



**Universitat Autònoma
de Barcelona**

Gestor de importaciones, transportes y previsiones

Memoria del proyecto

De Ingeniería Técnica en
Informática de Gestión
realizado por

Víctor Rodríguez Álvarez

y dirigido por

Just Piferrer Rodriguez

Escola d'Enginyeria

Sabadell, Septiembre de 2012

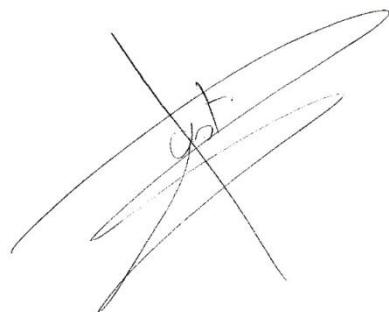
El sotasignant, ***Just Piferrer Rodriguez***,
professor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present
memòria ha estat realitzat sota la seva direcció
per ***Víctor Rodríguez Álvarez***.

I per a que consti firma la present.

Sabadell, ***setembre*** de ***2012***

A handwritten signature in black ink, appearing to read "Just Piferrer Rodriguez". The signature is fluid and cursive, with a large, stylized "J" at the beginning.

Signat: ***Just Piferrer Rodriguez***

Índice de contenido

Capítulo I. Introducción	8
1.1 Objetivos del proyecto	9
1.2 Descripción del proyecto	9
1.3 Motivaciones personales.....	10
1.4 Estructura de la memoria	10
1.5 Metodología de desarrollo	10
Capítulo II. Estudio de viabilidad	12
2.1 Introducción	13
2.2 Descripción del contexto	13
2.3 Funcionamiento del sistema	13
2.4 Perfil de usuario	14
2.5 Objetivos	14
2.6 Descripción general del sistema	14
2.7 Recursos necesarios	14
2.8 Presupuesto	15
2.9 Evaluación de riesgos	15
2.10 Planificación del proyecto	16
2.10.1 Diagrama de Gantt	16
2.11 Conclusiones	16
Capítulo III. Caracterización del sistema.....	17
3.1 Introducción	17
3.2 Componentes	18
3.2.1 Base de datos	18
3.2.2 Interfaz web	21
Capítulo IV. Programación de la interfaz web	25
4.1 Introducción	26
4.2 Modelos	26
4.2.1 AppModel.php.....	26
4.2.2 User.php.....	26
4.2.3 Container.php.....	27
4.2.4 ContainerLine.php	27
4.2.5 Order.php	27

4.2.6 OrderLine.php	27
4.2.7 Item.php.....	27
4.2.8 Origin.php.....	27
4.2.9 Destination.php.....	27
4.2.10 Transport.php.....	27
4.3 Vistas	28
4.3.1 Users.....	28
4.3.2 Containers	29
4.3.3 ContainerLines.....	36
4.3.4 Orders.....	37
4.3.5 OrderLines	39
4.3.6 Items.....	41
4.3.7 Origins	44
4.3.8 Destinations.....	45
4.3.9 Transports	46
4.4 Controladores.....	48
4.4.1 AppController.php.....	48
4.4.2 UsersController.php	49
4.4.3 ContainersController.php	49
4.4.4 ContainerLinesController.php	53
4.4.5 OrdersController.php	53
4.4.6 OrderLinesController.php.....	54
4.4.7 ItemsController.php	55
4.4.8 OriginsController.php	55
4.4.9 DestinationsController.php	56
4.4.10 TransportsController.php	57
Capítulo V. Pruebas y resultados	58
5.1 Introducción	59
5.2 Pruebas de unidad.....	59
5.2.1 Pruebas de caja blanca	59
5.2.2 Pruebas de caja negra	59
5.3 Pruebas de integración.....	60
5.3.1 Pruebas de caja blanca	60
5.3.2 Pruebas de caja negra	60

5.4 Pruebas de sistema	61
5.4.1 Pruebas de caja blanca	61
5.4.2 Pruebas de caja negra	61
5.5 Comentarios	62
Capítulo VI. Conclusiones y posibles ampliaciones.....	63
6.1 Conclusiones	64
6.2 Posibles ampliaciones.....	64
6.2.1 Gestión de usuarios por centro y rol	64
6.2.2 Ampliación de la información.....	64
6.2.3 Mejorar rendimiento de la aplicación web.....	65
6.2.4 Selección de múltiples líneas	65
6.2.5 Introducción de transporte aéreo	65
Capítulo VII. Bibliografía y referencias	66
7.1 Bibliografía	67
7.2 Referencias.....	67
7.3 Otros recursos	67

Índice de tablas y figuras

Tabla 1.1. Evolución básica de una importación	09
Figura 1.2. Metodología de desarrollo	11
Tabla 2.1. Presupuesto con precios de mercado	15
Tabla 2.2. Presupuesto proyecto final de carrera.....	15
Figura 2.3. Diagrama de Gantt	16
Figura 3.1. Esquema de la base de datos	18
Figura 3.2. Tabla “users”	19
Figura 3.3. Tabla “items”	19
Figura 3.4. Tabla “orders”	19
Figura 3.5. Tabla “order_lines”	20
Figura 3.6. Tabla “container”	20
Figura 3.7. Tabla “container_lines”	20
Figura 3.8. Tabla “origins”	21
Figura 3.9. Tabla “destinations”	21
Figura 3.10. Tabla “transports”	21
Tabla 3.11. Convenciones básicas de CakePHP	22
Figura 3.12. Arquitectura de software MVC.....	22
Figura 3.13. Funcionamiento MVC	23
Tabla 4.1. Relaciones entre modelos CakePHP	26
Figura 4.2. Modelo User/Validate.php	26
Figura 4.3. Vista Users/login.ctp.....	28
Figura 4.4. Código de la vista Users/login.ctp	28
Figura 4.5. Vista Users/add.ctp	29
Figura 4.6. Vista Users/index.ctp.....	29
Figura 4.7. Vista Containers/index.ctp (Rol “Export”)	30
Figura 4.8. Vista Containers/index.ctp (Rol “Import”)	30
Figura 4.9. Vista Containers/index.ctp (Rol “Admin”)	30
Figura 4.10. Iconos menú superior en vista Containers/index.ctp	31

Figura 4.11. Iconos menú inferior en vista Containers/index.ctp	31
Figura 4.12. Vista Containers/add.ctp	31
Figura 4.13. Vista Containers/edit.ctp	31
Figura 4.14. Vista Containers/filter.ctp	32
Figura 4.15. Vista Containers/filteretd.ctp	32
Figura 4.16. Vista Containers/filtereta.ctp	32
Figura 4.17. Vista Containers/forecastrange.ctp	32
Figura 4.18. Vista Containers/forecast.ctp	33
Figura 4.19. Vista Containers/updateata.ctp	33
Figura 4.20. Vista “Lista de carga”	34
Figura 4.21. Vista “Factura”	34
Figura 4.22. Vista Containers/desviationsrange.ctp	35
Figura 4.23. Vista Containers/desviations.ctp	35
Figura 4.24. Vista Containers/cddontimerange.ctp	35
Figura 4.25. Vista Containers/cddontime.ctp	36
Figura 4.26. Mensaje de aviso de contenedor sin órdenes asociadas	36
Figura 4.27. Vista ContainerLines/view.ctp (Rol “Admin”)	36
Figura 4.28. Vista ContainerLines/view.ctp (Rol “Export”)	37
Figura 4.29. Vista ContainerLines/view.ctp (Rol “Import”)	37
Figura 4.30. Vista ContainerLines/add.ctp	37
Figura 4.31. Vista Orders/index.ctp (Rol “Admin”)	38
Figura 4.32. Vista Orders/index.ctp (Rol “Export”)	38
Figura 4.33. Vista Orders/index.ctp (Rol “Import”)	38
Figura 4.34. Vista Orders/add.ctp	39
Figura 4.35. Vista Orders/edit.ctp	39
Figura 4.36. Vista OrderLines/view.ctp (Rol “Admin”)	40
Figura 4.37. Vista OrderLines/view.ctp (Rol “Export”)	40
Figura 4.38. Vista OrderLines/view.ctp (Rol “Import”)	40
Figura 4.39. Vista OrderLines/add.ctp	41
Figura 4.40. Vista OrderLines/edit.ctp	41

Figura 4.41. Vista Items/index.ctp (Rol “Admin”)	41
Figura 4.42. Vista Items/index.ctp (Rol “Export”)	42
Figura 4.43. Vista Items/index.ctp (Rol “Import”)	42
Figura 4.44. Vista Items/add.ctp	42
Figura 4.45. Vista Items/edit.ctp	43
Figura 4.46. Vista “Listado de artículos” en PDF	43
Figura 4.47. Vista Origins/index.ctp (Rol “Admin”)	44
Figura 4.48. Vista Origins/index.ctp (Rol “Export”)	44
Figura 4.49. Vista Origins/add.ctp	44
Figura 4.50. Vista Origins/edit.ctp	45
Figura 4.51. Vista Destinations/index.ctp (Rol “Admin”)	45
Figura 4.52. Vista Destinations/index.ctp (Rol “Import”)	45
Figura 4.53. Vista Destinations/add.ctp	46
Figura 4.54. Vista Destinations/edit.ctp	46
Figura 4.55. Vista Transports/index.ctp (Rol “Admin”)	46
Figura 4.56. Vista Transports/index.ctp (Rol “Export”)	47
Figura 4.57. Vista Transports/add.ctp	47
Figura 4.58. Vista Transports/edit.ctp	47
Figura 4.59. Vista Transports/budget.ctp (Rol “Admin”)	48
Figura 4.60. Vista Transports/Budget.ctp (Rol “Export”)	48
Figura 4.61. AppController – Función isAuthorized	49
Figura 4.62. Gestión del menú inferior (Controlador)	50
Figura 4.63. ContainersController – Función beforeFilter	52

Capítulo I.

Introducción

1.1 Objetivos del proyecto

El objetivo del proyecto es crear un sistema que permita gestionar las importaciones realizadas por una empresa, extraer previsiones para la planificación de la plantilla en el almacén de entrega y analizar la calidad del transporte.

Más concretamente, como función principal el aplicativo diseñado debe permitir el almacenamiento en base de datos de los artículos, pedidos (conjunto de artículos) y envíos (conjunto de pedidos) que la empresa importa así como proveer las herramientas necesarias para su creación, edición, actualización y seguimiento en tiempo real.

La obtención de previsiones para los almacenes de entrega y el análisis del transporte son funciones secundarias pero no por ello menos importantes ya que permitirán a las empresas usuarias planificar la plantilla necesaria para la recepción de la mercancía y proporcionarán información muy valiosa a la hora de negociar con las empresas de transporte.

1.2 Descripción del proyecto

El proyecto se compone de una base de datos y una aplicación web desarrollada en lenguaje PHP por su gran adaptabilidad a cualquier sistema operativo, gran posibilidad de desarrollo y facilidad de mantenimiento.

Para garantizar la correcta gestión de los datos, el sistema cuenta con tres tipos de rol de usuario (“Export”, “Import”, “Admin”) que permitirán realizar unas funciones u otras.

Los usuarios con rol “Export” se encargan de la creación del envío y de la exportación desde el país de origen mientras que los usuarios “Import” gestionan la recepción de la mercancía en destino y analizan la calidad del transporte. El rol “Admin” da acceso completo a la aplicación.

A continuación se detalla la línea de evolución básica de una importación en este sistema así como el autor para cada una de las acciones llevadas a cabo:

Actor	Acción
EXPORT	Creación de artículos
	Creación de las órdenes
	Añadir artículos a las órdenes
	Creación del envío
	Añadir órdenes al envío
	Seleccionar transporte
	Realizar envío
IMPORT	Actualizar con la fecha real de llegada
	Archivar envío una vez entregado
	Analís de calidad del transporte y la entrega

Tabla 1.1. Evolución básica de una importación

1.3 Motivaciones personales

La motivación principal de la elección del tema del proyecto es mi gran vocación por la logística y el transporte fruto de mi experiencia laboral en almacén y en flujos internacionales.

Desde el punto de vista informático quería profundizar en la creación de páginas web, gestión de accesos, bases de datos y usuarios y aprender a utilizar “frameworks¹”; motivo por el que, para este proyecto, se ha utilizado el “framework” CakePHP comentado más adelante en esta memoria.

1.4 Estructura de la memoria

Esta memoria se estructura en siete capítulos en los que se detallan los puntos más importantes del aplicativo desarrollado y de la planificación del proyecto.

En el primer capítulo se especifican los objetivos principales y se da una breve descripción del proyecto y de su futura funcionalidad.

El segundo capítulo se centra en el estudio de viabilidad. Partiendo de la explicación del contexto sobre el que se desarrolla el sistema se detalla: el funcionamiento del mismo, los usuarios a los que va destinado, objetivos, recursos necesarios, presupuesto y evaluación de riesgos. Una vez analizados los puntos anteriores se presentan la planificación del proyecto mediante un diagrama de Gantt y las conclusiones.

En el tercer capítulo se encuentra la explicación detallada de cada uno de los componentes del sistema, su estructura y la relación entre ellos. También se incluye información sobre el “framework” y “plugins²” utilizados en el proyecto.

En el capítulo número cuatro se describen las funcionalidades del proyecto así como las vistas y modelos relacionados a las tablas de la base de datos.

Posteriormente, en el capítulo cinco se explica la metodología de pruebas utilizada y se presentan algunos ejemplos de las pruebas realizadas y los resultados obtenidos tanto a nivel modular como de relación entre los módulos del sistema.

Finalmente, se encuentran los capítulos seis y siete, donde se comentan las conclusiones, posibles ampliaciones y bibliografía.

1.5 Metodología de desarrollo

Dada la naturaleza modular del proyecto se ha seguido la metodología de desarrollo evolutiva. Esto ha permitido simplificar las pruebas modulares y la corrección de errores.

Inicialmente se ha realizado el análisis de viabilidad y, una vez finalizado, se ha dividido el desarrollo del proyecto en diversos módulos siguiendo la evolución básica de las importaciones vista en el punto 1.2.

1. Framework: Marco de desarrollo rápido que incluye una estructura que sirve de base para crear aplicaciones web y provee herramientas que simplifican la programación de las partes comunes a todos los proyectos.

2. Plugin (Complemento): Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

Para cada uno de estos módulos se ha seguido el ciclo de diseño, codificación y pruebas de forma repetitiva antes de pasar al módulo siguiente, que tiene en cuenta los anteriores.

Posteriormente se realiza la integración de todos los módulos, las pruebas de relación entre módulos y para finalizar se realiza la documentación.

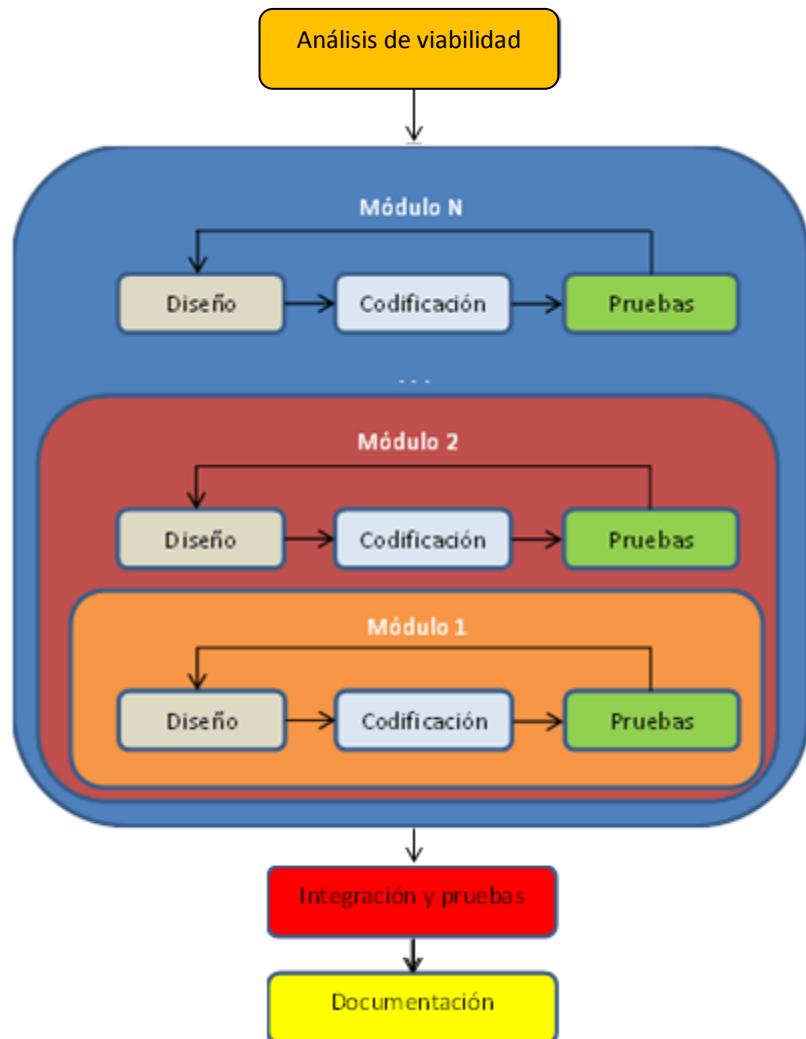


Figura 1.2 Metodología de desarrollo

Capítulo II.

