



Universidad de las Ciencias Informáticas

Facultad 4

**Módulos para la aplicación móvil del Sistema de Gestión Integral de la
empresa TRANSCARGO**

**Trabajo de diploma para optar por el título de Ingeniero
en
Ciencias Informáticas.**

Autor: Randy Muñoz Nuñez

Tutores: Ing. Celia Indira Hidalgo Tagle

Ing. Oreste Abraham Nillar Cambara

Ing. Orlando Grabiél Toledano López

<

“Año 60 de la Revolución”

DECLARACIÓN DE AUTOÍRIA

Declaramos que somos los únicos autores del trabajo “**Módulos para aplicación móvil del Sistema de Gestión Integral SIGAX de la empresa TRANSCARGO**” y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Randy Muñoz Nuñez

Autor

Ing. Celia Indira Hidalgo Tagle

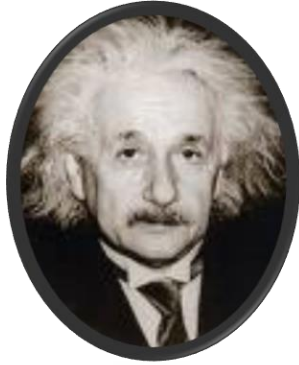
Tutor

Ing. Orlando Grabiél Toledano López

Tutor

Ing. Oreste Nillar Cambara

Tutor



“Una universidad es un lugar donde la universalidad del espíritu humano se manifiesta”.

Albert Einstein.

Dedicatoria

Dedico el presente trabajo de diploma a mis padres por su amor, confianza y apoyo incondicional, a mi abuela Estrella que donde quiera que esté espero que esté orgullosa, a mis hermanos que junto a mis padres son lo que más quiero en este mundo.

Agradecimientos

A mi madre y mi madre, son tantas cosas por lo que agradecerles que no las puedo resumir en unas oraciones, pero tengan claro, que todo lo que soy es gracias a ustedes.

A mis amistades que me apoyaron en cada momento que lo necesité.

A mis tutores, que me han ayudado tanto en todo el proceso, gracias por brindarme su apoyo.

A mi oponente Sandy Guerra Fernández, por poner su granito de arena en la terminación de este proyecto.

RESUMEN

Dentro de las Tecnologías de la Información y las Comunicaciones se encuentran los Sistemas de Gestión Integral, los cuales se han convertido en un pilar fundamental para muchas empresas, permitiendo componer y automatizar la mayoría de sus procesos. Una de las preferencias en los últimos años ha sido la adaptación de estos sistemas al entorno móvil, lo cual lleva a las empresas a sus máximas capacidades y ganancias. En la actualidad el Centro de Tecnologías para la Formación, el cual pertenece a la Universidad de las Ciencias Informáticas se encuentra desarrollando un Sistema de Gestión Integral con el nombre comercial TRANSCARGO. El sistema, actualmente en fase de desarrollo, no ha sido adaptadas la mayoría de sus funcionalidades para terminales móviles. Lo cual provoca una falta de movilidad en la empresa y la imposibilidad de acceder a este si no existe conexión de red. La presente investigación describe el desarrollo de una aplicación nativa que proveerá soluciones a los anteriores problemas. La aplicación fue desarrollada mediante la metodología AUP-UCI, utilizando como entorno de desarrollo Android Studio, el lenguaje utilizado para el desarrollo fue Java, integrado con el *framework* Odo Mobile. Con el desarrollo de la aplicación se posibilita acceder de forma personalizada a la información de los módulos del sistema SIGAX, logrando a la vez el almacenamiento de los datos en la base de datos del dispositivo, lo cual permite la futura consulta de los datos sin necesidad de tener conectado el dispositivo en alguna red.

Índice

DECLARACIÓN DE AUTOÍRIA.....	II
RESUMEN.....	VI
Índice.....	VII
Introducción.....	1
CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Conceptos asociados al dominio del problema.....	6
1.1.1 Dispositivos móviles.....	6
1.1.2 Comunicación inalámbrica.....	7
1.1.3 Aplicación móvil.....	8
1.1.4 ERP.....	8
1.2 Estudio de soluciones similares.....	9
1.2.1 Aplicaciones ERP móviles.....	9
1.2.2 Aplicaciones móviles para Odoo.....	11
1.2.3 Resultados obtenidos sobre el análisis de soluciones similares.....	13
1.3 Tipos de aplicaciones móviles.....	13
1.3.1 Aplicaciones Nativas.....	13
1.3.2 Aplicaciones Web.....	14
1.3.3 Aplicaciones Híbridas.....	14
1.3.4 Selección del tipo de aplicación a utilizar.....	16
1.4 Análisis sobre los sistemas operativos móviles.....	16
1.4.1 Sistema operativo Android.....	16
1.4.2 Sistema operativo iOS.....	18
1.4.3 Sistema operativo Windows Phone.....	18
1.4.4 Selección del sistema operativo a utilizar.....	18
1.5 Metodología de desarrollo.....	19
1.5.1 Metodologías ágiles.....	19
1.5.2 Metodología de desarrollo de software RUP.....	20

1.5.3	Metodología de desarrollo AUP	20
1.6	Herramienta y tecnologías a utilizar.	22
1.6.1	Lenguaje Unificado de Modelado.....	22
1.6.2	Lenguaje de programación a utilizar	23
1.6.3	Entorno de desarrollo integrado.....	23
1.6.4	Framework de desarrollo	25
1.7	Conclusiones parciales	27
CAPÍTULO II. ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA.....		28
2.1	Propuesta de solución	28
2.2	Modelo de dominio	28
2.3	Requisitos funcionales	29
2.4	Requisitos no funcionales	32
2.5	Historias de usuarios	32
2.6	Modelo de datos	34
2.7	Patrón Arquitectónico.....	36
2.8	Patrones de diseños	37
2.8.1	Patrones GRAPS.....	37
2.8.2	Patrones GoF	38
2.8.3	Diagrama de clases	39
2.9	Conclusiones parciales	39
CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA.....		41
3.1	Modelo de implementación	41
3.1.1	Pautas de codificación	41
3.1.2	Diagrama de Componentes	42
3.2	Pruebas y validación del software.....	43
3.2.1	Pruebas funcionales	43
3.2.2	Resultados de las pruebas funcionales.....	44
3.2.3	Pruebas de aceptación	45

3.2.4	Resultados de las pruebas de aceptación.....	46
3.2.5	Pruebas unitarias.....	47
3.3	Conclusiones parciales.....	49
	Conclusiones.....	50
	Recomendaciones.....	51
	Referencias Bibliográficas.....	52
	Glosario de términos.....	55
	Anexo 1: Historias de usuario.....	56
	Anexo 2: Pruebas de aceptación.....	115
	Anexo 3: Diagramas de clases.....	162

Introducción

Conforme pasa el tiempo, el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) han permitido a la sociedad poder participar en los mercados mundiales creando así oportunidades de desarrollo local, haciendo uso de servicios que prometen la comunicación y transmisión de la información, entre lugares alejados físicamente y de una manera rápida.

