasiones pueden contar con apoyo o guía del área de pruebas (GUTIERREZ, 2015).

En el caso de las pruebas internas propuesta por la metodología el proyecto decidió dentro de estas pruebas realizar las pruebas funcionales al módulo Arrendamiento de contenedores; a partir de este momento las pruebas internas serán tratadas como funcionales.

3.5.1 Métodos de prueba

Las técnicas de evaluación de *software* dinámicas o prueba proporcionan la creación de casos de prueba que provoquen fallos en el programa. Algunas de estas técnicas son (Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla, 2006):

- **Pruebas de caja blanca o estructural:** se centran en un análisis los detalles de código a evaluar, por lo que se hace necesario conocer la lógica del programa. El uso de estas posibilita la obtención de casos de prueba para que se ejecuten, al menos una vez, todas las sentencias del programa, y todas las condiciones tanto en su vertiente verdadera como falsa (Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla, 2006).
- **Pruebas de caja negra o funcional:** realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, solo la funcionalidad que debe realizar (Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla, 2006).

Para realizar las pruebas de caja negra y confeccionar los casos de prueba se exigen algunos criterios. Algunos de ellos son: (Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla, 2006):

- **Partición de equivalencia:** este divide el campo de entrada de un programa en un número finito de variables de equivalencia. Se definen dos tipos de variables de equivalencia, las válidas, que evidencian entradas válidas al sistema y las no válidas, que representan valores erróneos de entrada, aunque pueden existir valores no relevantes a los que no es necesario otorgarle un valor.
- **Análisis de valores límites:** maneja la habilidad del programa de manejar valores de datos que se encuentran en los límites aceptables del programa.

Con el objetivo de realizar las pruebas de caja negra o funcional, es necesario basarse en el artefacto Casos de Prueba propuesto por la metodología de desarrollo seleccionada. Un caso de prueba proporciona una forma de comprobar el correcto funcionamiento del sistema, estos incluyen las entradas, resultados y condiciones que se van a verificar, proporcionando una guía para el probador (RUMBAUG, y otros, 2007).

3.5.2 Diseño de caso de prueba

En los casos de prueba (CP) se incluyen la descripción de los principales escenarios, actores, posibles entradas, las variables que intervienen en el proceso y el flujo central. A continuación, se muestra el CP para el requisito Incluir contenedor arrendado. Para ver los restantes CP consultar el [Anexo #7](#).

Tabla 6: CP Incluir contenedor arrendado

| CP Incluir contenedor arrendado | | | |
|--|---|--|--|
| <p>Descripción general: Permite incluir nuevos contenedores arrendados al sistema. Condiciones de ejecución: El usuario debe estar autenticado en el sistema. El usuario debe poseer los permisos para incluir un contenedor arrendado.</p> <p style="text-align: center;">SC1 Incluir contenedor arrendado</p> | | | |
| Escenario | Descripción | Respuesta del sistema | Flujo central |
| EC 1.1 Incluir contenedor arrendado | Se selecciona la opción incluir un nuevo contenedor arrendado | El sistema debe permitir introducir los siguientes datos para incluir un contenedor arrendado, de manera obligatoria: <ul style="list-style-type: none"> - No. contenedor - Longitud contenedor - Tipo contenedor - Proveedor - Estado - Ubicación - Fecha alquiler - Fecha devolución - Cliente - Observaciones | Módulo Arrendamiento Contenedores / Contenedores/Incluir |