Estudio de viabilidad

2.1 Introducción

En este capítulo se analiza la viabilidad del proyecto hasta llegar a la conclusión de si es posible y rentable su desarrollo o no. Es una de las partes fundamentales de todo proyecto ya que permite anticipar de manera muy fiable los costes, recursos y riesgos, por lo que de él depende en gran medida el éxito del proyecto.

Partiendo del contexto de desarrollo se describe el funcionamiento esperado, los usuarios a los que va dirigido, los objetivos a conseguir, una visión de conjunto del sistema y las posibles alternativas a su desarrollo, como por ejemplo, la compra o utilización de otros sistemas ya creados. Posteriormente, se analizan los recursos necesarios y los riesgos y se realiza un presupuesto para el proyecto.

Con todos estos datos, se determina la viabilidad y se realiza la planificación del proyecto, en nuestro caso, mediante un diagrama de Gantt que se utilizará como guía para el desarrollo de las siguientes fases del proyecto.

2.2 Descripción del contexto

Actualmente, la producción mundial se centraliza en Asia y es importada por el resto de países. Por este motivo, cada vez se hacen más necesarios sistemas que permitan una buena gestión de las importaciones y las faciliten en la medida de lo posible.

De la misma manera, es imprescindible planificar la recepción de dichas importaciones en los almacenes de destino y para ello se necesitan previsiones lo más fiables posibles. Además, dado el gran volumen de mercancía transportada, se hace igual de importante la gestión de la calidad del transporte ya que supone uno de los mayores gastos para las empresas importadoras.

2.3 Funcionamiento del sistema

El sistema funciona en dos partes independientes gracias a los diferentes roles de usuario, permitiendo dar una formación más específica según las tareas a realizar por cada actor y disminuyendo el error humano.

Por un lado, puede identificarse la parte de exportación desde la que los usuarios con rol “Export” pueden crear, modificar y borrar tanto artículos, como órdenes y envíos. Además, son los encargados de gestionar las empresas de transporte y los orígenes o centros de exportación.

Por otro lado, la parte de importación es gestionada por los usuarios con rol “Import”, que sólo pueden consultar los artículos, órdenes y envíos pero son los encargados de obtener las previsiones para el almacén de entrega, actualizar las fechas de llegada y analizar la calidad del transporte. También se ocupan del archivo de las entregas ya realizadas y la gestión de los destinos o centros de importación.

Existe otro rol, el de administrador, con acceso total al sistema. Es el único capaz de crear nuevos usuarios y de asignarles rol.

2.4 Perfil de usuario

El perfil de usuario para este sistema es el de una empresa o autónomo que realiza importaciones y necesita conocer en cada momento el estado y calidad de sus envíos.

Es un sistema sencillo e intuitivo por lo que sólo será necesaria una pequeña formación para que cualquier persona con conocimientos mínimos de informática pueda trabajar con él.

2.5 Objetivos

El objetivo general es crear un sistema que permita llevar el control de las importaciones de una empresa en tiempo real.

Como objetivos secundarios, el programa debe ser capaz de ofrecer previsiones de llegada que permitan a los almacenes de destino planificar su plantilla de acuerdo a ellas y además debe permitir analizar la calidad del servicio de transporte para poder utilizar dicha información en futuras negociaciones con navieras, transitarias y otras compañías.

Debe ser también un sistema barato, sencillo, de bajo mantenimiento y debe garantizar la simplicidad de las posibles evoluciones o adaptaciones a requisitos de los usuarios.

2.6 Descripción general del sistema

El sistema consta de una base de datos y una interfaz web.

La primera sirve para almacenar toda la información que se genera y se modifica mediante la interfaz web.

La segunda, por otro lado, es la encargada de interactuar con el usuario y proveerle las funcionalidades necesarias para cumplir los objetivos descritos anteriormente. Ésta debe ser capaz de gestionar artículos, órdenes, envíos, transportes, orígenes y destinos así como de generar previsiones y ayudar a realizar el análisis de la calidad de transporte.

2.7 Recursos necesarios

Se diferencian los recursos necesarios en humanos, hardware y software.

Al tratarse de un proyecto final de carrera el único recurso humano necesario es el del propio alumno que lleva a cabo las tareas de estudio, diseño, codificación, pruebas y documentación del proyecto.

Los recursos de hardware necesarios son:

- Servidor: Alojar la interfaz web y la base de datos.

Los recursos de software necesarios son:

- Servidor Web: Gestión del servidor hardware.
- PhpMyAdmin: Configuración de la base de datos.
- Navegador web: Acceso a la interfaz web.
- Microsoft Office Excel: Diagrama de Gantt.
- Microsoft Office Word: Documentación
- Microsoft Office PowerPoint: Presentación

2.8 Presupuesto

En este punto se realizan dos presupuestos sobre el coste del proyecto. El primero tiene en cuenta precios de mercado mientras que el segundo es el presupuesto real al tratarse de un proyecto final de carrera.

En la siguiente tabla se observa el coste del proyecto a precios de mercado partiendo de la base de que la empresa no dispone de servidor ni de Microsoft Office.

Precio	Concepto
4980€	Recursos humanos (15€/h)
720€/año	Servidor (Hardware) (60€/mes)
Incluido	Servidor Web(Software)
Incluido	PhpMyAdmin (Software)
Gratis	Navegador web
379€	Microsoft Office
6079€	TOTAL

Tabla 2.1. Presupuesto con precios de mercado

La siguiente tabla muestra el coste del proyecto realizado como proyecto final de carrera y utilizando como servidor el PC del alumno.

Precio	Concepto
0€	Recursos humanos (0€/h)
0€/año	Servidor (Hardware)
Gratis	Servidor Web Apache(Software)
Gratis	PhpMyAdmin (Software)
Gratis	Navegador web
0€	Microsoft Office (Licencia estudiante)
0€	TOTAL

Tabla 2.2. Presupuesto proyecto final de carrera

Después de analizar este punto, puede verse que el coste de desarrollo como proyecto final de carrera es nulo y, por lo tanto, la viabilidad económica está asegurada. Es importante destacar que el coste a precios de mercado también es bastante bajo comparado con los importes de las mercancías importadas y más rentable que la adquisición de un sistema ya creado.

2.9 Evaluación de riesgos

Uno de los mayores riesgos es la seguridad de la información que se verá asegurada gracias a la gestión de accesos y de usuarios. Esta situación implica que es indispensable un servidor dedicado al proyecto o un servidor propio de la empresa para alojar la base de datos. Además, deberán realizarse copias de seguridad regularmente para garantizar la reconstrucción de la base de datos en caso de necesitarse.

Otro riesgo importante es el de no poder cumplir con la fecha de entrega ya que es posible que se necesite más tiempo del previsto para codificar alguna de las funciones del sistema o para estudiar el funcionamiento del “framework” y “plugins” utilizados.

2.10 Planificación del proyecto

La planificación del proyecto se ha dividido en varias partes para poder seguir la metodología de desarrollo explicada anteriormente (Punto 1.5). El diseño, codificación y pruebas se ha dividido modularmente y posteriormente se han añadido funcionalidades al sistema siguiendo la misma metodología. En el diagrama de Gantt incluido a continuación se observa el detalle de cada una de las etapas del proyecto y su duración.

2.10.1 Diagrama de Gantt

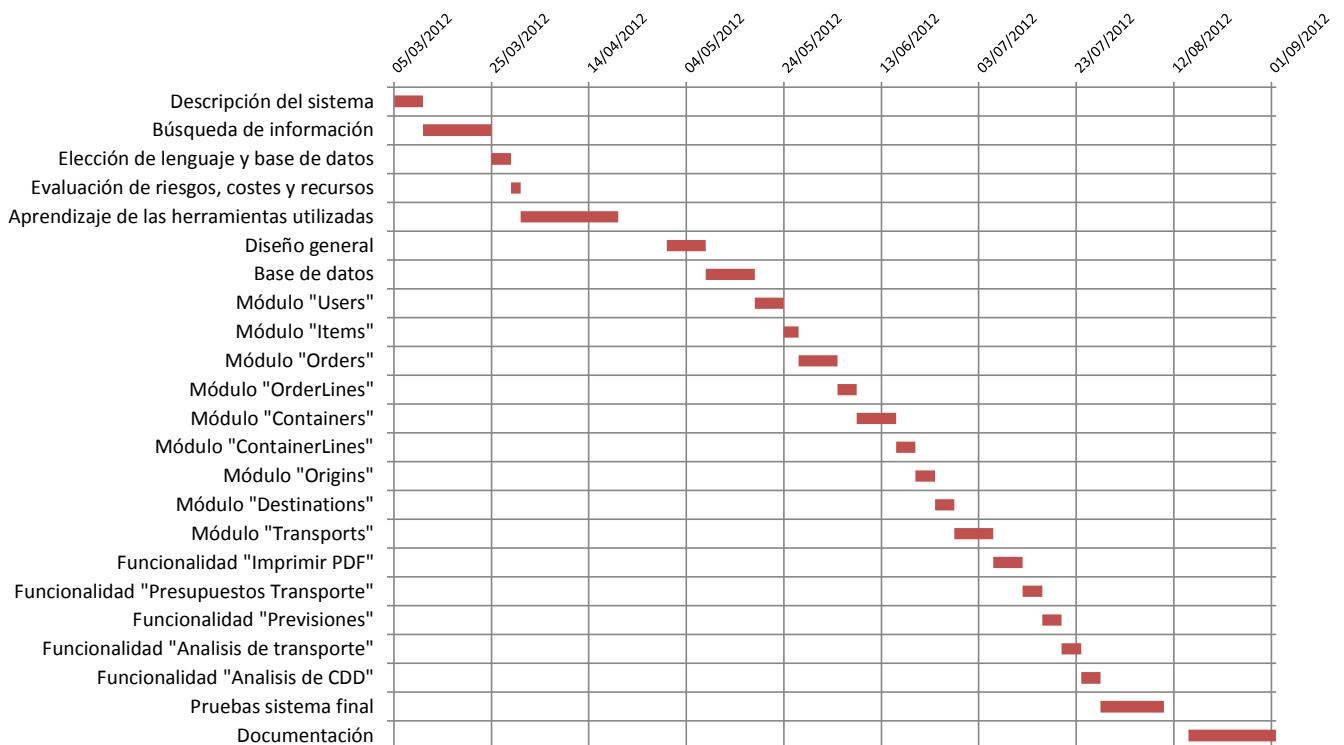


Figura 2.3. Diagrama de Gantt

2.11 Conclusiones

Tras realizar el análisis de viabilidad se decide seguir adelante con el proyecto ya que los costes son nulos, los recursos y objetivos asequibles y los riesgos salvables.

Por lo tanto, se comienza el proyecto utilizando como guía la planificación mostrada por el diagrama de Gantt del punto anterior.

Capítulo III.

Caracterización del

sistema

3.1 Introducción

En este apartado se detallan los componentes del sistema, su estructura, su funcionamiento y el motivo de su elección para este proyecto.

3.2 Componentes

Como ya se ha explicado anteriormente el sistema está formado por una base de datos y una interfaz web ya que esto asegura la compatibilidad con la mayoría de sistemas operativos, reduce los costes y no necesita instalación en los equipos usuarios. Además permite acceso desde teléfonos móviles y otros dispositivos a través de cualquier navegador.

3.2.1 Base de datos

La base de datos escogida para este proyecto es MySQL por su gran velocidad de lectura de datos y su precio gratuito para uso interno, lo que la hace perfecta para aplicaciones web. Además permite una instalación, configuración y mantenimiento muy sencillos.

Para asegurar las funcionalidades del sistema, la base de datos sigue el esquema incluido a continuación, en el que se pueden observar las diferentes tablas y sus relaciones.

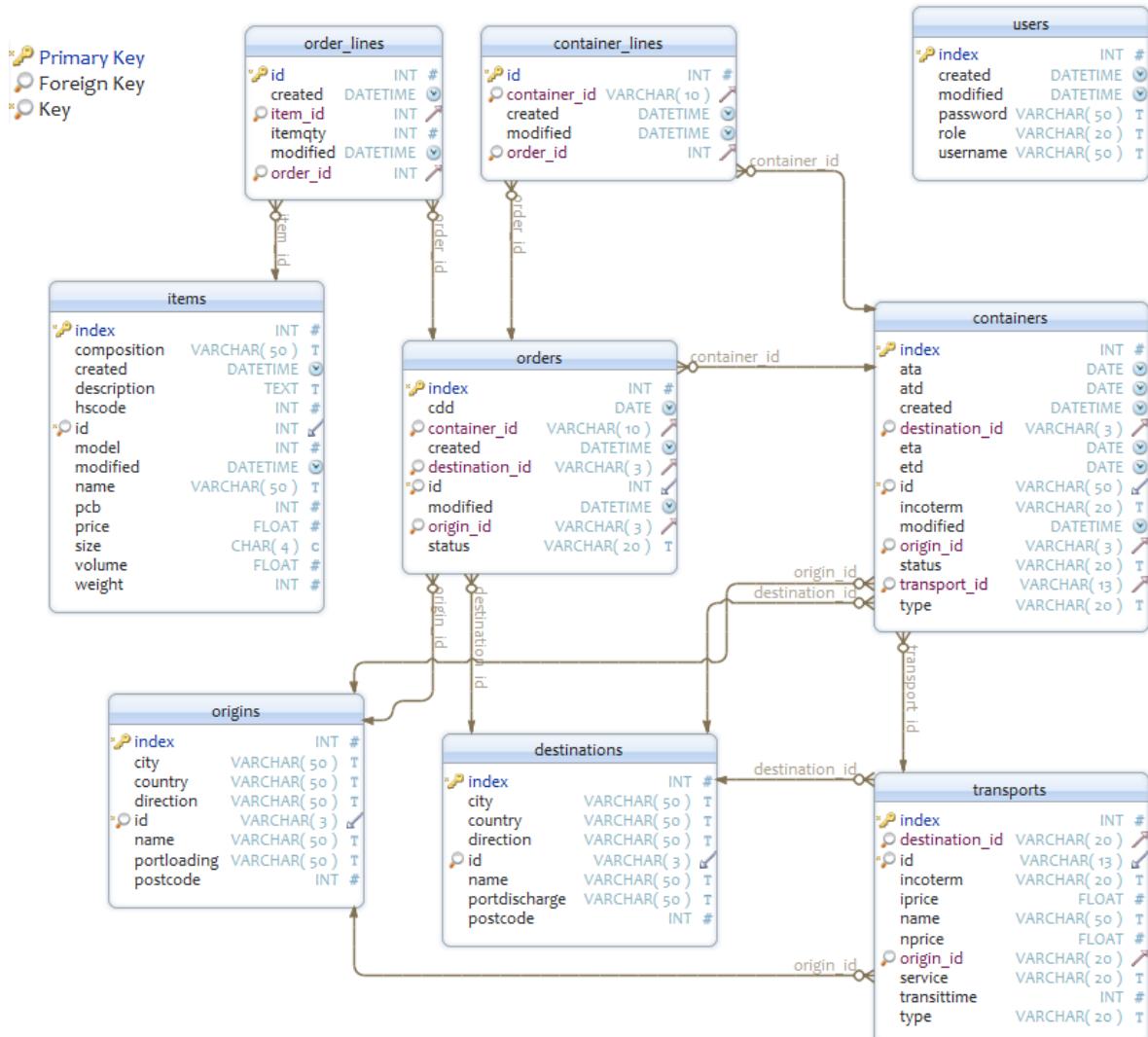


Figura 3.1. Esquema de la base de datos

3.2.1.1 Tabla: users

users			
index	INT #		
created	DATETIME	V	
modified	DATETIME	V	
password	VARCHAR(50)	T	
role	VARCHAR(20)	T	
username	VARCHAR(50)	T	

La tabla “users” recoge la información de los usuarios registrados en el “sistema. Para esta primera versión del proyecto sólo es necesario un usuario (“username”) y contraseña (“password”) para crear un nuevo usuario. Además, se le asigna un rol³ (“role”) según el cual tiene acceso a unas funcionalidades u otras de la interfaz web.

Figura 3.2 Tabla “users”

3.2.1.2 Tabla: items

items			
index	INT #		
composition	VARCHAR(50)	T	
created	DATETIME	V	
description	TEXT	T	
hscode	INT #		
id	INT #		
model	INT #		
modified	DATETIME	V	
name	VARCHAR(50)	T	
pcb	INT #		
price	FLOAT #		
size	CHAR(4)	C	
volume	FLOAT #		
weight	INT #		

En la tabla “items” se almacena la información relativa a los artículos como su nombre, composición, descripción, “HScode⁴” o código taric, modelo, precio, talla, volumen, peso y número identificador. Además, se incluye la cantidad de artículos que caben en una caja (“pcb”). Todos estos campos sirven para crear la documentación adjunta a las órdenes y para el cálculo del volumen, peso y número de cajas de cara a la preparación del transporte.