En el mundo empresarial, las TIC juegan un papel fundamental debido a que son herramientas que posibilitan la captura de manera rápida y segura de los datos. Con el objetivo de procesar dichos datos, se facilita la información que se pueda manejar en la empresa, obteniendo así un mejoramiento en la optimización de los procesos, agilizando operaciones y actividades empresariales. Esto provee a la empresa de un incremento en el ámbito de la productividad, debido a la mejora de los procesos productivos, dando paso a nuevos y altos niveles de competitividad, por lo cual se generará mayor cantidad de clientes. Todo esto se puede obtener con una buena selección de la tecnología a utilizar. Dentro de los procesos donde se utilizan las TIC en la actualidad están (1):

- El proceso productivo, con los conocidos y famosos ERP (*“Enterprise Resource Planning”* o Sistemas de Gestión Integral).
- Las actividades comerciales y de marketing (con las herramientas enmarcadas dentro del ámbito de la gestión de la relación con los clientes conocidos como CRM (*“Customer Relationship Management”*), modelo de gestión de toda la organización, basada en la satisfacción del cliente).
- Los procesos de desarrollo de nuevos productos, con herramientas de CAD (*“Computer-aided desing”* o Diseño asistido por computadoras). Se refiere al uso del amplio rango de herramientas computacionales que asisten a ingenieros, arquitectos y diseñadores.

Dentro de estas tecnologías, una de las más conocida y utilizada es la ERP. La misma consiste en un sistema integral de gestión empresarial. Este sistema integra totalmente operaciones, informaciones en todas las áreas operativas y administrativas de una empresa. Tiene como propósito otorgar un apoyo a los usuarios y clientes, dándole respuestas rápidas a sus problemas y el manejo eficiente de la información, lo que realmente le genera un valor a la compañía. Al ser un sistema que ofrece una interrelación de todas las áreas de la empresa, elimina costos innecesarios y se logran niveles altos de productividad (2).

Hoy por hoy la sociedad está inmersa en lo que se denomina revolución tecnológica de las comunicaciones inalámbricas, una revolución similar a la que protagonizaron en su momento la televisión, el ordenador o las mismas comunicaciones por cable, que supusieron nuevos modelos de negocio. Al no depender de cable, la movilidad es una de las principales ventajas de esta tecnología, donde la transferencia de información puede ser más rápida que cualquier otro tipo de medio de transmisión (3). Esto trae un efecto de expansión a nivel de productividad debido a la eficiencia de la transferencia de información y comodidad del usuario.

La difusión de la telefonía móvil y sus nuevas aplicaciones digitales implica cambios en la organización cotidiana de la vida privada, lo que a su vez repercute en las actividades laborales tomando en cuenta que ninguna organización funciona como un ente aislado de su contexto. En la actualidad, el mundo de las tecnologías y aplicaciones móviles es tan amplio que se presenta ante la sociedad productiva con una excelente proyección económica, extendiéndose a todos los sectores de producción y consumo en la sociedad. La influencia de dichas aplicaciones aumenta a medida que los efectos de la red lo hacen; es decir, cuando aumenta el número de personas que las utilizan en sus rutinas diarias. Los dispositivos móviles incorporan más y mejores aplicaciones, mientras que la calidad de las comunicaciones también se desarrolla. Dichas aplicaciones también generan innovación, ya que promueven y facilitan la invención de forma combinada con la producción de nuevos servicios, productos o procesos en el ámbito empresarial. A raíz de todo lo antes mencionado, se puede interpretar el impacto que estos cambios han generado en la en el aspecto administrativo de la sociedad, comprendiendo que los beneficios asociados a la difusión de una tecnología de uso general van más allá de su aplicación a los procesos de negocio y permiten generar mejoras en la calidad y eficiencia de los procesos empresariales dentro de las organizaciones. Ya se trate de empresas grandes o pequeñas, desde un punto de vista puramente económico, se puede identificar una serie de áreas en las que la presencia de aplicaciones móviles está impulsando cambios. Ya sea que funcione o no en combinación con la telefonía fija, la comunicación inalámbrica e instantánea permite una mayor flexibilidad de gestión y acelera los procesos que dependen de esta dentro de las organizaciones. (4)

Actualmente las aplicaciones móviles ya no solo se limitan a consultar su correo electrónico. Nuevas aplicaciones permiten acceder al usuario al sistema que se utiliza en su empresa, entre estas está las ERP, con la que se puede conectar y acceder a la aplicación, aunque no esté en su lugar de trabajo. Por lo que le trae beneficios a la empresa tales como (5):

- Aumento de la productividad.
- Reducción de los costos.
- Acelera la toma de decisiones.
- Las ventajas son muchas con respecto a la movilidad, ya que todas las funcionalidades y la información puede ser utilizada y consultada en diferentes áreas y en tiempo real.

La Empresa Transitaria de Cargas Internacional y Nacional, TRANSCARGO, es una entidad con más de 20 años de experiencia en el sector. Cuenta con profesionales especializados, con amplio conocimiento del transporte internacional, que proyectan, coordinan, controlan y dirigen todas las operaciones necesarias

para realizar el transporte y la logística internacional de mercancías. A lo largo de todos estos años TRANSCARGO se ha dedicado a prestar servicios transitarios y de operador logístico internacional para organizar eficientemente la cadena de servicios competitivos, desde el origen hasta la entrega en destino de las mercancías, personalizando los intereses de los clientes. (6)

En la actualidad, el Centro de Tecnologías para la Formación (FORTES) perteneciente a la Universidad de las Ciencias Informáticas (UCI) tiene como objetivo el desarrollo de un Sistema de Gestión Integral adaptado a las necesidades de esta empresa transitaria y logística, llamado “Sistema de Gestión para Actividades Transitarias, Aduanales y Logísticas de Cargas” (SIGAX). El sistema actualmente en desarrollo, aún no ha sido adaptado en su totalidad para su uso en dispositivos móviles, lo cual sería una ventaja de flexibilidad de la empresa y para las actividades de los empleados. No contar con el sistema adaptado al entorno móvil tiene desventajas que puede sufrir una empresa al no poder visualizar el contenido del sistema sin conexión a la red. En este caso se tienen las siguientes limitaciones:

- **Imposibilidad de acceder al sistema si no se tiene conexión a red:** El acceso a la información del negocio es primordial para la gestión de procesos. Al trabajador se le puede dificultar la búsqueda de información en el caso que ocurra algún error en la red, por lo cual el operador no pueda conectarse.
- **Falta de movilidad en la empresa:** No contar con la capacidad de que los operarios fuera de un entorno de oficina donde no llegue señal Wifi, puedan acceder a la información o simplemente desarrollar su trabajo, por lo que supone la imposibilidad de agilizar tareas debido a la no flexibilidad de los procesos.
- **Bajo nivel de usabilidad desde dispositivos móviles:** No contar con una aplicación nativa que se integre al sistema, trae como consecuencia que la calidad gráfica de las consultas a los datos serían de bajo nivel debido que se realizaría a través de un navegador.