| | | | |
|---|---|---|---|
| | | Y brinda las siguientes opciones: - Guardar o Descartar | |
| EC 1.2 Guardar los datos | Se Introducen y/o selecciona los datos del contenedor arrendado y selecciona la opción guardar. | El sistema valida y guarda los datos de nuevo contenedor arrendado | Módulo Arrendamiento Contenedores / Contenedores/Incluir/ Guardar |
| EC 1.3 Información incompleta | Existen campos obligatorios que no se completaron | El sistema señala el o los campos obligatorios que no hayan sido introducidos y/o seleccionados y muestra el mensaje de información: "Los siguientes campos son obligatorios"; indicando los campos en cuestión. | Módulo Arrendamiento Contenedores / Contenedores/Incluir/ Guardar |
| EC 1.4 Información incorrecta | Existen campos incorrectos | El sistema señala el o los campos incorrectos y muestra el mensaje de información: "Los siguientes campos son incorrectos"; indicando los campos en cuestión. | Módulo Arrendamiento Contenedores / Contenedores/Incluir/ Guardar |
| EC 1.5 Descartar cuando se hayan introducido los datos | Se selecciona el botón descartar cuando ya se han introducido datos | El sistema muestra un mensaje de información: Se ha modificado la entidad, sus cambios se perderán y solicita confirmación ¿Está seguro que quiere abandonar la página? El sistema brinda las siguientes opciones: - Aceptar - Cancelar | Módulo Arrendamiento Contenedores / Contenedores/Incluir/Descartar |
| EC 1.6 Descartar cuando se hayan introducido los datos opción Aceptar | Se selecciona la opción Aceptar | El sistema elimina los datos introducidos y regresa al listado de contenedores arrendados | Módulo Arrendamiento Contenedores / Contenedores/Incluir/Descartar/Aceptar |
| EC 1.7 Descartar cuando se hayan introducido los datos opción Cancelar | Se selecciona la opción Cancelar | El sistema mantiene los cambios realizados | Módulo Arrendamiento Contenedores / Contenedores/Incluir/Descartar/Cancelar |
| EC 1.8 Descartar | Se selecciona la opción Descartar | El sistema regresa al listado de contenedores arrendados. | Módulo Arrendamiento Contenedores / Contenedores/Incluir/Descartar |

3.6 Resultados de las pruebas

3.6.1 Resultados de las pruebas

Con el objetivo de verificar el cumplimiento de los requisitos se utilizan las pruebas de caja negra, teniendo en cuenta la técnica partición por equivalencia. Se hace uso de los casos de prueba con el fin de detectar la mayor cantidad de no conformidades posibles realizándose cuatro iteraciones. Las no conformidades (NC) detectadas se clasificaron en dos grupos, las significativas que incluyen las de funcionalidad y validación, mientras que las no significativas agrupan las de interfaces de usuario (errores ortográficos).

En la primera iteración se encontraron catorce no conformidades significativas (de color verde) y ocho no conformidades no significativas (de color azul). En la segunda iteración once significativas y seis no significativas, para la tercera iteración cinco y cuatro respectivamente. En la última iteración fueron encontradas solo una no conformidad significativa. El resultado de las pruebas funcionales por iteración se muestra en la tabla a continuación.

Tabla 7: Resultado de las pruebas por iteraciones

| | Iteración 1 | Iteración 2 | Iteración 3 | Iteración 4 | TOTAL |
|-----------------------------|-------------|-------------|-------------|-------------|-------|
| Interfaz de usuarios | 8 | 6 | 4 | 0 | 18 |
| Funcionalidad | 8 | 4 | 2 | 0 | 14 |
| Validación | 6 | 5 | 3 | 1 | 15 |
| TOTAL | 14/8 | 11/6 | 5/4 | 1/0 | 47 |

El gráfico que se muestra a continuación muestra la relación entre las NC significativas y no significativas.