Como se observa en la imagen, el campo id es clave de la tabla y no puede estar vacío ni repetirse ya que sirve para identificar el artículo y es a donde apuntan las claves externas de las otras tablas.

Figura 3.3. Tabla “items”

3.2.1.3 Tabla: orders

orders			
index	INT #		
cdd	DATE	V	
container_id	VARCHAR(10)	T	
created	DATETIME	V	
destination_id	VARCHAR(3)	T	
id	INT #		
modified	DATETIME	V	
origin_id	VARCHAR(3)	T	
status	VARCHAR(20)	T	

Esta tabla almacena la información genérica de las órdenes introducidas en el sistema. Dicha información es el origen, destino, estado actual, “cdd⁵” y el identificador de la orden.

Existen tres claves externas (“container_id”, “origin_id” y “destination_id”) y una clave primaria (“id”) que ayudan para relacionar las diferentes tablas de la base de datos.

Figura 3.4. Tabla “orders”

3. Rol: Función del usuario del sistema.

4. HScode o código taric: Nomenclatura de mercancías del sistema aduanero común de la Unión Europea que describe de manera inequívoca cada producto y sus características mediante un código numérico.

5. CDD (Contractual Delivery Date): Fecha contractual de entrega de la mercancía en destino.

3.2.1.4 Tabla: order_lines

order_lines	
id	INT #
created	DATETIME
item_id	INT #
itemqty	INT #
modified	DATETIME
order_id	INT #

La tabla “order_lines” almacena las diferentes líneas de cada orden, es decir, los diferentes artículos incluidos en cada pedido junto con sus cantidades.

Los campos “ítem_id” y “order_id” son claves externas mientras que la clave “id” es primaria.

Figura 3.5. Tabla “order_lines”

3.2.1.5 Tabla: container

containers	
index	INT #
ata	DATE
atd	DATE
created	DATETIME
destination_id	VARCHAR(3)
eta	DATE
etd	DATE
id	VARCHAR(50)
incoterm	VARCHAR(20)
modified	DATETIME
origin_id	VARCHAR(3)
status	VARCHAR(20)
transport_id	VARCHAR(13)
type	VARCHAR(20)

Esta es una de las tablas más importantes del proyecto ya que será la más utilizada por los usuarios.

En ella se guarda la información de los envíos y del transporte y permite obtener información sobre la calidad de este último gracias a las fechas estimadas y reales.

Sus campos se pueden dividir en campos de información general: “origin_id”, “destination_id” e “id”; campos de fechas: “etd⁶”, “eta⁷”, “atd⁸”, “ata⁹” y campos de transporte: “transport_id”, “type” e “incoterm¹⁰”.

Figura 3.6. Tabla “containers”

Como muestra la tabla adjunta, las claves externas son “destination_id”, “origin_id” y “transport_id” que apuntan a las tablas con el mismo nombre sin “_id”. La clave “id” servirá como identificador al que apuntan las claves externas de otras tablas y no puede repetirse ni estar vacía como en los casos anteriores.

3.2.1.6 Tabla: container_lines

container_lines	
id	INT #
container_id	VARCHAR(10)
created	DATETIME
modified	DATETIME
order_id	INT #

La funcionalidad de esta tabla es similar a la de “order_lines” pero esta vez entre “orders” y “containers”, es decir, esta tabla almacena las órdenes de cada envío.

Las claves externas, “container_id” y “order_id”, apuntan a sus tablas de referencia mientras que “id” es la clave primaria en este caso.

Figura 3.7. Tabla “container_lines”

6. ETD (Estimated time of departure): Fecha estimada de salida.

7. ETA (Estimated time of arrival): Fecha estimada de llegada.

8. ATD (Actual time of departure): Fecha real de salida.

9. ATA (Actual time of arrival): Fecha real de llegada.

10. Incoterm: Código que representa las condiciones de entrega de las mercancías entre comprador y vendedor.

3.2.1.7 Tabla: origins

origins			
index	INT #		
city	VARCHAR(50)	T	
country	VARCHAR(50)	T	
direction	VARCHAR(50)	T	
id	VARCHAR(3)	T	
name	VARCHAR(50)	T	
portloading	VARCHAR(50)	T	
postcode	INT #		

Figura 3.8. Tabla "origins"

Esta tabla almacena los orígenes desde los que se realizan las exportaciones de mercancía. En ella se encuentra el nombre del centro de exportación, su dirección, código postal, ciudad y país además del puerto de carga y su identificador.

El campo "id" es la clave a la que apuntan las tablas que necesitan relación con "origins".

3.2.1.8 Tabla: destinations

destinations			
index	INT #		
city	VARCHAR(50)	T	
country	VARCHAR(50)	T	
direction	VARCHAR(50)	T	
id	VARCHAR(3)	T	
name	VARCHAR(50)	T	
portdischarge	VARCHAR(50)	T	
postcode	INT #		

Figura 3.9. Tabla "destinations"

Esta tabla almacena los destinos a los que llegarán las importaciones, es decir, el nombre del centro de importación, su dirección, código postal, ciudad y país además del puerto de descarga y su identificador.

El campo "id" es la clave a la que apuntan las tablas que necesitan relación con "destinations".

3.2.1.9 Tabla: transports

transports			
index	INT #		
destination_id	VARCHAR(20)	T	
id	VARCHAR(13)	T	
incoterm	VARCHAR(20)	T	
iprice	FLOAT #		
name	VARCHAR(50)	T	
npice	FLOAT #		
origin_id	VARCHAR(20)	T	
service	VARCHAR(20)	T	
transittime	INT #		
type	VARCHAR(20)	T	

Figura 3.10. Tabla "transports"

La funcionalidad de esta tabla es guardar la información de los diferentes transportes disponibles para realizar las importaciones.

Incluye los campos de origen y destino que permiten filtrar los transportes válidos para cada envío e información sobre la compañía, precio, tipo de servicio, tiempo de tránsito y condiciones del transporte.

Los campos "destination_id" y "origin_id" son claves externas mientras que el campo "id" sirve de identificador y clave para el resto de tablas.

3.2.2 Interfaz web

La interfaz de usuario provee al usuario de las funciones necesarias y le permite interaccionar con él de la manera más sencilla y clara posible.

Para simplificar la programación y el posterior mantenimiento se ha decidido utilizar el patrón de diseño MVC (Modelo-Vista-Controlador) permitiendo así un desarrollo modular e independiente para cada parte del proyecto. Para ayudarnos con ello, se ha utilizado el "framework" CakePHP.

3.2.2.1 Framework: CakePHP

CakePHP es un marco de desarrollo rápido para PHP, libre y de código abierto. Se trata de una estructura que sirve de base para crear aplicaciones web y que provee herramientas que simplifican la programación de las partes comunes a todos los proyectos, como por ejemplo, la gestión de accesos y usuarios, las consultas a la base de datos, etc.

Gracias a la utilización del “framework” sólo se deben codificar las funciones específicas del sistema y ayuda a organizar el código mediante convenciones de nomenclatura y el patrón de diseño MVC (Modelo-Vista-Controlador).

Objeto	Carpeta	Convención
Base de datos	-	Nombre en minúsculas, plural y separando palabras con “_”
Clase	-	Nombre en CamelCase*
Modelo	App/Model	Nombre en singular y CamelCase*
Vista	App/View	Nombre de la carpeta en plural y CamelCase*
Controlador	App/Controller	Nombre en plural, CamelCase* y seguido de “Controller”

*CamelCase: Primera letra de cada palabra en Mayúscula.

Tabla 3.11. Convenciones básicas CakePHP

En la tabla anterior se ven las convenciones más importantes de CakePHP mientras que en el siguiente gráfico se puede observar la arquitectura de software MVC:

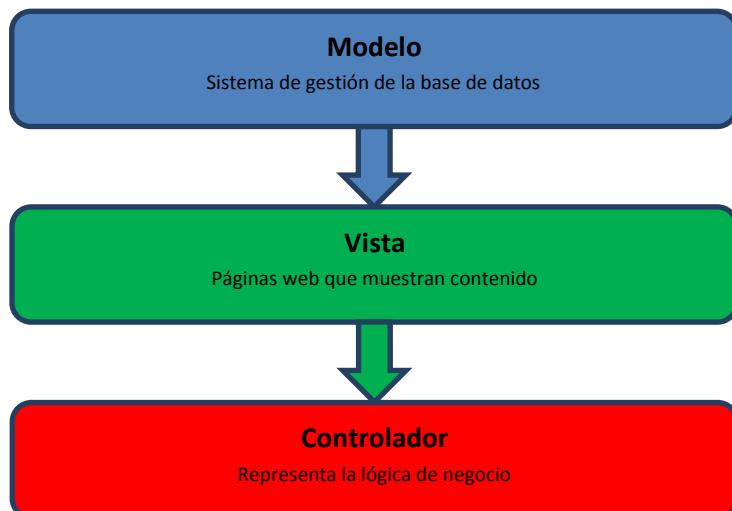


Figura 3.12. Arquitectura de software MVC

Mediante esta arquitectura se consiguen separar la gestión de la base de datos, la lógica de negocio y la interfaz de usuario de manera que todas las peticiones son gestionadas por el controlador que, a su vez, consulta a los modelos y envía la información a las vistas.

3.2.2.1.1 Modelos

Los modelos representan las tablas de las bases de datos y por lo tanto, cada tabla debe tener su propio modelo que definirá las reglas de validación de datos y las relaciones con los otros modelos.

Por este motivo, los modelos son la capa de datos de la aplicación.

3.2.2.1.2 Vistas

Las vistas son las salidas de respuesta enviadas al usuario una vez las peticiones han sido tratadas. Consisten en código HTML, PHP, XML, PDF, etc. legibles por el usuario.

Son consideradas la capa de presentación de la aplicación.

3.2.2.1.3 Controladores

Los controladores gestionan el flujo de la aplicación, por lo que cada petición web será gestionada por un controlador concreto. Éste es el encargado de realizar llamadas a modelos o de ejecutar la lógica de negocio apropiada para finalmente enviar la respuesta a la vista permitiendo así la interacción con el usuario.

Por todo esto, los controladores son la capa de control lógico de la aplicación.

3.2.2.1.4 Relación Modelo-Vista-Controlador

Ahora que ya se conoce cada una de las partes de la arquitectura MVC puede entenderse su funcionamiento en conjunto, mostrado en el siguiente gráfico.

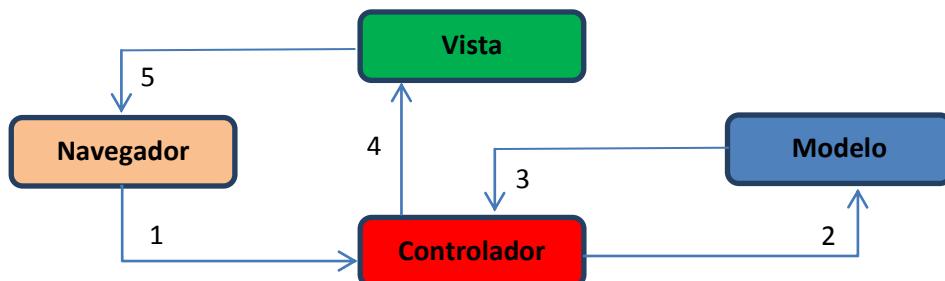


Figura 3.13. Funcionamiento MVC

Primero el usuario realiza una petición a través del navegador y ésta es recibida por el controlador que tras consultar los datos de los modelos y realizar la lógica de negocio envía la respuesta al navegador a través de la vista.

3.2.2.1.5 Seguridad y control de usuarios

Trabajando en CakePHP para gestionar el control de usuarios y accesos, basta con seguir las convenciones del componente “AuthComponent” que proporciona las funcionalidades de login y logout, siempre y cuando la tabla con la lista de usuarios se llame “users” y contenga los campos “username” y “password”. Además realiza la encriptación de éste último.

Al hacer login, el componente Auth almacena los datos del usuario con sesión abierta y reduce el control de acceso a la creación de una función en el controlador que deniegue o acepte la conexión de dicho usuario.

3.2.2.2 Plugin: CakePDF

CakePDF es un conjunto de librerías, gratuito y libre, que junto con CakePHP convierten páginas HTML en documentos PDF descargables desde la interfaz web.

Tras su instalación debe crearse una función en el controlador y una vista de CakePHP que se convertirá en el archivo PDF.

En el siguiente punto se explican dichas funciones y vistas así como las demás de las que se compone el sistema.

Capítulo IV.

Programación de la

interfaz web

4.1 Introducción

Tal y como se ha explicado anteriormente en esta memoria, el sistema se programa en PHP con la ayuda del “framework” CakePHP y el “plugin” CakePDF.

La documentación del código de este proyecto seguirá la misma organización MVC que el sistema en sí, por lo que, a continuación se detallan la programación de cada uno de sus modelos, vistas y controladores.

4.2 Modelos

Siguiendo las convenciones de CakePHP existe un modelo para cada una de las tablas de la base de datos y un modelo genérico llamado “AppModel”.

En los modelos se codifican funciones específicas para gestionar su tabla relacionada así como la configuración de las relaciones entre el modelo actual y el resto.

Asociación	Relación
hasOne	Uno a uno
hasMany	Uno a muchos
belongsTo	Muchos a uno
hasAndBelongsToMany	Muchos a muchos

Tabla 4.1. Relaciones entre modelos CakePHP

En cada modelo, se encuentra también el método de validación de datos que se ejecuta antes de realizar cualquier operación de guardado o modificación en la base de datos. Como ejemplo, se muestra una parte de la codificación de la validación del modelo “users” siendo muy similar para el resto de tablas.

```
// Configuración de la validación de datos
public $validate = array(
    'username' => array(
        'required' => array(
            'rule' => array('notEmpty'),
            'message' => 'A username is required'
        )
),
)
```

Figura 4.2. Modelo User/Validate.php

4.2.1 AppModel.php

Este modelo queda inalterado tal y como viene en la instalación de CakePHP al no ser necesaria ninguna modificación en él para la consecución de los objetivos de nuestro proyecto.

4.2.2 User.php

En este modelo se codifica la validación de datos y se define la función “beforesave”.

4.2.2.1 BeforeSave

Función que se ejecuta de forma automática antes de realizar una operación de guardado en la tabla “users” y que, en este caso, se encarga de encriptar la contraseña antes de almacenarla.

4.2.3 Container.php

En este modelo se codifica la validación de datos y las relaciones con los modelos “ContainerLine”, “Order”, “Transport”, “Origin” y “Destination”. Las dos primeras son relaciones “hasMany” ya que un container puede tener varias órdenes y varias líneas mientras que las tres últimas son relaciones “belongsTo” debido a que un container sólo puede tener un transporte, un origen y un destino.

4.2.4 ContainerLine.php

Se codifica la validación de datos y las relaciones “belongsTo” con los modelos “Container” y “Order” ya que cada línea del contenedor sólo puede pertenecer a un envío y contener como máximo una orden.

4.2.5 Order.php

En este modelo se codifica la validación de datos y varias funciones para esta tabla.

Además, se configuran las relaciones “hasMany” con la tabla “OrderLine” ya que una orden puede tener más de una línea, las relaciones “hasOne” para “ContainerLine” y “belongsTo” para “Container”, “Origin” y “Destination” porque sólo puede tener una de cada.

4.2.5.1 Updatestatus

Recibe un “id” y un “status” por parámetro y modifica el estado de la orden con el identificador recibido en la base de datos.

4.2.5.3 Updatecontainer

Función que recibe los “id” de una orden y de un contenedor y modifica el valor del “container” asociado a la orden recibida en la base de datos.

4.2.6 OrderLine.php

En este modelo se codifica la validación de datos y las relaciones “belongsTo” con los modelos “Item” y “Order” ya que cada línea de la orden sólo puede pertenecer a un pedido y contener como máximo un artículo.

4.2.7 Item.php

En este modelo se codifica la validación de datos y la relación “hasMany” con el modelo “OrderLine” ya que un artículo puede estar en varias líneas de órdenes.

4.2.8 Origin.php

Codifica la validación de datos y las relaciones “hasMany” con los modelos “Order” y “Container” ya que pueden repetirse en varios de ellos.

4.2.9 Destination.php

En este modelo se programa la validación de datos y las relaciones “hasMany” con los modelos “Order” y “Container” ya que pueden repetirse en varios de ellos.

4.2.10 Transport.php

Valida los datos y crea las relaciones “hasMany” con el modelo “Container” y “belongsTo” con los modelos “Order” y “Container”.

4.3 Vistas

Las vistas codifican las páginas web que permiten la interacción con el usuario.

Siguiendo las convenciones de CakePHP se crea una carpeta, con el nombre del modelo en “CamelCase”, donde se aloja una vista para cada función del controlador asociado que requiera mostrar o recibir datos del usuario. El nombre de dicha vista es el nombre de la función en el controlador con la extensión “.ctp”.