En la actualidad existe una primera versión desarrollada de la aplicación, la cual presenta los módulos para Comercial y Corresponsal, lo que imposibilita a otras áreas en la empresa como logística, transporte, Arrendamiento de contenedores, Agenciamiento aduanal, Facturación y Orden de pago a consultar la información tanto de sus áreas como la brindada por Comercial y Corresponsalía, que son las principales áreas de la empresa. Pero el sistema SIGAX del cual se pretende consumir servicios, ha cambiado elementos de su arquitectura en cuanto a su implementación tanto en organización de sus módulos y atributos, además ha migrado hacia versiones superiores de su *framework*. Por lo que la aplicación no está terminada, ya que no cuenta con los nuevos cambios.

A partir de la situación planteada se identifica como **problema a resolver** ¿Cómo permitir el acceso a datos del sistema SIGAX desde dispositivos móviles sin conexión permanente a la red?

Se define como **objetivo general**: Desarrollar una aplicación móvil que permita el acceso a datos del sistema SIGAX desde dispositivos móviles sin conexión permanente a la red.

Este problema enmarca como **objeto de estudio**: Las aplicaciones móviles para los Sistemas de Gestión Integral.

Se define como **campo de acción**: Aplicaciones móviles para el Sistema de Gestión Integral SIGAX. Para dar cumplimiento al objetivo general se desglosan los siguientes **objetivos específicos**:

- Analizar tecnologías que contribuyan a sustentar la posible propuesta de solución.
- Analizar metodologías, herramientas y tecnologías que contribuyen al desarrollo de la aplicación.
- Diseñar la propuesta de solución.
- Implementar la solución diseñada a partir de los requisitos funcionales y no funcionales definidos.
- Validar la propuesta de solución mediante pruebas.

Para dar cumplimiento a los objetivos planteados se trazaron las siguientes **tareas** a cumplir:

- Estudio de las aplicaciones móviles a escala internacional y nacional para Sistemas de Gestión Integral.
- Identificar las herramientas, tecnologías y técnicas actuales de desarrollo de aplicaciones Android.
- Definir los requisitos del software.
- Definir las metodologías de desarrollo, herramientas y tecnologías a utilizar.
- Realizar el diseño de la solución.
- Implementar funcionalidades.
- Probar el funcionamiento de las entidades.
- Comprobar la integración de la solución con el sistema gestor.

Métodos de investigación utilizados:

Métodos teóricos:

- **Analítico-Sintético**: Se aplica para identificar los elementos teóricos de la investigación empírica facilitando el análisis y comprensión de la documentación consultada, ayudando a establecer comparaciones entre varias aplicaciones Android, extraer y concretar los elementos y características

esenciales para el diseño de la propuesta de solución y realizar las conclusiones de la investigación científica.

- **Histórico-Lógico:** Se utilizó para realizar un estudio del estado del arte, analizando el uso de las aplicaciones para la gestión empresarial
- **Observación:** Su utilización sirvió de aporte al proceso de investigación la cual contribuyó a la clasificando de los sucesos de acuerdo con esquemas previstos. Se manejó este método con la finalidad de analizar el comportamiento de las aplicaciones estudiadas.
- **Experimental:** Se realizó durante el proceso de pruebas experimentales con el objetivo de asegurarse del cumplimiento del objetivo general planteado.

Estructura capitular:

Capítulo 1: Fundamentación teórica: Se ejecuta un estudio del estado del arte de aplicaciones Android para la gestión empresarial a escala internacional y nacional. Se definen tecnologías, herramientas, metodología a emplear, los cuales guíen el desarrollo de la aplicación.

Capítulo 2: Análisis y diseño de la solución: Se describe detalladamente las características del sistema a desarrollar, ofertando ejemplos de la metodología utilizada y la arquitectura de diseño.

Capítulo 3: Validación de la propuesta de solución: Se describen los principales aspectos vinculados a la implementación. Se especifican las pruebas ejecutadas a la aplicación con el objetivo de verificar su correspondencia con los requisitos definidos.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En este capítulo se define el marco teórico de la investigación, en la que se abordan una serie de conceptos asociados al dominio del problema para facilitar un mejor entendimiento del mismo. Se muestran y explican los lenguajes de programación, herramientas, tecnologías y metodologías a utilizar para el desarrollo de la aplicación, explicando el porqué de la selección de cada uno de estos componentes.

1.1 Conceptos asociados al dominio del problema

1.1.1 Dispositivos móviles

Dentro de los dispositivos móviles están los comunicadores de bolsillo, teléfonos con pantalla para Internet (*Internet Screen Phones*), sistemas de navegación de automóviles, sistemas de entretenimiento, teléfonos móviles, entre otros. De estos dispositivos los teléfonos móviles son dispositivos inalámbricos electrónicos basados en la tecnología de ondas de radio, que tiene la misma funcionalidad que cualquier teléfono de línea fija. Su principal característica es su portabilidad, ya que la realización de llamadas no es dependiente de ningún terminal fijo y no requiere ningún tipo de cableado para llevar a cabo la conexión a la red telefónica. Aunque su principal función es la comunicación de voz, como el teléfono convencional, su rápido desarrollo ha incorporado funciones adicionales como mensajería instantánea (SMS), agenda, juegos, cámara fotográfica, agenda, acceso a Internet, reproducción de video e incluso GPS y reproductor mp3. Inicialmente los teléfonos móviles sólo permitían realizar llamadas de voz y enviar mensajes de texto. Conforme a la tecnología fue avanzando, se incluyeron nuevas aplicaciones como juegos, alarma, calculadora y acceso WAP (acceso a Internet mediante páginas web especialmente diseñadas para móviles). Esta tecnología fue aplicada a los dispositivos móviles, los cuales son llamados en la actualidad como “*Smartphone*” (teléfonos inteligentes). Un Smartphone es un dispositivo electrónico que funciona como un teléfono móvil con características similares a las de un ordenador personal (7). Una característica importante de casi todos los teléfonos inteligentes es que permiten la instalación de programas para incrementar el procesamiento de datos y la conectividad. Estas aplicaciones pueden ser desarrolladas por el fabricante del dispositivo, por el operador o por un tercero (8). Debido a la investigación del actual subacápito, el autor de la presente investigación considera que los dispositivos móviles como: un dispositivo capaz de tener programas los cuales sirven de herramientas, facilitando la vida cotidiana de cualquier individuo al llevar a cabo tareas o procesos a través de dichas aplicaciones, dando paso a vincular las misma con la vida laboral. Por lo que da paso a la representación de un dispositivo móvil como medio fundamental para la propuesta de solución.