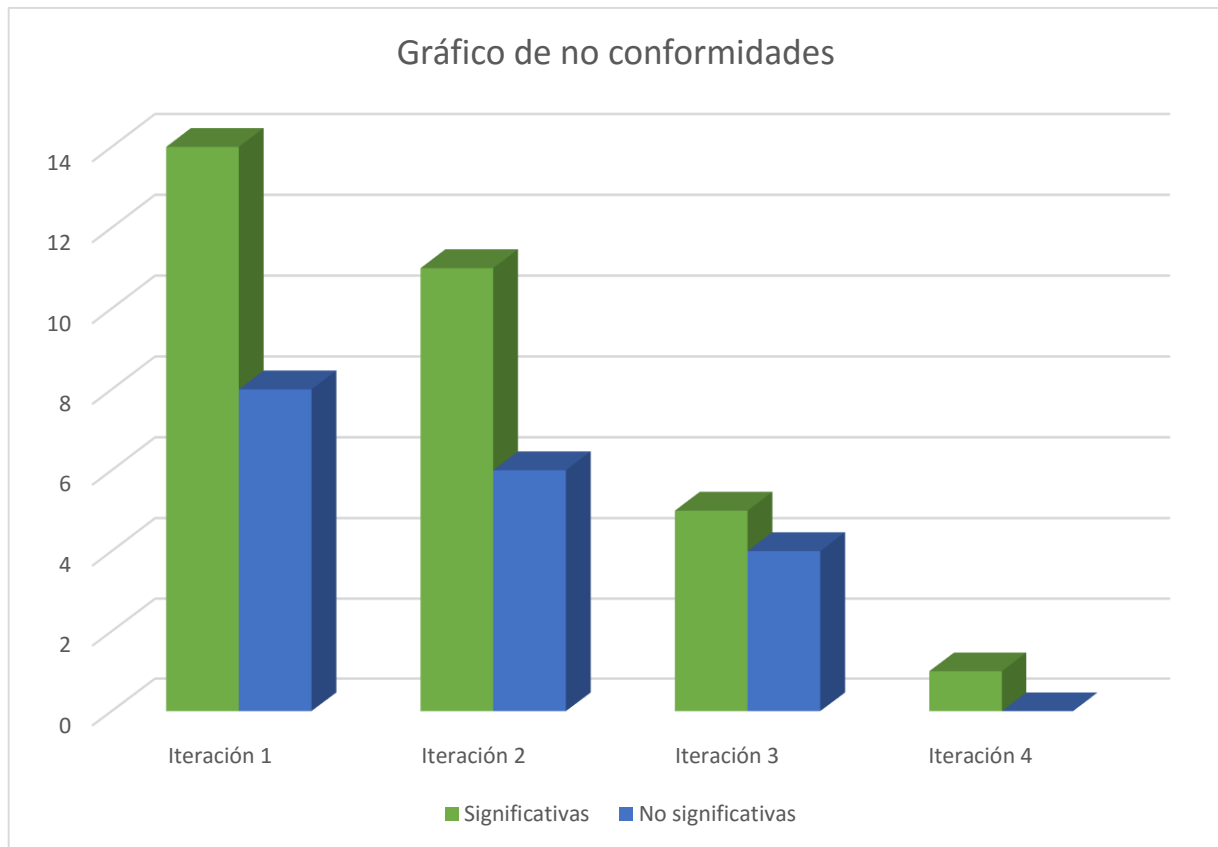


Figura 19: Resultados de las pruebas al módulo

3.7 Conclusiones parciales

En este capítulo se presentaron los pasos realizados para la implementación mostrando el diagrama de componente perteneciendo al módulo Arrendamiento de contenedores, que permite tener una vista general de todas las dependencias y funcionalidades necesarias para el correcto funcionamiento del módulo. Se definió el estándar de codificación a utilizar que incluye la estructura de los módulos desarrollados en Odo, así como, las pautas para los archivos XML y Python propiciando un mejor entendimiento del código. Las pruebas realizadas a la solución desarrollada, permitió corregir los errores encontrados al aplicar las mismas se obtuvo un módulo que responde en su totalidad a los requisitos obtenidos.