A continuación se muestran las vistas creadas para este proyecto.

4.3.1 Users

En este punto se explican las vistas del módulo de gestión de usuarios.

4.3.1.1 Users/login.ctp

Esta vista es la única de todo el sistema a la que puede acceder cualquier persona sin registrarse previamente. Se carga automáticamente al intentar acceder a cualquier página en caso de que no haya ningún usuario con sesión activa. Es también la vista a la que lleva la función de desconexión de usuario o logout.

The screenshot shows a simple login form. At the top, a red banner displays the text "Por favor, introduce tu usuario y contraseña". Below this, there are two input fields: one labeled "Usuario" and another labeled "Contraseña", both represented by empty rectangular boxes. At the bottom of the form is a green "Login" button.

Figura 4.3. Vista Users/login.ctp

Como se observa en la imagen anterior, esta vista consta de un formulario de entrada para los campos usuario y contraseña y un botón de login que ejecuta la acción del controlador.

Por ser el primer formulario comentado en esta memoria, se incluye a continuación su código PHP, que es muy similar al del resto de formularios de la aplicación y que, por lo tanto, no se muestran para simplificar el documento.

```
<div class="users form">
    <?php echo $this->Session->flash('auth'); ?>
    <?php echo $this->Form->create('User'); ?>
    <fieldset>
        <legend>
            <?php
                echo __("Por favor, introduce tu usuario y contraseña");
            ?>
        </legend>
        <?php
            echo $this->Form->input('username', array('label' => 'Usuario'));
            echo $this->Form->input('password', array('label' => 'Contraseña'));
        ?>
    </fieldset>
    <?php echo $this->Form->end(__('Login')); ?>
</div>
```

Figura 4.4. Código de la vista – Users/login.ctp

4.3.1.2 Users/add.ctp

Esta es la vista de creación de nuevos usuarios a la que sólo tienen acceso los administradores del sistema (Rol “Admin”).



Este formulario se titula “Nuevo usuario”. Contiene campos para “Usuario” y “Contraseña”, ambos obligatorios (indicated by asteriscos). Un campo desplegable para “Rol” muestra “Admin” como opción seleccionada. Un botón “Aceptar” en el pie de formulario.

Figura 4.5. Vista Users/add.ctp

Además de los campos usuario y contraseña, esta vez se añade una lista desplegable para asignar el Rol al nuevo usuario, según el cual tendrá unos permisos u otros para interactuar con el sistema. El botón Aceptar ejecuta la función del controlador.

4.3.1.3 Users/index.ctp

Muestra el listado de usuarios registrados en el sistema y las opciones de creación de nuevos usuarios y borrado de existentes. Sólo es accesible por usuarios “Admin”.

Listado de usuarios



Este es un listado de usuarios con los siguientes datos:

	Usuario	Rol	Fecha de registro
<input type="checkbox"/>	admin	admin	2012-07-14 13:20:48
<input type="checkbox"/>	export	export	2012-07-17 00:02:22
<input type="checkbox"/>	import	import	2012-08-30 12:22:44

En la parte inferior del listado hay dos iconos: un “+” para crear un nuevo usuario y un “x” para eliminar uno existente.

Figura 4.6. Vista Users/index.ctp

El menú superior es de navegación y el inferior, de opciones. Ambos se explican con más detalle en las vistas de “Container” ya que es el módulo en el que tienen más opciones activas.

4.3.2 Containers

En este punto se muestran las vistas asociadas a las funciones del modelo “containers”. Al ser este uno de los modelos básicos del sistema es el que cuenta con más funciones y vistas.

4.3.2.1 Containers/index.ctp

Esta es la vista principal del programa a la que se redirige al usuario después de hacer login ya que informa de los envíos pendientes tanto de exportación como de importación.

Existen dos menús, el superior permite la navegación por las diferentes páginas del sistema y ejecutar acciones generales mientras que el inferior realiza acciones sobre una línea previamente seleccionada.

Según el rol de usuario se tienen diferentes opciones en los menús y acceso a diferente información de la base de datos.

Para los usuarios con rol “Export” se muestra el listado de los contenedores que todavía no han sido enviados y los iconos para crear, modificar, eliminar, ver detalles, imprimir y enviar. Además, desde el menú superior se puede acceder a envíos, pedidos, artículos, orígenes y transportes.

Listado de envíos

N. de envío	Origen	Destino	ETD	ATD	ETA	ATA	Unidades	Cajas	Peso	Volumen
100000017	BKK	BCN	2012-09-05				7715	1013	11861.876 Kg	60.6695 m3
100000155	BKK	BCN	2012-09-04				8770	822	7739.064 Kg	51.75868 m3

Figura 4.7. Vista Containers/index.ctp (Rol “Export”)

Para los usuarios con rol “Import” se muestra el listado de los contenedores en camino, es decir, los que ya han sido enviados pero todavía no han llegado al almacén de destino y los iconos para ver detalle, imprimir, modificar fecha de llegada y archivar. Además, desde el menú superior se puede acceder a envíos, pedidos, artículos, destinos, histórico, previsión, histórico y análisis de desviaciones y CDD.

Listado de envíos

N. de envío	Origen	Destino	ETD	ATD	ETA	ATA	Unidades	Cajas	Peso	Volumen
100000005	BKK	BCN	2012-09-12	2012-09-08	2012-10-05		7300	386	2479.275 Kg	9.320444 m3

Figura 4.8. Vista Containers/index.ctp (Rol “Import”)

El rol “Admin” tiene acceso a todas las opciones, tanto de “Export” como de “Import”.

Listado de envíos

N. de envío	Origen	Destino	ETD	ATD	ETA	ATA	Unidades	Cajas	Peso	Volumen
100000005	BKK	BCN	2012-09-12	2012-09-08	2012-10-05		7300	386	2479.275 Kg	9.320444 m3
100000017	BKK	BCN	2012-09-05				7715	1013	11861.876 Kg	60.6695 m3
100000155	BKK	BCN	2012-09-04				8770	822	7739.064 Kg	51.75868 m3

Figura 4.9. Vista Containers/index.ctp (Rol “Admin”)

Además de las opciones arriba descritas, en esta vista se encuentran los iconos de desconexión de usuario y filtro así como las “checkbox” que permiten seleccionar la línea sobre la que aplicar las acciones de los iconos del menú inferior (excepto la opción de crear).

A continuación se adjunta una leyenda de los iconos y las acciones que realizan.

Iconos del menú superior:



Aplicar filtro



Desconexión de usuario

Figura 4.10. Iconos menú superior en vista Containers/index.ctp

Iconos del menú inferior:



Crear



Editar



Eliminar



Ver detalles



Imprimir en PDF



Seleccionar transporte / Enviar



Modificar fecha



Archivar

Figura 4.11. Iconos menú inferior en vista Containers/index.ctp

El menú superior, a excepción de la opción de filtrado, es común para todas las páginas en la que aparece mientras que el menú inferior tiene diferentes opciones activas según el módulo al que pertenece.

La codificación de ambos menús puede consultarse en los ficheros adjuntos en CDROM.

4.3.2.2 Containers/add.ctp

Crear envío

Nº de envío*

Origen
BKK

Destino
BCN

ETD
August 28 2012

Status
Open

Guardar

Esta vista permite crear nuevos envíos asignándoles un número para su identificación y escogiendo entre los diferentes orígenes y destinos disponibles. Además, requiere que se introduzca la fecha prevista de salida y el estado del contenedor, que para esta primera versión del programa es siempre “Open”. Con el botón Guardar se ejecuta la acción del controlador.

Por diseño del sistema, sólo tienen acceso a esta página los usuarios con rol “Export” o “Admin”.

Figura 4.12. Vista Containers/add.ctp

4.3.2.3 Containers/edit.ctp

Esta vista permite editar el origen, destino y fecha prevista de salida de envíos ya existentes.

Se accede a ella tras seleccionar el “checkbox” de algún contenedor, momento en el que se carga la información del envío en cuestión para mostrarla por pantalla.

Al hacer clic en Guardar, se actualiza la base de datos con la información que el usuario tiene en pantalla.

Editar envío

Origen
BKK

Destino
BCN

ETD
August 15 2012

Guardar

Figura 4.13. Vista Containers/edit.ctp

Sólo tienen acceso a esta página los usuarios con rol “Export” o “Admin”.

4.3.2.4 Containers/filter.ctp

Crear filtro

Nº de envío*

Origen

 All

Destino

 All

[Filtrar por fecha de salida](#)

[Filtrar por fecha de llegada](#)

Filtrar

Mediante esta vista se pueden realizar búsquedas y filtros de la tabla “containers”. El sistema permite realizar filtros de un único campo, dejando los demás con el valor por defecto, o de múltiples campos a la vez. La acción se ejecuta al hacer clic en el botón “Filtrar”.

Se incluyen links a las vistas de filtrado por fecha de salida y por fecha de llegada, que se explican a continuación.

Figura 4.14. Vista Containers/filter.ctp

4.3.2.5 Containers/filteretd.ctp

Vista similar a la anterior a la que se añaden los campos de selección de rango de fechas de salida.

Fechas estimadas de salida:

Desde

 August 28 2012

Hasta

 August 28 2012

Figura 4.15. Vista Containers/filteretd.ctp

4.3.2.6 Containers/filtereta.ctp

Fechas estimadas de llegada:

Desde

 August 28 2012

Hasta

 August 28 2012

Mismo caso que el anterior pero con fechas de llegada.

Figura 4.16. Vista Containers/filtereta.ctp

4.3.2.7 Containers/findext.ctp

Es la vista destino de las tres anteriores a la que se redirige al usuario tras aplicar un filtro. Es una copia de “index.ctp” en la que sólo se muestran los resultados del filtro aplicado.

Se mantienen las mismas restricciones de información que en el caso de su homóloga “index.ctp” por lo que los usuarios “Export” sólo tienen acceso a contenedores sin enviar, mientras que a los usuarios con rol “Import” se les listan los contenedores en camino.

4.3.2.8 Containers/forecastrange.ctp

Selección de rango para previsiones:

Desde

 August 28 2012

Hasta

 August 28 2012

Formulario de selección de rango de fechas para previsiones. En él, el usuario debe introducir el rango de fechas estimadas de llegada de las que necesita tener previsión y hacer clic en “Guardar” para ejecutar la acción.

Guardar

Por diseño del sistema los usuarios “Export” no tienen acceso a esta funcionalidad.

Figura 4.17. Vista Containers/forecastrange.ctp

4.3.2.9 Containers/forecast.ctp

La vista “forecast” es la vista de previsiones. En ella, se muestra el listado de contenedores con previsión de llegada entre las fechas seleccionadas y un resumen de la información más relevante, como el número de pedido, número de cajas y unidades totales.

Además, y para simplificar la planificación de actividad, se añade una línea con los totales de los campos anteriormente citados.

ETA	Contenedores	Pedidos	Cajas	Unidades
2012-09-09	10000001	13	1255	85166
2012-09-24	0998884999	5	972	13012
TOTAL	2	18	2227	98178

Figura 4.18. Vista Containers/forecast.ctp

4.3.2.10 Containers/updateata.ctp

Esta vista permite asignar o modificar la fecha real de llegada o “Actual Time of Arrival”. Esto sirve para analizar la calidad del transporte, calcular desviaciones entre el tiempo previsto y el real y obtener información muy valiosa para futuras negociaciones con navieras o transitarias.

Modificar fecha real de llegada:

ATA
August ▾ ▾ 28 ▾ 2012 ▾

Guardar

Figura 4.19. Vista Containers/updateata.ctp

4.3.2.11 Containers/descargar.ctp

La vista “descargar.ctp” es diferente al resto ya que se trata de una página HTML y PHP que el “plugin” CakePDF convierte en un fichero PDF. En ella, se define la estructura de los dos documentos básicos de importación-exportación, la lista de carga y la factura.

En la parte superior de la lista de carga o “packing list” se detalla la información general del pedido. Esto incluye el identificador de la orden, su fecha de creación, direcciones y centros de origen y destino, fechas previstas de salida y llegada, fecha contractual de entrega, identificador del contenedor en el que viaja e incoterm, que define las responsabilidades del transporte. A continuación, se agrega una línea por cada artículo del pedido en la que se informa de su número de identificación, su nombre, talla, cantidad, peso y volumen. Finalmente, se resume la información del pedido mediante el peso, volumen, cantidad y cajas totales.



LISTA DE CARGA

Numero de pedido: 40001	Fecha de creación: 2012-08-15 19:46:44							
Origen: BKK Direccion: Banga Samutprakan St. 52 Codigo postal: 14818 Ciudad: Bangkok Pais: Thailandia	Destino: BCN Direccion: C/ Arana 5 Codigo postal: 8954 Ciudad: Barcelona Pais: Espania							
ETD: 2012-08-15 Puerto de carga: Laem Chabang	ETA: 2012-09-09 Puerto de descarga: Barcelona		Container: 10000001 Incoterm: FOB Bangkok CDD: 2012-09-03					
Articulo	Nombre	Talla	Cajas	PCB	Unidades	Peso	Volumen	
10001	Raqueta SMX-1 JR	.	31	5	150	185.1	1.5	
10002	Raqueta SMX-2 JR	.	35	5	170	315.18	1.7	
10003	Raqueta SMZ-1 SR	.	60	5	295	558.73	2.95	
10004	Raqueta SMZ-2 SR	.	40	5	195	369.33	1.95	
Peso total: 1428.34 Kg Volumen total: 8.1 m3 Unidades totales: 810 Cajas totales: 166								

Figura 4.20. Vista “lista de carga”

En la parte superior de la factura se detalla la misma información que en la lista de carga y se añade la información sobre centro de coste y forma de pago. A continuación se inserta una lista por cada artículo pero, en este caso, toma más importancia el precio y la codificación internacional del producto mediante el HScode. Finalmente, se resume la información de la factura mediante el peso, volumen, cantidad, cajas y precio totales.



FACTURA

Numero de pedido: 40001	Fecha de creación: 2012-08-15 19:46:44							
Origen: BKK Direccion: Banga Samutprakan St. 52 Codigo postal: 14818 Ciudad: Bangkok Pais: Thailandia	Destino: BCN Direccion: C/ Arana 5 Codigo postal: 8954 Ciudad: Barcelona Pais: Espania							
ETD: 2012-08-15 Puerto de carga: Laem Chabang	ETA: 2012-09-09 Puerto de descarga: Barcelona		Container: 10000001 Incoterm: FOB Bangkok CDD: 2012-09-03 Centro de coste: Mismo que destino Fecha de pago: 30 días después de CDD					
Articulo	Nombre	Talla	HS CODE	Cajas	Unidades	Precio unit.	Precio	
10001	Raqueta SMX-1 JR	.	2147483647	31	150	7.2	1080	
10002	Raqueta SMX-2 JR	.	2147483647	35	170	7.5	1275	
10003	Raqueta SMZ-1 SR	.	2147483647	60	295	8.85	2610.75	
10004	Raqueta SMZ-2 SR	.	2147483647	40	195	10.4	2028	
Peso total: 1428.34 Kg Volumen total: 8.1 m3 Unidades totales: 810 Cajas totales: 166 Precio total: 6993.75 EUR								

Figura 4.21. Vista “Factura”

Al tratarse de la función del modelo contenedor, el resultado es un archivo PDF con una factura y una lista de carga para cada orden que éste tenga asignadas.

4.3.2.12 Containers/stored.ctp

Es una copia de “index.ctp” en la que sólo se muestran los contenedores archivados y a la que los usuarios con rol “Export” no tienen acceso por diseño de la aplicación.

4.3.2.13 Containers/desviationsrange.ctp

Consulta de desviaciones:

Fechas de llegada:
Desde
September 1 2012

Hasta
September 1 2012

Transportista
All

Origen
All

Destino
All

Guardar

Formulario de selección de rango de fechas y demás criterios para analizar las desviaciones respecto a la fecha estimada y real de llegada. El usuario debe seleccionar los criterios y hacer clic en “Guardar” para ejecutar la acción.

Para acotar la búsqueda es obligatorio ingresar el rango de fechas mientras que si se dejan las otras opciones por defecto se realiza la consulta sobre todas las posibilidades existentes para esos campos.

Por diseño del sistema los usuarios “Export” no tienen acceso a esta funcionalidad.

Figura 4.22. Vista
Containers/Desviationsrange.ctp

4.3.2.14 Containers/desviations.ctp

Esta vista permite a los usuarios “Import” analizar las desviaciones en fechas de llegada de los contenedores entregados entre el rango de fechas introducido en la página anterior.

El diseño de la página incluye un listado donde se muestran los contenedores con desviación y sus fechas estimada y real de llegada, el nombre del transportista, servicio y tipo de transporte así como los centros de origen y destino.

Al final de esta vista, se incluye una tabla resumen con los totales afectados por estas desviaciones para facilitar al usuario conocer sus repercusiones.