1.1.2 Comunicación inalámbrica

Se le llama comunicación inalámbrica a aquella que se lleva a cabo sin el uso de cables de interconexión entre los participantes. Existen varios tipos de comunicación inalámbrica tales como: señales de radio, microondas e infrarrojos. En todos estos casos una antena transmite una señal electromagnética la cual es recibida por otra antena. La distancia que puede alcanzarse depende de la potencia del emisor y la frecuencia de transmisión. Una de las muchas aplicaciones de la tecnología inalámbrica es que permite conectar nodos mediante enlaces inalámbricos y formar redes de muy diversa naturaleza. Esto, obviamente, posibilita que una red inalámbrica presente ventajas sobre una red por cable. Las más importantes son las siguientes:

- **Movilidad.** Estas redes proporcionan, a los usuarios de las mismas, información en tiempo real desde cualquier punto del entorno en el que se encuentran desplegadas. Esto constituye una mejora considerable respecto a las redes por cable en las que el usuario debe permanecer en una posición fija.
- **Escalabilidad.** En estas redes resulta muy sencillo la incorporación de otros usuarios a la red, al contrario de lo que ocurre en las redes cableadas donde dicha labor resulta más tediosa. En definitiva, estas redes se pueden configurar en muy diversas topologías adaptándose más eficientemente a la aplicación para la que se la requiere, con el beneficio que ello conlleva.

Debido al creciente desarrollo de las comunicaciones inalámbricas y la reducción en el coste de los dispositivos que incorporan esta tecnología han surgido, en un corto intervalo de tiempo, una gran cantidad de iniciativas. Esto ha dado lugar a la aparición de multitud de tecnologías cuyo principal objetivo es permitir el intercambio de información entre dispositivos inalámbricos y que se diferencian unas de otras por la tasa de transmisión y el rango de alcance. De las diferentes clasificaciones en cuanto a rango y alcance de señal de estas tecnologías están (9):

- **WPAN (*Wireless Personal Area Network*):** Permiten a los usuarios establecer comunicaciones inalámbricas entre dispositivos (PDA, portátiles, teléfonos móviles, etc) en un área reducida.
- **WLAN (*Wireless Local Area Network*):** Permiten a los usuarios establecer conexiones inalámbricas dentro de un área local (edificio corporativo, campus empresarial, aeropuerto, entre otros). Se pueden utilizar en lugares donde la instalación de un cableado extenso es prohibitiva, o bien para complementar a una red local de cable existente de manera que los usuarios puedan trabajar en diferentes lugares dentro de un edificio.
- **WMAN (*Wireless Metropolitan Area Network*):** Permite a los usuarios establecer conexiones inalámbricas entre varias ubicaciones dentro de un área metropolitana (entre varios edificios de

oficinas de una ciudad o en un campus universitario), evitando el coste de instalación de cables de cobre o fibra y el alquiler de las líneas.

- **WWAN (*Wireless Wide Area Network*):** Permiten a los usuarios establecer conexiones inalámbricas a través de redes remotas públicas o privadas. Estas conexiones se pueden mantener a través de zonas geográficas extensas, como ciudades o países, mediante el uso de antenas en varias ubicaciones o sistemas satélite.

Dentro de la WLAN, se encuentra una de las tecnologías inalámbricas más conocida que es la *Wifi*, la cual presenta elevadas tasas de transmisión (hasta 54 Mbps), alcances adecuados (hasta 100 metros), privacidad, sencillez y coste reducido. Se emplea como sustitutivo de cables en una red local y permite la conexión desde diferentes lugares dentro del rango de alcance y movilidad (9). A partir de la investigación anterior, el autor puede definir que la conexión inalámbrica es uno de los pilares esenciales para la propuesta de solución, debido las ventajas mencionadas de movilidad y escalabilidad.

1.1.3 Aplicación móvil

Las aplicaciones móviles se desarrollan en un entorno dinámico e incierto. Generalmente, son pequeñas, no críticas, aunque no menos importantes. Están destinadas a un gran número de usuarios finales, incluso a negocios empresariales y son liberadas en versiones rápidas para poder satisfacer las demandas del mercado. El desarrollo de aplicaciones móviles es, actualmente, un gran desafío, dado las demandas específicas y las restricciones técnicas de un entorno móvil, tales como dispositivos con capacidades limitadas, pero en evolución continua; varios estándares, protocolos y tecnologías de red, necesidad de operar sobre diferentes plataformas, requerimientos específicos de los usuarios y las exigencias estrictas en tiempo del mercado.

1.1.4 ERP

Un sistema ERP es una aplicación informática que permite gestionar todos los procesos en cuanto al negocio de una compañía de forma integrada. Sus siglas provienen del término en inglés *Enterprise Resource Planning*. Es un sistema de planificación de los recursos y gestión de la información, satisfaciendo las necesidades en cuanto a la gestión empresarial. Se trata de un programa de software integrado que permite a las empresas evaluar, controlar y gestionar más fácilmente su negocio en todos los ámbitos. Los sistemas ERP se caracterizan por su gran capacidad de adaptación, son abiertos y multiplataforma. Es un programa de gestión empresarial diseñado para cubrir todas las exigencias de las áreas funcionales de la empresa, de forma que crea un flujo de trabajo (*Workflow*) para los distintos usuarios, permitiendo agilizar los distintos tipos de trabajos, disminuyendo el tiempo real de las tareas repetitivas y permitiendo además el aumento de comunicación entre todas las áreas que integran la empresa. Por lo general este tipo de

sistemas está compuesto de módulos como Recursos Humanos, Ventas, Contabilidad y Finanzas, Compras, Producción entre otros, brindando información cruzada e integrada de todos los procesos del negocio, este programa tipo de programas se caracteriza por su facilidad de modularidad, integración de procesos, facilidad de consulta, estandarización e interfaces con otras aplicaciones (5). Por toda esta información el autor de esta investigación puede llegar a la conclusión que un ERP es un software de gestión empresarial que permite planificar y controlar los procesos y recursos de negocio de una empresa. Por la conclusión anterior se puede establecer un punto de partida para comprender la función del sistema SIGAX dentro de la empresa TRANSCARGO, el cual se encuentra en desarrollo sobre Odoo, sistema integrado ERP de código abierto.