Conclusiones generales

Con el desarrollo del presente trabajo investigativo, se arribó a las siguientes conclusiones:

- El análisis realizado como parte de la investigación sirvió de apoyo en la toma de decisiones con vista al desarrollo del módulo Arrendamiento de contenedores, demostrando que las soluciones similares no cumplen en su totalidad con los requisitos de la solución deseada, por lo que quedaron descartadas.
- El levantamiento de información sirvió de guía para el posterior análisis y elaboración de la propuesta de solución. Los artefactos generados por la metodología AUP en su variación UCI facilitaron la implementación del módulo.
- La implementación del módulo Arrendamiento de contenedores permitió darle solución a la situación planteada garantizando la informatización del arrendamiento de contenedores en la empresa.
- Las pruebas realizadas al módulo, permitieron garantizar la calidad del software y comprobar que los requisitos definidos en el sistema son los que el cliente desea.

Recomendaciones

Para perfeccionar y ampliar las funcionalidades de la solución propuesta, se proponen las siguientes recomendaciones:

- Agregar al módulo una funcionalidad para la generación de reportes de órdenes de liberación.
- Agregar al módulo una funcionalidad para mostrar los reportes de contenedores por estado en un gráfico de pastel.
- Adicionar al módulo una funcionalidad para realizar reportes por clasificación de daños en los reportes de contenedores averiados.

Glosario de términos

CASE: Acrónimo de Computer Aided Software Engineering (en español: Ingeniería de Software Asistida por Computadora), estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo de software en tareas como realizar un diseño del proyecto, implementación de código automáticamente con el diseño dado, documentación o detención de errores, entre otras.

Framework: Estructura conceptual y tecnológica de soporte definida, normalmente, con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Multiprocesos: Modalidad operativa de un sistema informático que permite la ejecución simultánea de varios programas compartiendo la memoria central y las unidades periféricas.

Multihilos: Contiene dos o más partes que se pueden ejecutar de manera concurrente o simultánea. A cada una de esas partes se les denomina hilo y cada uno de ellos define un camino de ejecución independiente.

HTTP: Acrónimo de HyperText Transfer Protocol (en español: Protocolo de Transferencia de Hipertexto) es el método más común de intercambio de información en la World Wide Web, el método mediante el cual se transfieren las páginas web a un ordenador.

W3C: World Wide Web Consortium.

ORM: Acrónimo de Object-Relational Mapping (en español: Mapeo Objeto-Relacional), es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

Widget: Es una aplicación o programa pequeño, que proveen acceso a funciones frecuentemente usadas y de información visual.

Plugin: Son programas que se agregan a otros ya existentes para ofrecer una nueva funcionalidad. Estos programas no funcionan de forma independiente.

PDF: Acrónimo de Portable Document Format (en español: Formato de Documento Portátil) es un formato de almacenamiento de documentos digitales independiente de plataformas de software o

hardware. Este formato es de tipo compuesto (imagen vectorial, mapa de bits y texto).

ACID: Es un grupo de cuatro propiedades que garantizan que las transacciones en las bases de datos se realicen de forma confiable. Estas propiedades son atomicidad, consistencia, aislamiento (*isolation* en inglés) y durabilidad.

Referencias bibliográficas

Academia, Real. DLE: transitario, ria - Diccionario de la lengua española - Edición del Tricentenario. [En línea] [Citado el: 20 de Noviembre de 2015.] <http://dle.rae.es/?id=aKdgVUI>.

ADINS. 2011. OpenObject. [En línea] 2011. [Citado el: 23 de Febrero de 2016.] <http://adins.be/en/openobject>.

—. 2011. Technical foundation. [En línea] 2011. [Citado el: 24 de Febrero de 2016.] <http://adins.be/en/openobject/36-fondation-technique>.

ALEGSA, Leandro. 2014. Definición de Python (lenguaje de programación). [En línea] 2014. [Citado el: 24 de Noviembre de 2015.] <http://www.alegsa.com.ar/Dic/python.php>.

Ambyssoft Inc. 2005. El Proceso Unificado Ágil v1.1. [En línea] 2005. [Citado el: 18 de Diciembre de 2015.] <http://www.cc.una.ac.cr/AUP/>.