Contenedor	ETA	ATA	Desviación	Transportista	Origen	Destino	Servicio	Tipo
10000001	2012-09-09	2012-09-11	RETRASO	TT2	BKK	BCN	SEA	40HC
0998884999	2012-09-24	2012-09-23	ADELANTO	TT1	BKK	BCN	SEA	40FT

Concepto	%	Contenedores	Pedidos	Cajas	Unidades
TOTAL CON RETRASO	50	1	13	1255	85166
TOTAL CON ADELANTO	50	1	5	972	13012

Figura 4.23. Vista Containers/desviations.ctp

4.3.2.15 Containers/cddontimerange.ctp

Consulta de entregas fuera de plazo:

Desde
September 1 2012

Hasta
September 1 2012

Transportista
All

Origen
All

Destino
All

Guardar

Formulario de selección de rango de fechas y demás criterios para analizar las entregas realizadas fuera de plazo, es decir, pasada su fecha contractual de entrega o “CDD”. El usuario debe seleccionar los criterios y hacer clic en “Guardar” para ejecutar la acción.

Para acotar la búsqueda es obligatorio ingresar el rango de fechas mientras que si se dejan las otras opciones por defecto se realiza la consulta sobre todas las posibilidades existentes para esos campos.

Por diseño del sistema los usuarios “Export” no tienen acceso a esta funcionalidad.

Figura 4.24. Vista
Containers/cddontimerange.ctp

4.3.2.16 Containers/cddontime.ctp

Gracias a esta vista, el usuario puede analizar de manera muy rápida y visual, la calidad de las entregas realizadas entre las fechas y criterios seleccionados en la página anterior.

Primero se muestra un listado que incluye las órdenes con “CDD” perdida, su información más importante, y una línea resumen de los pedidos y unidades entregados fuera de plazo para cada contenedor. Una vez terminado el listado, se incluye una tabla resumen con los pedidos y cantidades totales entregadas y su desglose según se haya respetado o no la “CDD”.

Contenedor	Pedido	Unidades	CDD	ETA	ATA	Origen	Destino	Transportista
10000001	40001	810	2012-09-03	2012-09-09	2012-09-11	BKK	BCN	TT2
10000001	40002	31540	2012-09-10	2012-09-09	2012-09-11	BKK	BCN	TT2
10000001	40003	48816	2012-09-10	2012-09-09	2012-09-11	BKK	BCN	TT2
10000001	3 (60 %)	81166 (95.3 %)						
0998884999	4000021	7300	2012-07-28	2012-09-24	2012-09-23	BKK	BCN	TT1
0998884999	4000022	5712	2012-07-26	2012-09-24	2012-09-23	BKK	BCN	TT1
0998884999	2 (100 %)	13012 (100 %)						

Concepto	Pedidos	% Pedidos	Unidades	% Unidades
CDD Fueras de plazo	5	71.43 %	94178	95.93 %
CDD Dentro de plazo	2	28.57 %	4000	4.07 %
Total recibido	7	-	98178	-

Figura 4.25. Vista Containers/cddontime.ctp

4.3.3 ContainerLines

Este módulo representa los detalles del contenedor por lo que las funciones descritas a continuación se encargan de gestionar dicha información.

4.3.3.1 ContainerLines/view.ctp

Esta vista permite ver los detalles del contenedor, es decir, las órdenes asociadas a él. En caso de que el contenedor no tenga ninguna orden asociada se muestra un aviso y se redirige al usuario hacia la página “ContainerLines/add.ctp” explicada en el punto siguiente.

Este contenedor todavía no tiene pedidos asociados

Figura 4.26. Mensaje de aviso de contenedor sin órdenes asociadas

En caso de que el contenedor ya tenga pedidos asociados se muestra un listado con la información básica de cada orden y una línea resumen con los totales.

Detalles del contenedor: 100000017

User	Envío	Pedidos	Artículos	Orígenes	Transportes	Destinos	Historico	Prevision	Desviaciones	CDD on Time	Actions
#	Pedido	CDD	Unidades	Cajas	Peso	Volumen					
	4000022	2012-07-26	5712	586	8425.455 Kg	37.1723 m3					
	4000158	2012-09-22	2003	427	3436.421 Kg	23.4972 m3					
		Total	7715	1013	11861.876 Kg	60.6695 m3					



Figura 4.27. Vista ContainerLines/view.ctp (Rol “Admin”)

Detalles del contenedor: 100000017

The screenshot shows a table with columns: #, Pedido, CDD, Unidades, Cajas, Peso, and Volumen. It lists two entries: one for Pedido 4000022 on 2012-07-26 with 5712 units, 586 boxes, 8425.455 Kg, and 37.1723 m3; and another for Pedido 4000158 on 2012-09-22 with 2003 units, 427 boxes, 3436.421 Kg, and 23.4972 m3. A summary row at the bottom shows Total: 7715 units, 1013 boxes, 11861.876 Kg, and 60.6695 m3.

#	Pedido	CDD	Unidades	Cajas	Peso	Volumen
■	4000022	2012-07-26	5712	586	8425.455 Kg	37.1723 m3
■	4000158	2012-09-22	2003	427	3436.421 Kg	23.4972 m3
	Total		7715	1013	11861.876 Kg	60.6695 m3



Figura 4.28. Vista ContainerLines/view.ctp (Rol “Export”)

Detalles del contenedor: 100000005

The screenshot shows a table with columns: #, Pedido, CDD, Unidades, Cajas, Peso, and Volumen. It lists one entry for Pedido 4000021 on 2012-07-28 with 7300 units, 386 boxes, 2479.275 Kg, and 9.320444 m3. A summary row at the bottom shows Total: 7300 units, 386 boxes, 2479.275 Kg, and 9.320444 m3.

#	Pedido	CDD	Unidades	Cajas	Peso	Volumen
■	4000021	2012-07-28	7300	386	2479.275 Kg	9.320444 m3
	Total		7300	386	2479.275 Kg	9.320444 m3



Figura 4.29. Vista ContainerLines/view.ctp (Rol “Import”)

El menú superior es común al de “Containers/Index.ctp” mientras que en el menú inferior sólo se activan las opciones para añadir, eliminar y ver detalles. Las tres son visibles desde los roles de “Export” y “Admin” mientras que para el rol de “Import” sólo estará disponible la última, ver detalles, dado que los contenedores a los que tiene acceso ya están cerrados y no pueden ser modificados.

4.3.3.2 ContainerLines/add.ctp

Mediante esta vista se adjuntan órdenes a los contenedores. Para ello, el aplicativo muestra el listado de pedidos compatibles con el contenedor en cuestión, es decir, con el mismo origen, destino y que todavía no tienen envío asociado.

Adjuntar pedido al contenedor actual

Pedido*

4000158
4000158
4000159

Guardar

Una vez seleccionada la orden se ejecuta la acción al pulsar el botón “Guardar” y la aplicación redirige a la vista “ContainerLines/view.ctp”.

Por diseño del sistema el rol “Import” no tiene acceso a esta página ya que sólo gestiona contenedores ya enviados.

4.3.4 Orders

Las órdenes o pedidos son otra de las partes básicas de toda importación. Permiten almacenar la información sobre qué artículos y qué cantidad deben enviarse a cada destino, además de otra información sobre fechas contractuales y transporte.

En este punto se detallan las vistas para la gestión de pedidos.

4.3.4.1 Orders/index.ctp

Gracias a esta vista, el usuario puede ver el listado de pedidos creados en el sistema.

Tal y como pasa con “Containers/index.ctp”, la lista mostrada y los menús dependen del rol del usuario. Para el rol “Export” se muestran las órdenes pendientes de envío mientras que para los usuarios “Import” se muestran las órdenes enviadas. “Admin”, como se ha mencionado a lo largo de este documento, tiene acceso a ambos listados.

Listado de pedidos

Listado de pedidos									
Pedido	Container	Origen	Destino	CDD	Unidades	Cajas	Peso	Volumen	
■ 4000021	100000005	BKK	BCN	2012-07-28	7300	386	2479.275	9.320444	    
■ 4000022	100000017	BKK	BCN	2012-07-26	5712	586	8425.455	37.1723	
■ 4000158	100000017	BKK	BCN	2012-09-22	2003	427	3436.421	23.4972	

Figura 4.31. Vista Orders/index.ctp (Rol “Admin”)

Listado de pedidos

Listado de pedidos									
Pedido	Container	Origen	Destino	CDD	Unidades	Cajas	Peso	Volumen	
■ 4000022	100000017	BKK	BCN	2012-07-26	5712	586	8425.455	37.1723	    
■ 4000158	100000017	BKK	BCN	2012-09-22	2003	427	3436.421	23.4972	

Figura 4.32. Vista Orders/index.ctp (Rol “Export”)

Listado de pedidos

Listado de pedidos									
Pedido	Container	Origen	Destino	CDD	Unidades	Cajas	Peso	Volumen	
■ 4000021	100000005	BKK	BCN	2012-07-28	7300	386	2479.275	9.320444	 

Figura 4.33. Vista Orders/index.ctp (Rol “Import”)

De nuevo se observa que el menú superior es común a los descritos anteriormente mientras que el menú inferior tiene activadas las opciones de creación, edición, borrado, visualización de detalles e impresión; todas visibles para “Export” y “Admin” mientras que “Import” solamente tiene acceso a ver detalles e imprimir.

4.3.4.2 Orders/add.ctp

Crear Pedido

Pedido*	<input type="text"/>
Origen	BKK
Destino	BCN
CDD	August 29 2012
Estado	New
<input type="button" value="Guardar"/>	

Esta vista permite crear nuevas órdenes asignándoles un número para su identificación y escogiendo entre los diferentes orígenes y destinos disponibles. Además requiere que se introduzca la fecha contractual de entrega, que sirve de referencia a la hora de realizar el envío, y el estado de la orden que para esta primera versión del programa es siempre “New”. Con el botón Guardar se ejecuta la acción del controlador.

Por diseño del sistema sólo tienen acceso a esta página los usuarios con rol “Export” o “Admin”.

Figura 4.34. Vista Orders/add.ctp

4.3.4.3 Orders/edit.ctp

Esta vista permite editar el origen, destino y fecha contractual de entrega de envíos ya existentes.

Se accede a ella tras seleccionar el “checkbox” de algún pedido, momento en el que se carga la información de la orden para mostrarla por pantalla.

El botón “Guardar” inicia la actualización de los datos.

Por diseño del sistema sólo tienen acceso a esta página los usuarios con rol “Export” o “Admin”.

Editar pedido

Origen	BKK
Destino	BCN
CDD	September 3 2012
Estado	New
<input type="button" value="Guardar"/>	

Figura 4.35. Vista Orders/edit.ctp

4.3.4.4 Orders/descargar.ctp

Vista igual a la de “Containers/descargar.ctp” pero que sólo muestra una lista de carga y una factura que se corresponden con la única orden seleccionada.

4.3.5 OrderLines

Este módulo representa los detalles del pedido por lo que las funciones descritas a continuación se encargan de gestionar dicha información.

4.3.5.1 OrderLines/view.ctp

En esta vista se muestran los detalles de un pedido, es decir, la lista de artículos que lo forman junto con la información más relevante de cada uno de ellos, como pueden ser cantidad total, número de cajas, peso y volumen.

Detalles del pedido: 4000021



#	Artículo	Nombre producto	Cajas	Unidades	Peso	Volumen
■	10008	Pelota AZ750	7	1231	369.3 Kg	0.39392 m3
■	10025	Bolsa T7580	149	1485	1536.975 Kg	8.7615 m3
■	10030	Grip Confort 120	230	4584	573 Kg	0.165024 m3
Total			386	7300	2479.275 Kg	9.320444 m3

Figura 4.36. Vista “OrderLines/view.ctp” (Rol “Admin”)

Detalles del pedido: 4000022



#	Artículo	Nombre producto	Cajas	Unidades	Peso	Volumen
■	10020	Bolsa T8870	137	1231	1569.525 Kg	8.4939 m3
■	10026	Bolsa R382	449	4481	6855.93 Kg	28.6784 m3
Total			586	5712	8425.455 Kg	37.1723 m3

Figura 4.37. Vista “OrderLines/view.ctp” (Rol “Export”)

Detalles del pedido: 4000021



#	Artículo	Nombre producto	Cajas	Unidades	Peso	Volumen
■	10008	Pelota AZ750	7	1231	369.3 Kg	0.39392 m3
■	10025	Bolsa T7580	149	1485	1536.975 Kg	8.7615 m3
■	10030	Grip Confort 120	230	4584	573 Kg	0.165024 m3
Total			386	7300	2479.275 Kg	9.320444 m3

Figura 4.38. Vista “OrderLines/view.ctp” (Rol “Import”)

Nuevamente, se observa que el menú superior es común y el inferior sólo tiene determinadas funciones activadas. En este caso, los usuarios “Export” y “Admin” podrán crear, editar y eliminar líneas de pedidos mientras que los usuarios “Import” únicamente podrán ver el listado.

4.3.5.2 OrderLines/add.ctp

Crear linea

Articulo*
10001

Cantidad*

↑ ↓

Guardar

Esta vista permite agregar artículos a los pedidos. Para ello el usuario debe seleccionar el artículo dentro del listado desplegable e ingresar la cantidad ordenada. Posteriormente se ejecuta la acción al hacer clic en “Guardar”.

Por diseño del sistema sólo tienen acceso a esta página los usuarios con rol “Export” o “Admin”.

Figura 4.39. Vista OrderLines/add.ctp

4.3.5.3 OrderLines/edit.ctp

Gracias a esta vista, el usuario puede modificar la cantidad y el artículo pedidos.

Se accede a ella desde “OrderLines/view.ctp” tras seleccionar el “checkbox” de alguna línea y hacer clic en el botón para editar. En ese momento, el aplicativo carga la información de la línea seleccionada para mostrarla por pantalla. Haciendo clic en guardar se procede a realizar los cambios en la base de datos.

Solamente pueden acceder a esta página los usuarios con rol “Export” o “Admin”.

Modificar linea de pedido

Articulo*
10001

Cantidad*

↑ ↓

Guardar

Figura 4.40. Vista OrderLines/edit.ctp

4.3.6 Items

En este módulo se gestiona la base de datos de artículos del sistema. Para ello, se utilizan las siguientes vistas.

4.3.6.1 Items/index.ctp

Mediante esta vista se muestra el listado de artículos dados de alta en el sistema.

Listado de artículos

Articulo	Nombre	Descripcion	Talla	Modelo	HSCODE	Peso (g)	Volumen (m3)	PCB (Uds/Caja)	Precio (EUR)
10001	Raqueta SMX-1 JR	Raqueta de tenis - Nivel 1 JR	.	1231	2147483647	1234	0.01	5	7.3
10002	Raqueta SMX-2 JR	Raqueta de tenis - Nivel 2 JR	.	1232	2147483647	1854	0.01	5	7.5
10003	Raqueta SMZ-1 SR	Raqueta de tenis - Nivel 1	.	1233	2147483647	1894	0.01	5	8.85

1 | 2



Figura 4.41. Vista “Items/index.ctp” (Rol “Admin”)

Listado de artículos

Artículo	Nombre	Descripción	Talla	Modelo	HS CODE	Peso (g)	Volumen (m3)	PCB (Uds/Caja)	Precio (EUR)
■ 10001	Raqueta SMX-1 JR	Raqueta de tenis - Nivel 1 JR	.	1231	2147483647	1234	0.01	5	7.3
■ 10002	Raqueta SMX-2 JR	Raqueta de tenis - Nivel 2 JR	.	1232	2147483647	1854	0.01	5	7.5
■ 10003	Raqueta SMZ-1 SR	Raqueta de tenis - Nivel 1	.	1233	2147483647	1894	0.01	5	8.85

1 | 2



Figura 4.42. Vista "Items/index.ctp" (Rol "Export")

Listado de artículos

Código	Nombre	Descripción	Talla	Modelo	HS CODE	Peso (g)	Volumen (m3)	PCB	Precio
■ 10001	Raqueta SMX-1 JR	Raqueta de tenis - Nivel 1 JR	.	1231	2147483647	1234	0.01	5	7.3
■ 10002	Raqueta SMX-2 JR	Raqueta de tenis - Nivel 2 JR	.	1232	2147483647	1854	0.01	5	7.5
■ 10003	Raqueta SMZ-1 SR	Raqueta de tenis - Nivel 1	.	1233	2147483647	1894	0.01	5	8.85

1 | 2 | 3 | 4 | 5 | 6 | 7

Figura 4.43. Vista "Items/index.ctp" (Rol "Import")

Como se comentó anteriormente, el menú superior es común al resto de módulos pero esta vez se añade un ícono que permite imprimir toda la lista de artículos en PDF. El menú inferior, en este caso, dispone de las opciones para crear y editar para los usuarios "Export" y "Admin" mientras que está desactivado para los usuarios "Import".