1.2 Estudio de soluciones similares

En el presente acápite se realiza un análisis de las principales aplicaciones móviles para la gestión de procesos empresariales mediante dispositivos móviles. La investigación se enfoca en la integración de sistemas ERP con aplicaciones móviles y el análisis de las aplicaciones móviles desarrolladas para la integración con sistemas desarrollados con Odoo.

1.2.1 Aplicaciones ERP móviles

En el presente sub-acápite se estará tratando de algunos sistemas ERP móviles existentes en la actualidad. El análisis de estas soluciones puede servir como pilar para determinar si alguna de ellas puede ser utilizada con el objetivo de dar solución al problema de la presente investigación.

A3ERP

El software A3ERP ofrece en tiempo real una visión global, el control y trazabilidad de todos los procesos productivos y administrativos de una empresa, garantizando una gestión eficiente, ya que optimiza recursos, simplifica y automatiza procesos dando la facilidad de la toma de decisiones. Permite una gestión unificada de las diferentes áreas de la empresa (laboral, comercial, compras, logística, producción y financiera). Además, se adapta a cada empresa con distintas versiones que responden a sus distintas necesidades y a cualquier sector de actividad. *Sales mobility* es una solución móvil para la gestión comercial, muy fácil de usar pensada para que la fuerza de ventas de tu empresa pueda realizar todas las gestiones de auto venta y preventa desde una *Tablet* o *Smartphone* en cualquier momento y desde cualquier lugar y sincronizarlas de forma automática con A3ERP, la solución integral de gestión para pequeñas y medianas empresas. Algunas de sus principales características son (10):

- **Global:** Gestiona todas las áreas la empresa de forma fácil y ágil, aumentando la productividad y la eficiencia, reduciendo costes y ganando competitividad.

- **Escalable:** Puede crecer en funcionalidades y puestos de trabajo al ritmo en que tu empresa lo requiera, sin tener que cambiar de sistema de ERP.
- **Adaptable:** Su capacidad de parametrización permite ajustarlo a las necesidades de alguna actividad, gestión y a las particularidades de una empresa.
- **Información en tiempo real:** Se puede conocer al momento el estado real de la empresa en todas las áreas, ayudando a tomar las decisiones más adecuadas.

SAGE X3

Sage X3 (*Sage Business Cloud Enterprise Management*), ofrece capacidades de gestión de la producción, la cadena de suministro y financieras más rápidas, sencillas y flexibles por una pequeña parte del coste y la complejidad de los sistemas ERP convencionales. Transforma el modo en el que una empresa compete y crece gracias a una solución de gestión empresarial más rápida, sencilla y flexible con un coste y una complejidad mucho menores que los sistemas ERP empresariales convencionales. Sage X3 aporta tanto la capacidad de disponer de los datos actualizados relevantes, como la opción de modificarlos desde cualquier dispositivo, ya sea ordenador, Tablet o móvil, contribuyendo de manera directa a garantizar el éxito de la organización. Sage X3 está diseñado desde su versión 7.0 para acceder a través de cualquier navegador de Internet y es capaz de adaptarse al tamaño de la pantalla donde se visualice (diseño web adaptable). Además, para incrementar la usabilidad en dispositivos con un tamaño menor de pantalla, Sage ha creado aplicaciones móviles para teléfonos con Sistema Operativo Android o iOS. Gracias a estas características se puede tener acceso a datos de las diferentes áreas como (11):

- **Rapidez:** *Enterprise Management* acelera todos los procesos empresariales (desde las compras hasta el almacenamiento, las ventas, el servicio de atención al cliente y la gestión financiera) y ofrece información más rápida sobre los costes y el rendimiento en todos los pasos y a nivel global.
- **Sencilla:** se adapta al flujo de trabajo, a las preferencias y a los cargos exclusivos de los usuarios, ofrece acceso seguro móvil y web a los datos que necesitan, a la vez que simplifica la gestión de la infraestructura de software de la empresa con un paquete de aplicaciones coherente.
- **Flexible:** ofrece aplicaciones y opciones de configuración flexibles para admitir los procesos específicos del sector, que se pueden usar tanto alojados en tu empresa o alojados en un *hosting* privado exclusivo para la empresa, lo que te proporciona todo el control de la estrategia de TI (Tecnologías de la información).

1.2.2 Aplicaciones móviles para Odoo

Como fue mencionado anteriormente, el sistema SIGAX aún en desarrollo utilizando la plataforma Odoo, lo cual hace necesario el análisis de aplicaciones con las que se puedan establecer una conexión con el sistema SIGAX. Odoo es uno de los sistemas ERP más conocido en el mundo, un completo sistema de gestión empresarial de código abierto creado en Python, sin coste de licencias que cubre las necesidades de las áreas de: Contabilidad y Finanzas, Ventas, RRHH, Compras, Proyectos, Almacenes (SGA), CRM y Fabricación entre otras (12). En el presente sub-acápite se estará abordando sobre aplicaciones móviles con las que se les puede integrar a un sistema desarrollado en Odoo.

MyOdoo

MyOdoo es un sistema ERP para dispositivos móviles, soportado en los sistemas operativos de Android y iOS. Capaz de establecer conexión con Odoo y acceder a sus módulos. MyOdoo es un modo de planificación móvil de recursos empresariales el cual permite acceder a (13):

- **Odoo messages:** Administrar lista de mensajes y tareas.
- **Odoo contacts:** Acceso a la agenda de Odoo en línea y actualizar los detalles de sus contactos.
- **Odoo Timesheets:** Rellenar hojas de horas en 2 clics mientras se trabaja en proyectos, tareas o contratos.
- **Odoo notes:** Crear y acceder a las notas del Smartphone o Tablet, dando la posibilidad de compartir y colaborar con los compañeros de trabajo u otros.
- **Odoo sales:** Administración de ventas. Desde presupuestos a facturas, tomando el control de los flujos comerciales desde la aplicación.
- **Odoo activities:** Registrar el tiempo en cualquier momento y lugar incluso sin red. Descubriendo así cuánto tiempo se emplea en las tareas diarias.

Entre sus principales funcionalidades están:

- **Aplicación modular** (Utilizando el mismo enfoque que las apps Odoo)
- **Soporte de varias cuentas:** Desde MyOdoo se podrá acceder a los datos desde varias cuentas de usuarios registradas en el sistema.