Bargaw. 2014. Odoo - Bargaw. [En línea] 2014. [Citado el: 4 de Marzo de 2016.] <http://www.bargaw.com/erpcrmodoo/>.

BEDOYA, Andrés. 2011. Patrones GOF. [En línea] 2011. [Citado el: 15 de Marzo de 2016.] <http://geektheplanet.net/5462/patrones-gof.xhtml>.

BRESSANO, Mario Osvaldo. 2010. WebBress. Education & Web Design. [En línea] 2010. [Citado el: 29 de Abril de 2016.] http://www.clubdelsuran.com.ar/site/materiales/proyecto/diagramas_del_uml.pdf.

CALVO LOPEZ, Eva. 2010. [En línea] 2010. [Citado el: 31 de Enero de 2016.] <http://upcommons.upc.edu/bitstream/handle/2099.1/8947/Memoria.pdf?sequence=1&isAllowed=y>.

CASTRO, Eliseo. 2008. UML - Lenguaje de Modelamiento Unificado. [En línea] 2008. [Citado el: 26 de Noviembre de 2015.] <http://es.slideshare.net/ecastrojimenez/uml-lenguaje-de-modelamiento-unificado-presentation>.

ClubEnsayos. 2011. Concepto E Importancia De La Gestion Empresarial - Composiciones de Colegio. [En línea] 2011. [Citado el: 10 de Diciembre de 2015.] <https://www.clubensayos.com/Negocios/Concepto-E-Importancia-De-La-Gestion-Empresarial/69236.html>.

CUBIZA. 2011. CUBIZA. Empresa central de equipos. [En línea] 2011. [Citado el: 11 de Enero de 2016.]

2016.] <http://www.cubiza.cu/empresa.html>.

Definición. 2010. Definición de contenedor - Qué es, Significado y Concepto. [En línea] 2010. [Citado el: 11 de Noviembre de 2015.] <http://definicion.de/contenedor/>.

DefinicionABC. 2010. Definición de Arrendamiento » Concepto en Definición ABC. . [En línea] 2010. [Citado el: 13 de Noviembre de 2015.] <http://www.definicionabc.com/derecho/arrendamiento.php>.

Departamento de Ciencias de la Computación e Inteligencia Artificial. 2008. Especificación de requerimientos. Diseño de base de datos. [En línea] Universidad de Granada, 2008. [Citado el: 10 de Diciembre de 2015.] <http://elvex.ugr.es/idbis/db/docs/design/2-requirements.pdf>.

Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla. 2006. Características y Fases de la Prueba. [En línea] 2006. [Citado el: 20 de Abril de 2016.] <http://www.lsi.us.es/docencia/get.php?id=361>.

Domatrix. 2012. Odoos – OpenERP – ERP, CRM, MRP, SGA 100% Libre – | Sin Licencias ¿Por qué elegir OpenErp? [En línea] 2012. [Citado el: 25 de Enero de 2016.] <http://openerpspain.com/openerp/por-que-elegir-openerp/>.

—. **2012.** Odoos – OpenERP – ERP, CRM, MRP, SGA 100% Libre – | Sin Licencias ¿Que es OpenERP? ERP 100% Libre. [En línea] 2012. [Citado el: 20 de Enero de 2016.] <http://openerpspain.com/openerp/que-es-openerp/>.

EXXIS. 2010. ¿Qué es un ERP? [En línea] 2010. [Citado el: 29 de Enero de 2016.] http://www.exxis.cl/index.php?option=com_content&view=article&id=3:ique-es-un-erp&catid=1:preguntas-frecuentes&Itemid=3.

Grupo Azucarero AZCUBA. 2014. ESAZUCAR | AZCUBA. [En línea] 2014. [Citado el: 10 de Enero de 2016.] <http://www.azcuba.cu/?p=918>.

GUTIERREZ, Ismael. 2015. Pruebas de Software: Niveles de Prueba. [En línea] 2015. [Citado el: 15 de Mayo de 2016.] <http://pruebasdesw.blogspot.com/2015/08/niveles-de-prueba.html>.