4.3.6.2 Items/add.ctp

Crear Artículo

Código*	HSCODE*
<input type="text"/>	<input type="text"/>
Nombre*	Peso*
<input type="text"/>	<input type="text"/>
Descripción*	Volumen*
<input type="text"/>	<input type="text"/>
Talla*	Artículos por caja*
<input type="text"/>	<input type="text"/>
Modelo*	Precio*
<input type="text"/>	<input type="text"/>
<input type="button" value="Guardar"/>	

En esta vista el usuario debe introducir los datos de los nuevos artículos. Al hacer clic en "Guardar" se crea una nueva línea en la base de datos con la información en pantalla.

Por diseño del sistema, esta página sólo es accesible para los usuarios con rol "Export" o "Admin".

Figura 4.44. Vista Items/add.ctp

4.3.6.3 Items/edit.ctp

A esta vista se accede desde “Items/index.ctp” tras seleccionar el “checkbox” de alguna línea y hacer clic en el botón de edición. En ese momento, el aplicativo carga la información de la línea seleccionada para mostrarlo por pantalla.

Tras realizar los cambios oportunos, se debe hacer clic en el botón “Guardar” para almacenarlos.

Por diseño del sistema sólo tienen acceso a esta página los usuarios con rol “Export” o “Admin”.

Figura 4.45. Vista Items/edit.ctp

4.3.6.4 Items/descargar.ctp

Esta vista permite al usuario obtener en PDF un listado de todos los artículos del sistema.

ID	Nombre	Descripción	Talla	Modelo	HSCODE	Peso	Volumen	PCB	Precio
10001	Raqueta SMX-1 JR	Raqueta de tenis - Nivel 1 JR	.	1231	2147483647	1234	0.01	5	7.2
10002	Raqueta SMX-2 JR	Raqueta de tenis - Nivel 2 JR	.	1232	2147483647	1854	0.01	5	7.5
10003	Raqueta SMZ-1 SR	Raqueta de tenis - Nivel 1	.	1233	2147483647	1894	0.01	5	8.85
10004	Raqueta SMZ-2 SR	Raqueta de tenis - Nivel 2	.	1234	2147483647	1894	0.01	5	10.4
10005	Raqueta SMZ-3 SR	Raqueta de tenis - Nivel 3	.	1235	2147483647	2048	0.01	5	12.25
10006	Pelota AZ700	Pelota de tenis - N.1	.	12354	2147483647	275	0.00032	200	0.15
10007	Pelota AZ720	Pelota de tenis - N.2	.	12356	2147483647	278	0.00032	200	0.26
10008	Pelota AZ750	Pelota de tenis - N.3	.	12355	2147483647	300	0.00032	200	0.57
10009	Pelota AB1500	Pelota de tenis - N.3	.	12358	2147483647	305	0.00032	200	0.95
10010	Pelota AB1700	Pelota de tenis - N.4	.	12359	2147483647	307	0.00032	200	1.13
10011	Red tenis	Red para pista de tenis	.	12372	2147483647	7057	0.055	1	3.48
10020	Bolsa T8870	Funda raqueta tenis	.	12359	2147483647	1275	0.0069	9	6.58
10021	Bolsa T9870	Funda raqueta tenis	.	12361	2147483647	405	0.0038	15	2.58
10022	Bolsa Z585	Funda raqueta tenis	.	12362	2147483647	897	0.0085	6	2.57
10023	Bolsa Z800	Funda raqueta tenis	.	12363	2147483647	924	0.0063	10	1.28
10024	Bolsa Y920	Funda raqueta tenis	.	12364	2147483647	485	0.0067	10	1.26
10025	Bolsa T7580	Funda raqueta tenis	.	12365	2147483647	1035	0.0059	10	4.2
10026	Bolsa R382	Funda raqueta tenis	.	12366	2147483647	1530	0.0064	10	3.58

page 1 / 2

Figura 4.46. Vista “Listado de artículos” en PDF.

La lista creada se ordena ascendentemente por número de ID.

4.3.7 Origins

Se utiliza este módulo para gestionar los orígenes o centros de exportación.

Por diseño del sistema, los usuarios “Import” no pueden acceder a ninguna de las páginas de este modelo, detalladas a continuación.

4.3.7.1 Origins/index.ctp

Mediante esta página se muestra al usuario el listado de orígenes disponibles y las opciones de creación, edición y eliminación de éstos.

Listado de orígenes

Usuarios Envíos Pedidos Artículos Orígenes Transportes Destinos Historico Prevision Desviaciones CDD on Time D							
Código	Origen	Dirección	Cod. Postal	Ciudad	País	Puerto de carga	
<input type="checkbox"/>	VTM	DPP Vietnam	Cokngoen St. 23	2058	Ho Chi Minh	Vietnam	Ho chi minh city
<input type="checkbox"/>	CHN	DPP China	Gangme Poe 32	48168	Shanghai	China	Shanghai
<input type="checkbox"/>	BKK	DPP Thailand	Banga Samutprakan St. 52	14818	Bangkok	Thailandia	Laem Chabang



Figura 4.47. Vista Origins/index.ctp (Rol “Admin”).

Listado de orígenes

Envíos Pedidos Artículos Orígenes Transportes D							
Código	Origen	Dirección	Cod. Postal	Ciudad	País	Puerto de carga	
<input type="checkbox"/>	VTM	DPP Vietnam	Cokngoen St. 23	2058	Ho Chi Minh	Vietnam	Ho chi minh city
<input type="checkbox"/>	CHN	DPP China	Gangme Poe 32	48168	Shanghai	China	Shanghai
<input type="checkbox"/>	BKK	DPP Thailand	Banga Samutprakan St. 52	14818	Bangkok	Thailandia	Laem Chabang



Figura 4.48. Vista Origins/index.ctp (Rol “Export”).

La vista es la misma para usuarios con rol “Export” y “Admin” a excepción del menú superior de navegación en el que cada uno tiene las mismas opciones que en módulos anteriores.

4.3.7.2 Origins/add.ctp

Crear Origen

Código de origen (3 letras)*	<input type="text"/> Ciudad*
<input type="text"/> Origen*	<input type="text"/> País*
Dirección*	<input type="text"/> Puerto de carga*
<input type="text"/> Cod. Postal*	<input type="button" value="Guardar"/>

Vista de creación de nuevos orígenes. El usuario debe llenar los campos y al pulsar en “Guardar” se crea una nueva línea en la tabla “Origins” de la base de datos con la información introducida en el formulario.

Figura 4.49. Vista Origins/add.ctp

4.3.7.3 Origins/edit.ctp

A esta vista se accede desde “Origins/index.ctp” tras seleccionar el “checkbox” de alguna línea y hacer clic en el botón editar. En ese momento, el aplicativo carga la información de la línea seleccionada para mostrarlo por pantalla y, tras realizar las modificaciones necesarias, el usuario debe hacer clic en “Guardar” para actualizar la base de datos.

Editar origen

Código de origen (3 letras)* VTM	Ciudad* Ho Chi Minh
Origen* DPP Vietnam	País* Vietnam
Dirección* Coknogen St. 23	Puerto de carga* Ho chi minh city
Cod. Postal* 2058	Guardar

Figura 4.50. Vista Origins/edit.ctp

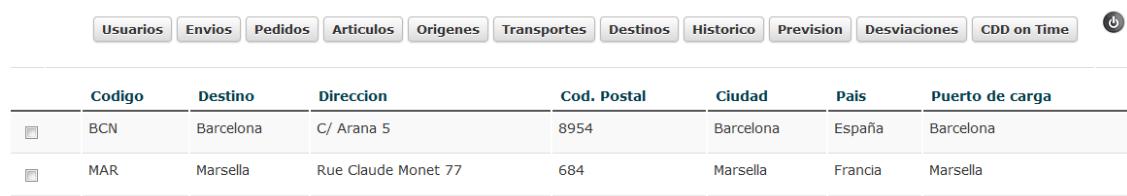
4.3.8 Destinations

En este punto se detallan las vistas de gestión de destinos a las que los usuarios “Export” no tienen acceso por diseño del sistema.

4.3.8.1 Destinations/index.ctp

Mediante esta página se muestra al usuario el listado de destinos disponibles y las opciones de creación, edición y eliminación de éstos.

Listado de destinos



	Código	Destino	Dirección	Cod. Postal	Ciudad	País	Puerto de carga
<input type="checkbox"/>	BCN	Barcelona	C/ Arana 5	8954	Barcelona	España	Barcelona
<input type="checkbox"/>	MAR	Marsella	Rue Claude Monet 77	684	Marsella	Francia	Marsella

Figura 4.51. Vista Destinations/index.ctp (Rol “Admin”)

Listado de destinos



	Código	Destino	Dirección	Cod. Postal	Ciudad	País	Puerto de carga
<input type="checkbox"/>	BCN	Barcelona	C/ Arana 5	8954	Barcelona	España	Barcelona
<input type="checkbox"/>	MAR	Marsella	Rue Claude Monet 77	684	Marsella	Francia	Marsella

Figura 4.52. Vista Destinations/index.ctp (Rol “Import”)

La vista es similar para usuarios con rol “Import” y “Admin” a excepción del menú superior de navegación en el que cada uno tiene las mismas opciones que en módulos anteriores.

4.3.8.2 Destinations/add.ctp

Crear Destino

Código de origen (3 letras)*	Ciudad*
<input type="text"/>	<input type="text"/>
Destino*	País*
<input type="text"/>	<input type="text"/>
Dirección*	Puerto de descarga*
<input type="text"/>	<input type="text"/>
Cod. Postal*	
<input type="text"/>	<input type="button" value="Guardar"/>

Vista de creación de nuevos destinos. El usuario debe llenar los campos y al pulsar en “Guardar” se crea una nueva línea en la tabla “destinations” de la base de datos con la información de pantalla.

Figura 4.53. Vista Destinations/add.ctp

4.3.8.3 Destinations/edit.ctp

A esta vista se accede tras seleccionar un “checkbox” de alguna de las líneas de la página “Destinations/index.ctp” y hacer clic en el botón editar. En ese momento, el aplicativo carga la información de la línea seleccionada para mostrarlo por pantalla. Una vez terminadas las modificaciones el sistema actualiza la base de datos al hacer clic en “Guardar”.

Editar Destino

Código de origen (3 letras)*	Ciudad*
<input type="text"/> BCN	<input type="text"/> Barcelona
Destino*	País*
<input type="text"/> Barcelona	<input type="text"/> España
Dirección*	Puerto de descarga*
<input type="text"/> C/ Arana 5	<input type="text"/> Barcelona
Cod. Postal*	
<input type="text"/> 8954	<input type="button" value="Guardar"/>

Figura 4.54. Vista Destinations/edit.ctp

4.3.9 Transports

A continuación se muestran las páginas de gestión del transporte, accesibles sólo por los usuarios con rol “Export” o “Admin”.

4.3.9.1 Transports/index.ctp

Esta vista ofrece al usuario el listado de transportes introducidos en el sistema y le da opción de crear nuevos o modificar y borrar los existentes.

Listado de transportes:

Listado de Transportes									
	Transportista	Origen	Destino	Servicio	Tipo	Incoterm	Tiempo transito	Precio Internacional	Precio nacional
<input type="checkbox"/>	TT1	BKK	BCN	SEA	20FT	FOB Bangkok	27 días	1080 €	320 €
<input type="checkbox"/>	TT1	BKK	BCN	SEA	40FT	FOB Bangkok	27 días	2080 €	320 €
<input type="checkbox"/>	TT1	BKK	BCN	SEA	40HC	FOB Bangkok	27 días	2250 €	380 €



Figura 4.55. Vista Transports/index.ctp (Rol “Admin”)

Listado de transportes:

Transportista	Origen	Destino	Servicio	Tipo	Incoterm	Tiempo transito	Precio Internacional	Precio nacional
□ TT1	BKK	BCN	SEA	20FT	FOB Bangkok	27 dias	1080 €	320 €
□ TT1	BKK	BCN	SEA	40FT	FOB Bangkok	27 dias	2080 €	320 €
□ TT1	BKK	BCN	SEA	40HC	FOB Bangkok	27 dias	2250 €	380 €



Figura 4.56. Vista Transports/index.ctp (Rol “Export”)

De la misma manera que se ha mencionado en los módulos anteriores, el menú superior es común y tiene unas u otras opciones según el rol del usuario. El menú inferior es igual para “Export” y “Admin”.

4.3.9.2 Transports/add.ctp

Figura 4.57. Vista Transports/add.ctp

En esta vista el usuario puede crear nuevos transportes. Para ello debe introducir la empresa, el origen, destino, tipo de contenedor, tiempo de tránsito y precios internacional y nacional.

Para esta primera versión del programa, se asume que todos los envíos son marítimos y se realizan bajo el incoterm “FOB¹¹” por lo que estos campos no se muestran en este formulario por el momento.

4.3.9.3 Transports/edit.ctp

A esta vista se accede tras seleccionar un “checkbox” de alguna de las líneas de la página “Transports/index.ctp” y hacer clic en el botón editar. En ese momento, el aplicativo carga la información de la línea seleccionada para mostrarlo por pantalla.

Al hacer clic en “Guardar” se actualiza la base de datos con la información en pantalla.

Editar transporte

Figura 4.58. Vista Transports/edit.ctp

4.3.9.4 Transports/budget.ctp

Esta página muestra los transportes compatibles con el contenedor seleccionado y permite al usuario seleccionar el que más convenga o crear uno nuevo. Se accede a ella tras seleccionar un envío del listado de “Containers/index.html” y hacer clic en el icono de enviar.

11. FOB (Free on Board): Incoterm que limita la responsabilidad del vendedor hasta la carga de la mercancía en el barco. A partir de ese momento toda responsabilidad sobre la mercancía pasa a ser del comprador.

Dado que la selección del transporte se realiza antes del envío, el programa se diseña de tal manera que los usuarios de importación no tengan acceso a esta página.

Listado de transportes:

#	Transportista	Servicio	Tipo	Llenado	Incoterm	Transito	ETD	ETA	CDD	Flete	Entrega	Total
<input type="checkbox"/>	TT1	SEA	40FT	89.62%	FOB Bangkok	27 dias	2012-09-05	2012-10-07	0%	2080 €	320 €	2400 €
<input type="checkbox"/>	TT2	SEA	40FT	89.62%	FOB Bangkok	25 dias	2012-09-05	2012-10-05	0%	2120 €	335 €	2455 €

Figura 4.59. Vista Transports/budget.ctp (Rol “Admin”)

Listado de transportes:

#	Transportista	Servicio	Tipo	Llenado	Incoterm	Transito	ETD	ETA	CDD	Flete	Entrega	Total
<input type="checkbox"/>	TT1	SEA	40FT	89.62%	FOB Bangkok	27 dias	2012-09-05	2012-10-07	0%	2080 €	320 €	2400 €
<input type="checkbox"/>	TT2	SEA	40FT	89.62%	FOB Bangkok	25 dias	2012-09-05	2012-10-05	0%	2120 €	335 €	2455 €

Figura 4.60. Vista Transports/budget.ctp (Rol “Export”)

4.4 Controladores

En este apartado de la memoria se explica el código de las funciones más significativas de cada controlador del proyecto. No se incluye el código completo de todas las funciones ya que éste se puede consultar en el CDROM adjunto.

Para garantizar la integridad de los datos, se han incluido restricciones en las funciones de borrado que impiden su ejecución si la línea a borrar está asociada a líneas de otras tablas. Por ejemplo, no se podrá eliminar un origen si existe alguna orden que lo tenga asociado.

4.4.1 AppController.php

Es el controlador base generado en la instalación de CakePHP del que heredan todos los demás. Para este proyecto, se añade la función “isAuthorized” a este controlador para gestionar el control de acceso a las diferentes páginas del sistema.

4.4.1.1 AppController: Función isAuthorized

Esta función comprueba si el usuario en sesión tiene rol y si éste es “Admin”. En caso positivo devuelve “true”, es decir, autoriza al usuario a ver todas las páginas. En caso negativo, deniega el acceso a todas las páginas y muestra el aviso correspondiente.

Como puede verse en el primer comentario de la figura mostrada a continuación, “isAuthorized” configura el acceso general y en cada controlador se creará una función para gestionar el acceso a cada acción de manera individual.

```

// Control de acceso general
// En cada controlador se asignarán los permisos según el rol
public function isAuthorized($user) {
    // Configuración de acceso del rol admin
    if (isset($user['role']) && $user['role'] === 'admin') {
        return true;
    }
    // Denegar el acceso por defecto al resto de roles
    $this->Auth->authError = "AVISO: No estas autorizado a ver esta página";
    $this->Session->setFlash($this->Auth->authError);
    return false;
}

```

Figura 4.61. AppController – Función isAuthorized

4.4.2 UsersController.php

Este controlador se encarga de la gestión de usuarios y por lo tanto, sus funciones más importantes son login, logout y creación de usuarios.

4.4.2.1 UsersController: Función login

La función login permite al usuario abrir su sesión en el sistema. Primero se muestra el formulario de la vista “login.ctp” y se espera a que el usuario lo complete con sus datos de registro. Posteriormente se comprueban los datos y se redirige o muestra el mensaje de error según la información introducida.

4.4.2.2 UsersController: Función logout

Esta función cierra la sesión actual y redirige a la página de “login” ya que es la única accesible para usuarios no registrados.