- **Widget para Gastos y Hojas de horas:** MyOdoos cuenta con *widget*¹ para visualizar los gastos y las hojas de horas registrados en el sistema.
- **Sincronización de la Agenda y el Calendario:** Permite mostrar datos de la agenda y el calendario registrados en el sistema.
- **Modo fuera de línea:** MyOdoos tendrá sus datos disponibles sin conexión y con su Smart Offline Mode, los datos se sincronizarán en segundos garantizando que se obtendrán los datos necesarios, dando la posibilidad de poderlos consultar en la posterioridad sin conexión.

Modoo

Modoo (Mobile Odoos) es sistema OpenERP, la cual ayudará a generar una aplicación para iOS o Android. Esta aplicación se conectará e incluirá los módulos del sistema OpenERP utilizando servicios web. Los gerentes y los empleados importantes tendrán acceso a la aplicación móvil y podrán ver datos importantes, podrán realizar tareas como aprobar el pedido de venta, aprobar solicitud de permiso entre otras. Algunos módulos que tiene la aplicación son:

- Calendario
- Facturas de ventas
- Lista de productos
- Tareas
- CRM

Los datos modificados o introducidos en la aplicación se sincronizarán con Odoos haciéndolos disponibles para todos los usuarios. El control de acceso de los módulos se basa en los permisos establecidos en Odoos, permite que los eventos en Odoos se sincronizan con el calendario del dispositivo móvil y cualquier usuario o administrador puede iniciar sesión (14).

Odoos Mobile

Odoos Mobile es un framework de código abierto, el cual permite integrar Odoos en una aplicación Android. Este marco contiene su propio ORM para manejar la base de datos local del móvil (SQLite). Cuenta con servicios y proveedores previamente desarrollados para sincronizar los datos de la aplicación con el sistema en Odoos, permitiendo el acceso de los módulos desarrollados en el sistema. Con el acceso a los módulos del sistema Odoos, la aplicación podrá consultar registros, ventas, reportes y tareas en marcha. Cuenta con

¹ *Widget*: Pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets. Entre sus objetivos están en dar fácil acceso funciones frecuentemente usadas y proveer de información visual.

notificaciones las cuales mantendrán informado al usuario sobre cualquier tarea o acción que el mismo siga. El objetivo de esta aplicación no es llevar todo el contenido de los módulos al dispositivo, sino definir algunas características específicas, dándole más provecho de su uso en un dispositivo móvil (15).

1.2.3 Resultados obtenidos sobre el análisis de soluciones similares

Tras llevar acabo el análisis de soluciones similares para la gestión empresarial, se ha llegado a la conclusión de que ninguno de los sistemas analizados es aplicable al sistema SIGAX de la empresa TRANSCARGO. Debido a que los sistemas ERP móviles vistos, todos son hechos para integrarlos con sistemas ERP específicos, por lo que su uso no es extensible al sistema en cuestión. Debido a que en el caso de las aplicaciones móviles para Odo, su funcionamiento está orientado al acceso a los módulos predefinidos del ERP (inventario, ventas, CRM, entre otros), por lo que no es posible el acceso desde estas aplicaciones a los módulos definidos en el sistema SIGAX. Por tanto, se puede llegar a la conclusión de la necesidad del desarrollo de una aplicación personalizada. También la aplicación estará centrada en mostrar datos críticos del sistema y no a realizar transformaciones en el mismo. Aunque se reutilicen características de los sistemas similares como:

- Al igual que las aplicaciones similares analizadas, la propuesta de solución no tendrá como objetivo llevar todo el sistema de gestión al entorno móvil, solamente sus características.
- La aplicación estará ajustada para la visualización de información crítica para el negocio que en realizar cambios en el sistema.
- Se podrá acceder al sistema sin conexión permitiendo además consultar los datos deseados, previamente de haber realizado una sincronización desde el dispositivo con el sistema con el objetivo de actualizar los datos de la aplicación.

1.3 Tipos de aplicaciones móviles

En el presente acápite se estará tratando sobre los tipos de aplicaciones móviles con el objetivo de llegar a sus características y definir el tipo de aplicación móvil a utilizar. De los tipos de aplicaciones móviles existentes están las: nativas, web e híbridas.

1.3.1 Aplicaciones Nativas

Una aplicación nativa es la que se desarrolla de forma específica para un determinado sistema operativo. Estas aplicaciones pueden acceder a los sistemas operativos del equipo móvil para facilitar el uso de dispositivos como brújula, cámara, correo, GPS, entre otros. El código fuente se compila para obtener el código ejecutable, proceso similar que al utilizado para las tradicionales aplicaciones de escritorio. Dentro de las ventajas y desventajas de aplicaciones están (16):

Ventajas

- Acceso completo al dispositivo, en cuanto al sistema y al hardware
- Envío de notificaciones o “avisos” a los usuarios
- Acceso completo a la aplicación, en algunas no es necesario que estén conectadas a Internet para su funcionamiento

Desventajas

- El código del cliente no es reutilizable para diferentes sistemas operativos
- Requiere de un costo para distribuirla en una tienda, y dependiendo el sistema, para su uso del entorno de desarrollo
- Necesitan aprobación para ser publicadas en la plataforma

1.3.2 Aplicaciones Web

Son aquellas que son desarrolladas con lenguajes de programación web como: CSS, JavaScript, PHP y HTML, donde este último garantiza la compatibilidad en distintas plataformas, donde se utiliza el navegador del cliente como interfaz de usuario. Las aplicaciones web con el paso del tiempo presentan un alto nivel de dinamismo lo que las asemeja cada vez más a una aplicación de escritorio. Las aplicaciones web suelen necesitar un menor tiempo de desarrollo que sus equivalentes nativos. Al analizar la integración de las aplicaciones con las funcionalidades que ofrecen los móviles, se puede detectar que los lenguajes web no tienen todavía compatibilidad con todas las funcionalidades/Apis nativas (tales como el GPS, acelerómetro, captura de imágenes, audio y vídeo, 3D, agenda de contactos, calendario, entre otros). Otro de los puntos críticos de algunas de estas aplicaciones es que no funcionan sin conexión (17).

1.3.3 Aplicaciones Híbridas

Como su nombre lo indica son aplicaciones desarrolladas en parte con el entorno de desarrollo nativo y en parte en lenguaje web. Al igual que las aplicaciones web parte de las híbridas requieren una menor inversión inicial, debido a que la mayor parte del desarrollo no se debe repetir para cada sistema operativo, aunque al menos necesitan el desarrollo del contenedor nativo que aloje las partes en lenguaje web. Dado que su funcionamiento se basa en recursos web, tanto como las aplicaciones web, las aplicaciones híbridas necesitan una buena capacidad de concurrencia en los servidores donde se albergan y una buena conectividad para funcionar de manera óptima (17).