IBERTRANSIT. 2013. Transitario | IBERTRANSIT. [En línea] 2013. [Citado el: 10 de Noviembre de 2015.] <http://www.ibertransit.com/inicio-transitario>.

Implementacion SIG. 2015. Que es un Sistema de Gestión. [En línea] 2015. [Citado el: 12 de Diciembre de 2015.] <http://www.implementacionsig.com/index.php/23-noticiac/28-que-es-un->

sistema-de-gestion.

Innova. 2013. ADUANAS - Contenedores. Definición. . [En línea] 2013. [Citado el: 13 de Noviembre de 2015.] <http://www.aduanas.gub.uy/innovaportal/v/2542/8/innova.front/contenedores-definicion.html>.

JetBrains. 2010. PyCharm. [En línea] 2010. [Citado el: 6 de Febrero de 2016.] <https://www.jetbrains.com/pycharm/>.

LARMAN, Craig. 1999. *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. España : México : Prentice-Hall, cop. 1999, 1999. ISBN: 970-17-0261-1.

LARMAN, Craig y HALL, Prentice. 2003. El Modelo de Diseño. [En línea] 2003. [Citado el: 28 de Abril de 2016.] is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDiseno.pdf.

—. **2003.** Modelo de Dominio. [En línea] 2003. [Citado el: 1 de Mayo de 2016.] <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf>.

MARTINEZ, Rafael. 2010. Sobre PostgreSQL | www.postgresql.org.es. [En línea] 2010. [Citado el: 8 de Diciembre de 2015.] http://www.postgresql.org.es/sobre_postgresql.

MENDOZA, Miguel Ángel. 2015. ¿Ciberseguridad o seguridad de la información? Aclarando la diferencia. [En línea] 2015. [Citado el: 20 de Febrero de 2016.] <http://www.welivesecurity.com/la-es/2015/06/16/ciberseguridad-seguridad-informacion-diferencia/>.

NUBISTALIA. 2011. Odoo on demadn - SaaS - OpenERP - ¿Para qué sirve un ERP? [En línea] 2011. [Citado el: 1 de Febrero de 2016.] <http://www.openerpondemand.com/qu%C3%A9-es-odoo/para-qu%C3%A9-sirve-un-erp/>.

—. **2011.** OpenERP OnDemand. ¡El ERP lider de Europa desde 99€/mes! - Odoo - Módulos. [En línea] 2011. [Citado el: 6 de Febrero de 2016.] <http://www.nubistalia.es/soluciones-y-servicios/odoo-openerp/odoo-m%C3%B3dulos/>.

O'Reilly Media, Inc. 2010. XML.com. [En línea] 2010. [Citado el: 12 de Febrero de 2016.] <http://www.xml.com/pub/a/98/10/guide0.html?page=2>.

Odoo. Odoo Guidelines — odoo 8.0 documentation. [En línea] [Citado el: 20 de Abril de 2016.] <https://www.odoo.com/documentation/8.0/reference/guidelines.html>.

ODOO. Open Source ERP and CRM | Odoo. [En línea] [Citado el: 8 de Febrero de 2016.]

<https://www.odoo.com/>.

Odoo S.A. 2012. Product Container | Odoo Apps. [En línea] 2012. [Citado el: 2 de Marzo de 2016.] https://www.odoo.com/apps/modules/7.0/product_container/.

—. **2012.** Product Container Tracking | Odoo Apps. [En línea] 2012. [Citado el: 3 de Marzo de 2016.] https://www.odoo.com/apps/modules/7.0/product_container_tracking/.

Oposiciones TIC. 2011. Arquitectura cliente servidor - Oposiciones TIC. [En línea] 2011. [Citado el: 8 de Marzo de 2016.] <http://oposicionestic.blogspot.com/2011/06/arquitectura-cliente-servidor.html>.