4.4.2.3 UsersController: Función add

Carga la vista y valida los datos introducidos por el usuario. En caso de validación correcta guarda el nuevo usuario en la base de datos y en caso de que haya errores en el formulario o cualquier otro problema al ejecutar el guardado muestra un mensaje de error.

4.4.2.4 UsersController: Función delete

La función de eliminación recibe un identificador por parámetro y, tras comprobar su existencia, lo borra de la base de datos.

4.4.2.5 UsersController: Función index

Realiza una búsqueda de todos los usuarios en la base de datos y le envía la información a la vista para mostrársela al usuario.

4.4.2.6 UsersController: Función beforeFilter

Función de gestión de accesos. En este caso permite a todos los usuarios con sesión activa el acceso a las funciones de login y logout. El usuario “Admin” podrá acceder a todas las páginas por defecto, tal y como se ha explicado anteriormente en el controlador AppController.

4.4.3 ContainersController.php

Este controlador se encarga de la gestión de contenedores. Sus funciones se detallan a continuación pero hay que destacar las que permiten mostrar la lista de contenedores, filtrar, descargar la documentación asociada a cada envío, generar previsiones y ayudar con el análisis de calidad del transporte.

4.4.3.1 ContainersController: Función index

Función base del programa que muestra el listado de envíos y es la primera que se ejecuta tras la conexión de un usuario.

La parte fundamental del código de esta función, se encarga de realizar la búsqueda de todos los contenedores de la tabla “containers” y enviárselo a la vista para que pueda mostrárselo al usuario por pantalla. Será la vista la que se encargue de filtrar por el estado del contenedor y mostrar únicamente los pendientes.

El resto del código de esta función gestiona el menú inferior mediante un listado de condiciones. A continuación, se presenta un ejemplo que gestiona la opción de creación de nuevos contenedores, que tras detectar la pulsación del botón “add”, redirige al usuario a la función “add” del controlador, con su vista correspondiente.

```
// Gestión del menú inferior
// Comprobar valores recibidos de la vista
if ($this->request->is('post')) {
    // Si el usuario hace clic en menu inferior - add
    if(isset($this->request->data['addbutton_x'])){
        // Redireccionar a función add
        $this->redirect(array('action' => 'add'));
    }
}
```

Figura 4.62. Gestión del menú inferior (Controlador)

El resto de opciones del menú inferior se gestionan con condiciones similares a la anterior, motivo por el que no se detallan en esta memoria.

4.4.3.2 ContainersController: Función stored

Es una función análoga a la anterior necesaria para permitir a la vista realizar el filtro por estado y mostrar solamente los contenedores archivados.

4.4.3.3 ContainersController: Función add

Crea el listado de orígenes y destinos disponibles y los envía a la vista. Posteriormente, recoge la información introducida por el usuario y guarda el nuevo contenedor en la base de datos. En caso de error, muestra un mensaje y espera la introducción de nuevos datos.

Una vez añadido el contenedor a la base de datos, redirige al usuario al índice.

4.4.3.4 ContainersController: Función edit

De la misma forma que la función anterior, “edit” crea el listado de orígenes y destinos y, además, carga la información del contenedor recibido por parámetro para enviarla a la vista. Después, recoge la información de pantalla y modifica la base de datos de acuerdo a ésta para finalmente redirigir al usuario a la página con el listado de contenedores. En caso de error, muestra un mensaje por pantalla y espera nuevos datos.

4.4.3.5 ContainersController: Función delete

Recibe un número de contenedor como parámetro y, tras comprobar su existencia, lo borra de la base de datos y actualiza el estado de las órdenes asociadas a él.

4.4.3.6 ContainersController: Función filter

Permite obtener de la vista, los datos para realizar el filtro. Para ello primero crea una lista con los orígenes y destinos disponibles y se los envía a la vista. Después, espera a que el usuario introduzca los criterios y le redirige a la función que realiza el filtro pasando los criterios como parámetro.

4.4.3.7 ContainersController: Función filteretd

Esta función es igual a la anterior pero capaz de filtrar por rango de fechas estimadas de salida.

4.4.3.8 ContainersController: Función filtereta

Función análoga anterior pero para fechas de llegada.

4.4.3.9 ContainersController: Función findex

Esta función es similar a “index”. Sin embargo, ésta realiza el listado de contenedores teniendo en cuenta las restricciones que recibe por parámetro.

4.4.3.10 ContainersController: Función descargar

Gracias a esta función el usuario puede obtener en PDF los documentos básicos de importación: factura y lista de carga.

La codificación del controlador es muy sencilla gracias a la utilización del “plugin” CakePDF para CakePHP. Simplemente se realiza la búsqueda de los datos y se envían a la vista.

4.4.3.11 ContainersController: Función forecastrange

Es la función de configuración del rango de fechas para las que generar las previsiones.

4.4.3.12 ContainersController: Función forecast

Es una función que, pese a su alto grado de sencillez, servirá de gran ayuda para planificar la plantilla en los almacenes de recepción ya que genera un listado de los envíos que se prevén entregar entre las fechas seleccionadas en la función anterior y un resumen de las cantidades, peso y volumen totales.

La función realiza una búsqueda de todos los contenedores con fecha prevista de llegada entre las recibidas por parámetro y lo envía a la vista, que se encarga de realizar los cálculos totales y mostrárselo al usuario.

4.4.3.13 ContainersController: Función updatestatus

Esta es una función que recibe por parámetro el “id” y el nuevo “status” de un contenedor y lo actualiza en la base de datos, modificando también el estado de las órdenes asociadas a él.

4.4.3.14 ContainersController: Función updatetransport

Es la función que actualiza un contenedor recibido por parámetro con la información relativa al transporte que recibe de la misma manera.

4.4.3.15 ContainersController: Función updateata

Esta función permite actualizar la fecha real de llegada.

4.4.3.16 ContainersController: Función desviationsrange

“Desviationsrange” permite al usuario seleccionar los criterios para el análisis de transporte. Primero crea y envía a la vista los listados de orígenes, destinos y transportistas disponibles y espera a que el usuario seleccione los que le interesan. Al validar el formulario, la función envía sus valores a “desviations” por parámetro.

4.4.3.17 ContainersController: Función desviations

La función “desviations” permite al usuario analizar las diferencias entre las fechas previstas y reales de llegada, es decir, analizar la calidad del transporte. Para conseguirlo, realiza una búsqueda de los contenedores que cumplen las restricciones que recibe por parámetro y envía el resultado a la vista, que será la encargada de mostrarlos tal y como se ha explicado en el punto 4.3.2.14.

4.4.3.18 ContainersController: Función cddontimerange

Es la función de selección de criterios para el análisis de las entregas fuera de plazo. Tiene los mismos campos y funcionamiento que la función “desviationsrange” pero envía los valores a la función “cddontime”, explicada a continuación.

4.4.3.19 ContainersController: Función cddontime

Gracias a esta función, el usuario puede saber qué órdenes se entregarán fuera de plazo y el resumen total para un rango de fechas seleccionado en la función anterior. Para ello, realiza una búsqueda de los contenedores que cumplen las restricciones que recibe por parámetro y envía el resultado a la vista, que será la encargada de mostrarlos tal y como se ha explicado en el punto 4.3.2.16.

4.4.3.20 ContainersController: Función beforeFilter

Esta función gestiona el acceso según el rol de usuario. En la imagen incluida a continuación se aprecia el listado de acciones permitidas para cada rol de usuario.

```
// Da permisos según el rol del usuario
// Se carga automáticamente al cargar el controlador
public function beforeFilter() {
    parent::beforeFilter();
    $usuario = $this->Session->read('Auth.User');
    // Gestión de accesos para export
    if ($usuario['role'] == 'export'){
        $this->Auth->allow('add', 'edit', 'index', 'delete', 'updatestatus',
            'descargar', 'updatetransport', 'findex', 'filter', 'filtereta', 'filteretd');
    }
    // Gestión de accesos para import
    if ($usuario['role'] == 'import'){
        $this->Auth->allow('index', 'updatestatus', 'updateata', 'descargar',
            'findex', 'filter', 'filtereta', 'filteretd', 'stored', 'forecastrange',
            'forecast', 'desviationsrange', 'desviations', 'cddontimerange', 'cddontime');
    }
}
```

Figura 4.63. ContainersController – Función beforefilter

Cabe recordar que el rol “Admin” tiene acceso total gracias a la función de “AppController”.

4.4.4 ContainerLinesController.php

“ContainerLinesController” contiene las funciones que relacionan pedidos y contenedores.

4.4.4.1 ContainerLinesController: Función view

Mediante esta función se crea el listado de órdenes asociadas a un contenedor y se gestiona también el caso extremo en que el envío no tiene ninguna orden asociada todavía. En este caso, muestra un aviso y redirige al usuario hacia la función “add” y su correspondiente vista.

4.4.4.2 ContainerLinesController: Función add

Es la función que permite asociar órdenes a un contenedor. Primero crea un listado de las órdenes compatibles y las envía a la vista para que sean mostradas en un menú desplegable.

Después, espera la confirmación del formulario, crea la nueva línea en el contenedor y actualiza los datos de las órdenes asociadas mostrando un mensaje de confirmación y redirigiendo a la vista de detalles del contenedor. En caso de error, se muestra un aviso por pantalla.

4.4.4.3 ContainerLinesController: Función delete

La función de borrado elimina la línea y actualiza las órdenes asociadas liberándolas para que puedan formar parte de otro contenedor. Después muestra el mensaje de confirmación y redirige al usuario hacia la vista de detalles del contenedor.

4.4.4.4 ContainerLinesController: Función beforefilter

Esta función gestiona el acceso según el rol de usuario. En este caso el usuario “Export” tendrá acceso a las tres funciones descritas anteriormente mientras que los usuarios de rol “Import”, sólo podrán acceder a “view”.

4.4.5 OrdersController.php

Este controlador gestiona los pedidos del sistema. A continuación se describen sus funciones sin mostrar código ya que éste es muy similar a los descritos anteriormente.

4.4.5.1 OrdersController: Función index

Es una función muy similar a la que lleva su mismo nombre en el resto de controladores. Se encarga de crear el listado de las órdenes del sistema y enviárselo a la vista para que sea mostrado al usuario. Además, gestiona el menú inferior de la misma manera que la función del controlador “containers” pero con menos opciones activas.

4.4.5.2 OrdersController: Función add

La función “add” permite introducir nuevas órdenes en el sistema. Primero crea un listado de los orígenes y destinos disponibles y los envía a la vista para que el usuario pueda seleccionarlos de una lista desplegable. Una vez validado el formulario, se encarga de crear la nueva orden en la base de datos y redirigir al usuario hacia la función “index”.

Esta función muestra un mensaje al usuario para confirmar la correcta creación de la orden o avisar de cualquier error que se haya producido.

4.4.5.3 OrdersController: Función edit

De la misma manera que la función anterior, crea una lista de orígenes y destinos disponibles que, junto con la información que obtiene sobre la orden que le llega por parámetro, es enviada a la vista para ser mostrada al usuario. Una vez éste ha validado el formulario, la función recoge los datos y actualiza los registros de la base de datos.

4.4.5.4 OrdersController: Función delete

Borra la orden de la base de datos y muestra el mensaje de confirmación.

4.4.5.5 OrdersController: Función descargar

De la misma forma que la función “descargar” del controlador “Containers”, ésta genera un documento PDF con los documentos de importación. La diferencia, en este caso, es que únicamente genera documentos para la orden seleccionada mientras que en el caso del otro controlador se generan para todas las órdenes asociadas al contenedor de referencia.

4.4.5.6 OrdersController: Función beforefilter

Es la función de gestión de acceso según el rol de usuario que permite el acceso a las funciones “index” y “descargar” al rol “Import” y acceso a todas las funciones al rol “Export”.

4.4.6 OrderLinesController.php

Este controlador gestiona las líneas de los pedidos, es decir, los artículos que forman cada orden así como sus cantidades. Sus funciones son muy similares a las del controlador “ContainerLinesController”.

4.4.6.1 OrderLinesController: Función view

Mediante esta función se crea el listado de artículos asociados a una orden y se gestiona también el caso extremo en que el pedido no tiene ningún artículo todavía. En este caso, muestra un aviso y redirige al usuario hacia la función “add” y su correspondiente vista.

4.4.6.2 OrderLinesController: Función add

La función descrita en este apartado permite añadir artículos a un pedido. Primero crea un listado con todos los artículos del sistema y los envía a la vista para que el usuario pueda seleccionarlos de un menú desplegable. Después espera los datos del formulario y crea la nueva línea en el pedido. Para finalizar, muestra un mensaje de confirmación y redirige al usuario hacia la vista de detalles de la orden. En caso de error, muestra un aviso por pantalla.

4.4.6.3 OrderLinesController: Función edit

Es la función que permite editar las líneas de un pedido. Primero crea un listado con todos los artículos del sistema y los envía a la vista junto con la cantidad actual de esa línea para que el usuario pueda modificarlos. Después espera los datos del formulario y modifica el pedido en la base de datos. Para finalizar, muestra un mensaje de confirmación y redirige al usuario hacia la vista de detalles de la orden. En caso de error, muestra un aviso por pantalla.

4.4.6.4 OrderLinesController: Función delete

La función de borrado elimina el artículo, que recibe por parámetro, de la orden. Después muestra el mensaje de confirmación y redirige al usuario hacia la vista de detalles.

4.4.6.5 OrderLinesController: Función beforefilter

La función “beforefilter” gestiona el acceso según el rol de usuario. En este caso, el usuario “Export” tendrá acceso a todas las funciones descritas anteriormente mientras que los usuarios del rol “Import”, sólo podrán acceder a “view”.

4.4.7 ItemsController.php

Este controlador se dedica a la gestión de los artículos del sistema. Permite listar, crear, editar, borrar e imprimir el listado de artículos.

4.4.7.1 ItemsController: Función index

Mediante esta función se crea el listado de artículos del sistema que se envía a la vista para mostrarse por pantalla, además de gestionarse el menú inferior de la misma manera que en los controladores anteriores.

4.4.7.2 ItemsController: Función add

Esta función permite añadir artículos a la base de datos. La función espera los datos del formulario y los guarda en la base de datos mostrando un mensaje de confirmación y redirigiendo al usuario hacia la lista de artículos. En caso de error, se muestra un aviso.

4.4.7.3 ItemsController: Función edit

Es la función que permite modificar la información de los artículos. Para ello carga la información de la base de datos y la envía a la vista. Después, espera la validación del formulario y guarda los cambios.

4.4.7.4 ItemsController: Función delete

La función de borrado elimina el artículo que recibe por parámetro. Después muestra el mensaje de confirmación y redirige al usuario hacia la lista. Esta función se ha codificado pero no está disponible para ningún tipo de usuario a fin de garantizar la integridad de los datos. El motivo de este diseño es que eliminar un artículo en uso afecta a las órdenes que lo contienen y en muchos casos, no es posible editar estas órdenes porque ya han sido enviadas.

4.4.7.5 ItemsController: Función descargar

Mediante esta función se crea un archivo PDF con la información de la lista de artículos. Su funcionamiento es igual al de las funciones con el mismo nombre en otros controladores.

4.4.7.6 ItemsController: Función beforefilter

Permite el acceso al rol “Export” a todas las funciones del controlador mientras que al rol “Import” solamente le permite ver la lista de artículos e imprimirla.

4.4.8 OriginsController.php

Este controlador se encarga de la creación, modificación, eliminación y listado de orígenes.

4.4.8.1 OriginsController: Función index

La función “index” envía la lista de orígenes almacenados en el sistema a la vista y gestiona el menú inferior.

4.4.8.2 OriginsController: Función add

Esta función permite la creación de nuevos orígenes. Para ello, valida la información recibida de la vista y la guarda en la base de datos. Después, muestra un mensaje de confirmación o error y redirige al usuario hacia la lista de orígenes.

4.4.8.3 OriginsController: Función edit

La función permite la modificación de orígenes existentes. Para conseguirlo, carga la información del origen que recibe por parámetro y la manda a la vista para que el usuario pueda modificarla. Posteriormente, valida la información recibida de la vista y la guarda en la base de datos, muestra un mensaje de confirmación o error y redirige al usuario hacia la lista de orígenes.

4.4.8.4 OriginsController: Función delete

Mediante esta función, se borra el origen que recibe por parámetro de la base de datos y el usuario es redirigido hacia la función de lista de orígenes.

4.4.8.5 OriginsController: Función beforefilter

Esta función permite el acceso al rol “Export” a todas las funciones del controlador mientras que al rol “Import” sólo se le permite ver la lista de orígenes.

4.4.9 DestinationsController.php

Controlador análogo al anterior que gestiona destinos en lugar de orígenes.

4.4.9.1 DestinationsController: Función index

Esta función envía la lista de destinos almacenados en el sistema a la vista y gestiona el menú inferior.

4.4.9.2 DestinationsController: Función add

La función “add” permite la creación de nuevos destinos. Para esto, valida la información recibida de la vista y la guarda en la base de datos. Después, muestra un mensaje de confirmación o error y redirige al usuario hacia la lista de destinos.