Tabla 1: Tabla de comparación de los sistemas operativos móviles

Características	Página móvil	Aplicación nativa	Aplicación híbrida
Plataforma	Navegadores móviles	iPhone OS (iOS), Windows Mobile, Blackberry OS, Symbian, Android	iPhone OS (iOS), Windows Mobile, Blackberry OS, Symbian, Android
Distribución	URL y códigos QR	Tiendas de aplicación según la plataforma	Tiendas de aplicación según la plataforma
Instalación	Se puede acceder directamente y puede quedar disponible mediante un <i>launcher</i> en el dispositivo	Se realiza una vez y queda disponible	Se realizan una vez y queda disponible para todas las plataformas
Costos de desarrollo	Menores	Mayores	Menores que los de las nativas
Rendimiento	HTML5 mejora la infraestructura de la red	Más rápido, especialmente si quiere procesos gráficos pesados	Se desarrolla como nativo cuando el rendimiento sea esencial
Integración de hardware	Limitada	Completa	Buena
Acceso fuera de línea	Solo en algunos dispositivos mediante HTML5	Completo	Acceso a la aplicación aunque no a sus funcionalidades esenciales
Usabilidad	Buena	Gran cantidad de efectos amigables en la interfaz atractivos para el usuario	Utiliza lo mejor de lo nativo y lo mejor de la red

1.3.4 Selección del tipo de aplicación a utilizar

A partir del análisis anterior se puede llegar a la decisión de implementar una aplicación nativa, debido a las ventajas que esta presenta. Se pretende que el sistema a desarrollar no tenga que estar conectada a una red para poder visualizar los datos con los que se cuenta, si anteriormente la aplicación se conectó y sincronizó con el sistema SIGAX, por lo que como se expone anteriormente, no se puede contar con una aplicación Web o Híbrida donde es necesario contar con una buena conexión para lograr un buen funcionamiento de la aplicación. Además, se puede sacar mayor provecho gráfico basándose en el nivel de usabilidad de la aplicación.

1.4 Análisis sobre los sistemas operativos móviles

Los sistemas operativos para dispositivos móviles suelen ser menos robustos que los diseñados para computadoras de escritorio o portátiles. Orientados hacia la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos, orientados a necesidades específicas. En el presente acápite se abordará sobre los sistemas operativos móviles más utilizados donde se identificarán sus características.

1.4.1 Sistema operativo Android

El sistema operativo Android es una capa compleja entre el hardware y el usuario, que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndole de los complicados procesos necesarios para llevarlas a cabo.

Android presenta una serie de cualidades, las cuales se combinan en una misma solución (18):

- **Plataforma realmente abierta:** Es una plataforma de desarrollo libre basada en Linux y de código abierto.
- **Adaptable a cualquier tipo de hardware:** Android no ha sido diseñado exclusivamente para su uso en teléfonos y tabletas. Hoy en día podemos encontrar relojes, cámaras, electrodomésticos y gran variedad de sistemas empotrados que se basan en este sistema operativo.
- **Portabilidad asegurada:** Las aplicaciones finales son desarrolladas en Java lo que nos asegura que podrá ser ejecutada en cualquier tipo de CPU. Esto se consigue gracias al concepto de Máquina Virtual.
- **Arquitectura basada en componentes inspirados en Internet:** Por ejemplo, el diseño de la interfaz de usuario se hace en XML, lo que permite que una aplicación se ejecute con una correcta visualización en un móvil de pantalla reducida o en un TV.

- **Aceptable nivel de seguridad:** Los programas se encuentran aislados unos de otros gracias al concepto de ejecución dentro de una caja que hereda de Linux. Además, cada aplicación dispone de una serie de permisos que limitan su rango de actualización (Servicios de localización, acceso a Internet, entre otras).
- **Alta calidad de gráficos y sonido:** Gráficos vectoriales suavizados, animaciones inspiradas en Flash, gráficos en tres dimensiones basados en OpenGL (Especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D). Incorpora códec estándar más comunes de audio y video.

Dicho sistema operativo permite programar aplicaciones empleando una variación de Java llamada Dalvik (Máquina virtual que utiliza la plataforma), y facilita todas las interfaces necesarias para desarrollar fácilmente aplicaciones que accedan al funcionamiento del teléfono (como el GPS, las llamadas, la agenda, etcétera) empleando el lenguaje de programación Java. Su sencillez, junto a la existencia de herramientas de programación gratuitas, es principalmente la causa de que existan cientos de miles de aplicaciones disponibles, que amplían la funcionalidad de los dispositivos y mejoran la experiencia del usuario (18).