ORÉ, Alexander. 2009. Pruebas Funcionales - Software Testing and QA. [En línea] 2009. [Citado el: 20 de Abril de 2016.] http://www.calidadyssoftware.com/testing/pruebas_funcionales.php.

Patrón Modelo-Vista-Controlador. **Fernández Romero, Yenisleidy y Díaz González, Yanette. 2012.** 1, La Habana : Revista Telem@tica, 2012, Revista Digital de las Tecnologías de la Información y las Comunicaciones, Vol. XI, págs. 47-57. ISSN 1729-3804.

PRESSMAN, Roger S. 2005. *Ingeniería de software, un enfoque practico.* s.l. : McGraw-Hill Companies, 2005. ISBN 8448132149.

PRESSMAN, Roger S. 2006. *Ingeniería del Software. Un enfoque Práctico .* Madrid : McGraw-Hill, 2006. ISBN: 970-10-5473-3.

Prodware. 2016. Definición de ERP. [En línea] 2016. [Citado el: 30 de Enero de 2016.] <https://www.elegirerp.com/definicion-erp>.

RedAccenir, S.L. 2009. IMB Container - Descargar Gratis. [En línea] 2009. [Citado el: 30 de Enero de 2016.] <http://www.portalprogramas.com/imb-container/>.

RUMBAUG, James, JACOBSON, Ivar y BOOCH, Grady. 2007. El Lenguaje Unificado de Modelado. Manual de referencia. [En línea] 2007. [Citado el: 22 de Abril de 2016.] <http://librosgratisparaeluniversitario.blogspot.com/2014/03/el-lenguaje-unificado-de-modelado.html>. ISBN 978-84-7829-087-1.

SANCHEZ RODRIGUEZ, Tamara. 2015. *Metodología de desarrollo para la UCI.* La Habana : s.n., 2015.

Significados. 2013. Significado de Metodología - Qué es, Concepto y Definición. [En línea] 2013. [Citado el: 27 de Noviembre de 2015.] <http://www.significados.com/metodologia/>.

TRINCHET SOLER, Rafael y PEDRIANES VIGO , Margarita. 2004. *Origen, estado actual y perspectivas de la Red Nacional de Cirugía Pediátrica.* La Habana : s.n., 2004. ISSN 1024-9435, ISSN-e 1530-2880.

UDIMA. 2009. XML ¿Qué es? | Manual de XML. [En línea] 2009. [Citado el: 15 de Febrero de 2016.] <http://www.mundolinux.info/que-es-xml.htm>.

Visual Paradigm International. 2014. Visual Paradigm Searchfor Visual Paradigm. [En línea] 2014. [Citado el: 5 de Diciembre de 2015.] <http://www.visual-paradigm.com/search/search.jsp?query=Visual+Paradigm&x=22&y=9>.

WESKE, Mathias. 2007. *Business Process Management. Concepts, Languajes, Architectures.* New York : Springer Berlin Heidelberg, 2007. pág. 5. ISBN 978-3-540-73521-2.

Bibliografía

BAHIT, Eugenia. 2012. *Teoría sintáctico-gramatical de Objetos.* s.l. : Bubok Publishing S.L, 2012. ISBN 1210292587717.

HERNÁNDEZ LEÓN, Alfredo Roylo y COELLO GONZÁLEZ, Sayda. 2011. *El procesos de investigación científica.* Ciudad de la Habana : Universitaria del Ministerio de Educación Superior, 2011. ISBN 978-959-16-1307-3.

SUMMERFIELD, Mark. 2014. *Python in practice.* New Jersey : Addison-Wesley, 2014. ISBN 978-0-321-90563-5.

VERMA, Rahul y GIRIDHAR, Chetan. 2011. *Design Patterns in Python.* s.l. : Testeing Perspective, 2011.

GRAU, Ricardo, CORREA, Cecilia y ROJAS, Mauricio. 2004. *Metodología de la investigación.* Segunda. Ibague : El Poirá Editores S.A, 2004. ISBN 958-8028-10-8.