4.4.9.3 DestinationsController: Función edit

Esta función permite la modificación de destinos existentes. Se carga la información del destino que recibe por parámetro y la manda a la vista para que el usuario pueda modificarla. Posteriormente valida la información recibida y la guarda en la base de datos, muestra un mensaje de confirmación o error y redirige al usuario hacia la lista.

4.4.9.4 DestinationsController: Función delete

Mediante esta función se borra el destino que recibe por parámetro de la base de datos y al usuario se le redirige hacia la función de lista de destinos.

4.4.9.5 DestinationsController: Función beforefilter

Es la función que permite el acceso al rol “Import” a todas las funciones del controlador mientras que al rol “Export” solamente le permite ver la lista de destinos.

4.4.10 TransportsController.php

Este módulo se encarga de la gestión de los transportes. Permite listar, crear, modificar y borrar los transportes de la base de datos. Además, contiene una función que muestra los transportes compatibles con cada contenedor y añade información acerca del día de llegada, precios y demás información de importancia sobre el transporte.

4.4.10.1 TransportsController: Función index

Como en el resto de controladores, la función index envía a la vista una lista de los transportes de la base de datos y gestiona el menú inferior.

4.4.10.2 TransportsController: Función add

Esta función permite añadir nuevos transportes a la base de datos. Para hacerlo, envía las listas de orígenes y destinos disponibles a la vista y espera la validación del formulario para guardar la información en la base de datos. Finalmente, muestra un mensaje de confirmación o error y redirige al usuario hacia la lista de transportes.

4.4.10.3 TransportsController: Función edit

Permite la modificación de un transporte existente que recibe por parámetro cargando su información y mandándola a la vista para que el usuario pueda modificarla. Posteriormente valida la información recibida y la guarda en la base de datos, muestra un mensaje de confirmación o error y redirige al usuario de nuevo a la lista.

4.4.10.4 TransportsController: Función delete

Borra el transporte de la base de datos y redirige hacia la lista de transportes.

4.4.10.5 TransportsController: Función Budget

Esta función es de gran utilidad ya que simplifica la asignación de transporte a cada contenedor mediante una lista de transportes compatibles con sus precios y fecha de llegada prevista según el servicio. Además, calcula el tipo de contenedor más apropiado según el volumen total para que la tasa de llenado sea lo más alta posible.

4.4.10.6 TransportsController: Función beforefilter

En este caso, la función de gestión de acceso permite a los usuarios de tipo “Export” el acceso a todas las páginas mientras que a los usuarios “Import” sólo les permite consultar la lista de transportes.

Capítulo V.

Pruebas y resultados

5.1 Introducción

En este capítulo se describen las pruebas realizadas a fin de garantizar la solidez del sistema. Debido a la metodología de desarrollo utilizada en la que se iban realizando pruebas al finalizar la codificación de cada módulo, éstas se dividen en pruebas de unidad, pruebas de integración y pruebas de sistema. Las dos primeras, se realizaron de forma repetitiva para cada módulo mientras que la tercera se realizó al finalizar el programa.

A cada uno de los tipos anteriores se les aplicó la estrategia de pruebas de caja blanca y caja negra intentando así garantizar al máximo posible la calidad del software.

Las pruebas de caja blanca son aquellas que intentan garantizar la correcta implementación de la función, ejecutando, al menos una vez, todas las líneas de código, llevando los bucles a los casos extremos y probando todos los posibles caminos lógicos.

Por otro lado, las pruebas de caja negra se centran en el funcionamiento del software sin tener en cuenta su lógica interna. Para ello se consideran únicamente las entradas y salidas del sistema y se comprueba su corrección.

A continuación se detallan las pruebas y se muestran ejemplos de los resultados obtenidos.

5.2 Pruebas de unidad

En este punto se incluyen las pruebas realizadas independientemente en cada módulo.

Dado el gran número de pruebas de este tipo realizadas sólo se incluyen algunos ejemplos, utilizando como base el módulo “containers” debido a que es el más representativo.

5.2.1 Pruebas de caja blanca

Para las pruebas de unidad de caja blanca se ejecutaron todas las funciones con valores válidos, inválidos, vacíos, repetidos, extremos y reales. Tras encontrar algún error, éste fue corregido al momento y se volvieron a realizar todas las pruebas desde el principio.

5.2.1.1 Ejemplos de pruebas de caja blanca

- Creación, modificación y filtrado de contenedores con valores correctos, incorrectos, existentes, cero, negativos, extremadamente altos o nulos.
- Eliminación de contenedores.
- Impresión de contenedores.
- Modificación de fechas.
- Archivo de contenedores.
- Comprobación de todas las opciones del menú inferior y específicas del módulo.

Para garantizar la calidad de las pruebas deben realizarse para los diferentes estados del contenedor y desde los diferentes roles de usuario.

5.2.2 Pruebas de caja negra

En la fase de test de caja negra se ejecutaron todas las funciones de todos los módulos con varias entradas y se observó si la salida era la esperada o no.

5.2.2.1 Ejemplos de pruebas de caja negra y sus resultados

- Creación y modificación de un contenedor con campos válidos.
El contenedor se guarda correctamente en la base de datos.
- Creación y modificación de un contenedor con campos inválidos.
El sistema no valida el campo y paraliza la acción mostrando un mensaje de error.
- Eliminar un contenedor.
El sistema borra el contenedor de la base de datos.
- Imprimir un contenedor.
El sistema crea un archivo PDF con las listas de carga y las facturas correctamente.
- Enviar un contenedor.
El sistema carga y muestra la página de presupuesto de transportes compatibles.
- Archivar un contenedor.
El sistema cambia el estado del contenedor correctamente y no se visualiza en la vista "index", pero sí en la vista "stored".

Para garantizar la calidad de las pruebas, se realizaron para los diferentes estados del contenedor y desde los diferentes roles de usuario.

5.3 Pruebas de integración

En este punto se detallan las pruebas sobre las funciones que acceden a varios módulos.

Para documentar este tipo de pruebas se detallan las más significativas y se da una explicación general del resto.

5.3.1 Pruebas de caja blanca

En esta parte de las pruebas, se intentó ejecutar todo el código que permite la interacción entre los diferentes módulos.

5.3.1.1 Ejemplos de pruebas de caja blanca

- Asignación de una orden a un contenedor.
- Eliminación de contenedores.
- Cambios de estado de contenedores.
- Asignación de transporte a un contenedor.
- Modificación de un transporte.
- Comprobación de todas las opciones de los menús.

5.3.2 Pruebas de caja negra

En esta fase se ejecutaron las funciones con interacción entre módulos proporcionándoles diferentes entradas y analizando las salidas tras la ejecución.

5.3.2.1 Ejemplos de pruebas de caja negra y sus resultados

- Asignación de una orden a un contenedor.
Se crea la línea del contenedor en la tabla "containerlines" de la base de datos.
Se modifica el estado de la orden en la base de datos.
Se modifica el campo "container_id" de la orden en la base de datos.

- Eliminación de contenedores

Se borra la línea del contenedor en la tabla “containers” de la base de datos.

Se borran las líneas del contenedor en la tabla “containerlines” de la base de datos.

Se modifica el estado de las órdenes asociadas en la base de datos.

Se borra el campo “container_id” de las órdenes asociadas en la base de datos.

- Asignación de transporte a un contenedor

Se actualiza la tabla “containers” de la base de datos.

Se modifica el estado de las órdenes asociadas en la base de datos.

- Modificación de un transporte.

Se actualiza la tabla “transports” de la base de datos.

No se modifica nada en los contenedores con ese transporte asociado.

5.4 Pruebas de sistema

En este punto se prueba el sistema completo intentando simular el trabajo en la vida real.

5.4.1 Pruebas de caja blanca

En esta parte de las pruebas, se intentó ejecutar todo el código del programa.

5.4.1.1 Ejemplos de pruebas de caja blanca

- Creación, login, logout y eliminación de usuarios.
- Creación, modificación, eliminación, listado e impresión de artículos.
- Creación, modificación, eliminación, listado e impresión de órdenes.
- Creación, modificación, eliminación y listado de líneas de pedido.
- Creación, modificación, eliminación y listado de líneas de contenedor.
- Creación, modificación, eliminación, listado e impresión de contenedores.
- Creación, modificación, eliminación y listado de orígenes.
- Creación, modificación, eliminación y listado de destinos.
- Creación, modificación, eliminación y listado de transportes.
- Filtro, archivo y actualización de fechas de contenedores.
- Creación de previsiones.
- Creación de presupuestos de transportes.
- Envío de contenedores.
- Análisis de CDD y transportes.
- Comprobación de todas las opciones de los menús.

5.4.2 Pruebas de caja negra

En esta fase se ejecutaron todas las funciones del programa simulando el trabajo en la vida real y analizando las salidas tras la ejecución de cada una de ellas y en conjunto.

5.4.2.1 Ejemplos de pruebas de caja negra y sus resultados

- Creación de un usuario

Se crea el usuario en la base de datos.

- Login de usuario

Se abre la sesión para ese usuario.

- Creación de varios artículos

Se crean los artículos en la base de datos.

- Creación de varias órdenes
 - Se crean las órdenes en la base de datos.*
- Asignación de artículos a las órdenes
 - Se crean las líneas de las órdenes en la base de datos.*
- Creación de varios contenedores
 - Se crean los contenedores en la base de datos.*
- Asignación de órdenes a los contenedores
 - Se crean las líneas de los contenedores en la base de datos.*
 - Se modifican los estados de las órdenes asociadas y su "container_id".*
- Asignación de transporte a los contenedores.
 - Se actualiza la tabla "containers" en la base de datos.*
 - Se modifican los estados de las órdenes asociadas.*
- Archivo de un contenedor.
 - Se modifica el estado del contenedor en la base de datos.*
 - Se modifican los estados de las órdenes asociadas en la base de datos.*

5.5 Comentarios

Gracias a la realización de las pruebas descritas en este capítulo, principalmente las pruebas unitarias al ser las primeras en realizarse, se encontraron y corrigieron multitud de errores.

No obstante, es imposible garantizar al 100% la calidad del software ya que siempre cabe la posibilidad de no haber explorado alguno de los caminos lógicos.

Capítulo VI.

Conclusiones y posibles ampliaciones

6.1 Conclusiones

Tras la finalización del proyecto se observa que se han conseguido los objetivos del proyecto. Se ha provisto al usuario de una aplicación capaz de gestionar las importaciones en tiempo real con todas las herramientas requeridas por el objetivo principal.

Además, se ha creado una funcionalidad para generar previsiones asumiendo así el más importante de los objetivos secundarios y varias funcionalidades más que facilitarán el trabajo de los usuarios, como por ejemplo, impresión de documentos en PDF o presupuestos de transporte.

Para el último de los objetivos secundarios se han desarrollado las funcionalidades de análisis de transporte y de CDD, que permiten detectar y mostrar al usuario las diferencias entre fechas estimadas y reales de llegada, visualizar el transporte realizado en un rango de fechas y listar las órdenes que llegaron con retraso respecto a la fecha contractual.

Por todo ello, los objetivos se han asumido y se han añadido funcionalidades que aportan valor añadido al programa. No obstante, el sistema es todavía muy simple y sería interesante realizar las ampliaciones detalladas en el punto siguiente.

6.2 Posibles ampliaciones

6.2.1 Gestión de usuarios por centro y rol

Al ser ésta la primera versión del sistema y ser un proyecto final de carrera, se ha realizado la gestión de usuarios teniendo en cuenta únicamente sus roles. Una posible ampliación sería añadir un centro a cada usuario y permitirle acceso únicamente a los datos relacionados con ese centro.

Por ejemplo, un usuario con rol “Import” podría tener el centro de importación “Barcelona” asociado y por lo tanto sólo tendría acceso a contenedores y órdenes con destino Barcelona. Por otro lado, un usuario “Export” podría tener asociado el centro de exportación “Bangkok” y únicamente podría crear órdenes, contenedores y transportes con ese origen.

En este caso solamente los usuarios “Admin” tendrían acceso a la información global del sistema y serían capaces de gestionar orígenes y destinos.

A nivel técnico esta evolución no sería muy complicada de llevar a cabo gracias al desarrollo modular del aplicativo. Bastaría con añadir un nuevo campo al formulario de registro de nuevos usuarios, modificar la tabla users de la base de datos para almacenar este nuevo campo y añadir la restricción de centro mediante una condición “if” en las vistas.

6.2.2 Ampliación de la información

Por el mismo motivo que en el punto anterior, la información de las tablas es suficiente para gestionar las importaciones pero resulta muy básica si se traslada el proyecto a la vida real.

Una gran mejora para el proyecto sería incluir más datos en todas las tablas, como por ejemplo, más información relacionada con los artículos, datos sobre el país de fabricación y el fabricante y no sólo del centro de exportación, etc.

A nivel técnico esta evolución sería sencilla pero más laboriosa que la anterior ya que supondría añadir columnas a las tablas de la base de datos y campos a los formularios afectados por las nuevas columnas.

6.2.3 Mejorar rendimiento de la aplicación web

Al tratarse de páginas sin demasiado contenido no se ha desarrollado la carga modular de las mismas a excepción de la cabecera y el pie de página. Esto ocasiona que el menú se cargue siempre con el contenido y por lo tanto reduce el rendimiento de la aplicación.

Inicialmente se diseñaron menús diferentes para cada módulo por lo que esta opción quedó descartada. Sin embargo, a lo largo del desarrollo del sistema se modificó este diseño para hacer un menú de navegación común a todos los módulos.

A nivel técnico esta solución es algo compleja porque todos los módulos están afectados y deberían ser editados a pesar de que sería una gran mejora para el proyecto de cara a futuras evoluciones y ampliaciones ya que bastaría con modificar un menú en vez de codificar la misma modificación en todas las vistas.

6.2.4 Selección de múltiples líneas

Actualmente el sistema sólo es capaz de tratar un “checkbox” activo por lo que en alguna ocasión puede generar tareas repetitivas para el usuario. Para evitar esto, sería interesante ampliar la codificación y permitir al usuario seleccionar varias líneas para realizar la misma tarea, en todas ellas, mediante un único paso.

6.2.5 Introducción de transporte aéreo

A día de hoy son muchas las empresas que utilizan el medio aéreo para transportar la mercancía más urgente debido a que, por cuestiones logísticas y de marketing entre otras, no pueden esperar el tiempo de tránsito marítimo. En este punto, el programa no ofrece soluciones al usuario por lo que será un tema importante en las próximas evoluciones.

A nivel técnico el programa ya cuenta con parte de la codificación necesaria en el controlador “Transports” que necesita ser complementada tanto en este controlador como en el controlador “Containers”.

Capítulo VII.

Bibliografía y

referencias

7.1 Bibliografía

- Desarrollo y gestión de proyectos informáticos:
Escritor: S. McConnell, editorial: McGraw Hill, año 2000.
- Fundamentos de bases de datos:
Escritores: A. Silberschatz, H.F. Korth, S. Sudarshan, editorial: McGraw Hill, año 2006.
- PHP 5, Manual imprescindible:
Escritor: L.M. Cabezas Granado, editorial: Ediciones Anaya Multimedia, año 2004.
- Desarrollo Web con PHP y MySQL:
Escritor: J.A Gallego Vázquez, editorial: Ediciones Anaya Multimedia, año 2003.

7.2 Referencias

- Wikipedia: Información de todos los puntos del proyecto.
<http://www.wikipedia.org/>
- W3schools: Información sobre páginas web y lenguajes de programación.
<http://www.w3schools.com/>
- Web oficial de CakePHP: Información y archivos de CakePHP.
<http://www.cakephp.org>
- API oficial de CakePHP: Documentación de los detalles internos del “framework” CakePHP.
<http://api.cakephp.org>
- Cookbook de CakePHP: Información y ejemplos de CakePHP.
<http://book.cakephp.org>
<http://book.cakephp.org/es>
- Manual de CakePHP: Información y ejemplos de CakePHP.
Documento PDF disponible en el CDROM adjunto a esta memoria.
- Bakery de CakePHP: Almacén de recursos y ejemplos relacionados con CakePHP.
<http://bakery.cakephp.org>
- Google Groups: Preguntas y respuestas sobre CakePHP.
<http://groups.google.com/group/cake-php/>
<http://groups.google.com/group/cakephp-esp/>
- Github: Proyectos CakePHP y “plugins” compatibles compartidos.
<https://github.com>
<https://github.com/coisa/CakePDF>
- Web oficial de TCPDF: Información, ejemplos y archivos de TCPDF.
<http://www.tcpdf.org/>
<http://www.tcpdf.org/examples.php>

7.3 Otros recursos

Además de los anteriormente mencionados se han utilizado recursos proporcionados por la empresa en la que trabaja el autor del proyecto sobre importación-exportación, transporte nacional e internacional, “incoterms”, planificación de actividad, gestión de plantilla, control de gastos, gestión de proyectos y pruebas.