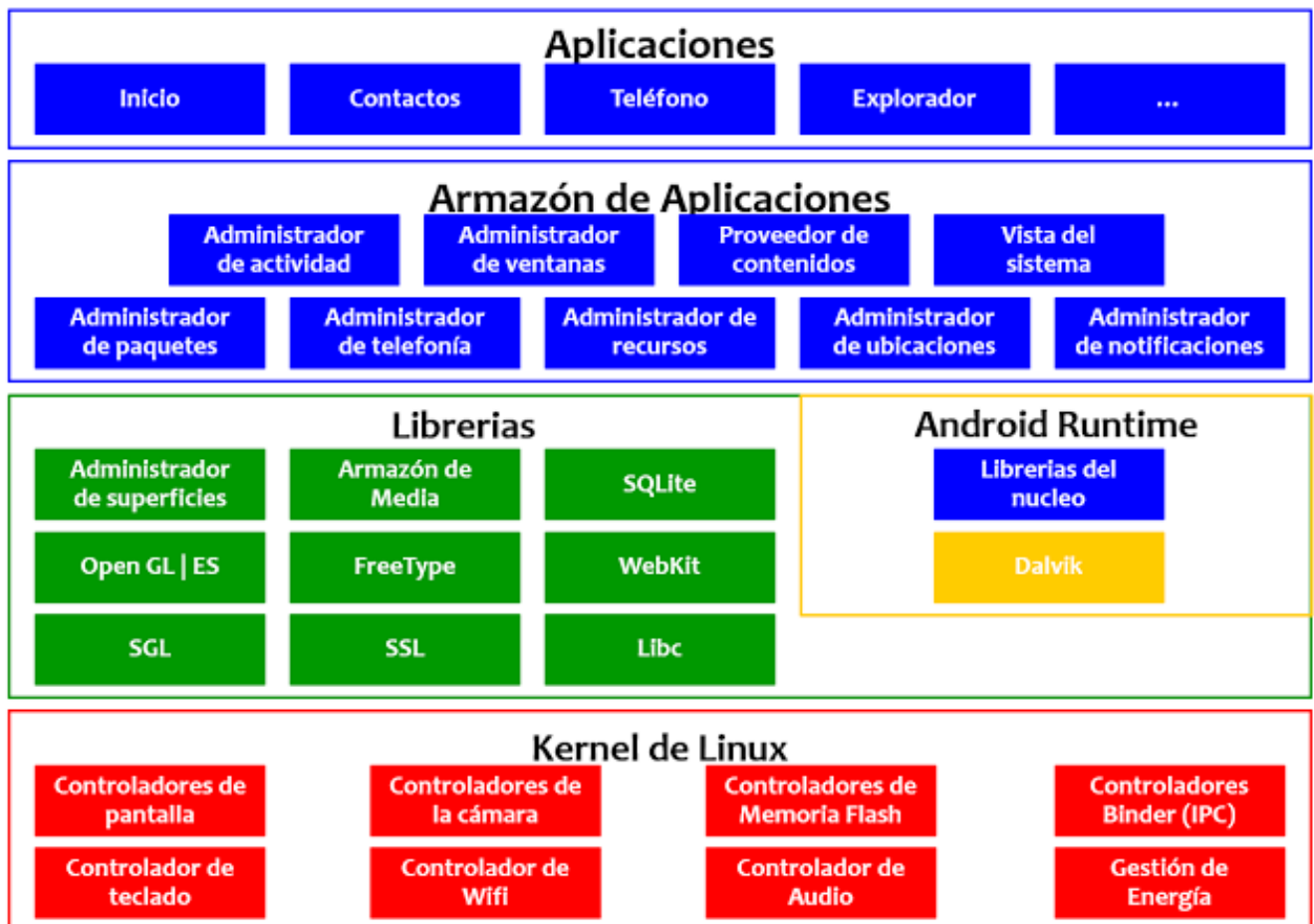


Figura 1:Arquitectura de componentes del sistema operativo Android

Fuente: <http://webipedia.es/2016/12/19/introduccion-a-android-22/>

1.4.2 Sistema operativo iOS

iOS es un sistema operativo móvil creado por Apple.inc inicialmente desarrollado para el iPhone y posteriormente usado en dispositivos como el iPod Touch, iPad y el Apple TV, todos productos originales de Apple. Este sistema operativo gestiona el hardware del dispositivo en el que se encuentra instalado para implementar aplicaciones nativas. El sistema viene con aplicaciones de aplicaciones como teléfono, correo y navegador Safari que prestan servicios estándar del sistema para el usuario. Todo el sistema se encuentra en la partición “/root” y ocupa menos de 500MB. El sistema operativo iOS fue desarrollado usando el conocimiento del sistema operativo MAC OS X. La arquitectura de iOS está constituida por una serie de capas o niveles de abstracción, las cuales son (19):

- La capa del núcleo del sistema operativo.
- La capa de core services.
- La capa de media.
- La capa de “Cocoa Touch”.

1.4.3 Sistema operativo Windows Phone

El sistema operativo Windows Phone fue desarrollado por la empresa Microsoft para teléfonos inteligentes y otros dispositivos móviles, el cual fue lanzado con la finalidad de suplantar el conocido Windows Mobile. Con esta plataforma, Microsoft llega como una nueva propuesta de movilidad diseñada para integrar y maximizar experiencias a través de la web, la PC y el teléfono, mediante aplicaciones y servicios. Microsoft ha establecido una serie de requisitos mínimos de hardware para que las compañías que fabrican los teléfonos puedan utilizar Windows Phone 7 en sus equipos. Todo esto brinda ciertas ventajas a los desarrolladores de aplicaciones, pues reduce la fragmentación de la plataforma al no tener una variedad infinita de tamaños, formas y recursos de hardware. Windows Phone se está disponible en los equipos de sus principales socios comerciales, tales como HTC, HP, LG, Toshiba, Sony Ericsson, Samsung, entre otros (20).

1.4.4 Selección del sistema operativo a utilizar

Con el objetivo de llevar a cabo la propuesta de solución se ha decidido utilizar Sistema operativo Android como plataforma de despliegue y desarrollo, debido a las ventajas expuestas anteriormente. Según en el sitio Clarín.com, la compañía que se encarga de analizar el tráfico en la web informó en el 2017, que el 37,93% de los usuarios navega desde un dispositivo con Android, una fracción apenas más grande que el 37,91% de Windows, como registros globales. De manera específica en la India, por ejemplo, el 61,78% de

los usuarios que acceden a Internet usan Android (21). En Cuba el sistema operativo de los dispositivos móviles de la mayor parte de la población es Android. En la mayoría de los casos esto es debido a que en la Empresa de Telecomunicaciones de Cuba (ETECSA), la mayor cantidad de dispositivos que sale a la venta en dicha institución son con el sistema operativo Android.

1.5 Metodología de desarrollo

Según Somerville, para muchas personas el software es solo un programa de computadora. Sin embargo, comenta que son todos aquellos documentos asociados a la configuración de datos que se necesitan para hacer que estos programas operen de manera adecuada. Estos productos de software se desarrollan para algún cliente en particular o para un mercado en general. Para el diseño y desarrollo de proyectos de software se aplican metodologías, modelos y técnicas que permiten resolver el problema. Todo proyecto necesita hacer uso de las metodologías de desarrollo de software, teniendo en cuenta que las mismas establecen una serie de pasos y procedimientos que deben seguirse, si se desea obtener un proyecto exitoso. Utilizando metodologías para la construcción del software, se logra que los proyectos lleguen finalmente a ser exitosos desde los puntos de vista de objetivos de negocio, costos, funcionalidad, sencillez y capacidad de soporte. En general, las metodologías llevan a cabo una serie de procesos comunes que constituyen buenas prácticas para lograr los objetivos antes mencionados, independientemente de cómo hayan sido diseñadas cada una. Permiten que el trabajo sea más organizado, que se ahorren recursos, tiempo y se obtenga un producto con mayor calidad.

1.5.1 Metodologías ágiles

Existen diversas metodologías ágiles. Todas pretenden de una forma u otra desarrollar software de manera rápida y respondiendo satisfactoriamente a los cambios que surjan durante el ciclo de vida del proyecto. El Manifiesto Ágil que se desarrolló en EEUU en el año 2011, plantea los siguientes principios.

- La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporten valor.
- Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- La atención continua a la calidad técnica y el buen diseño mejora la agilidad. La simplicidad es esencial.

- En intervalos regulares el equipo reflexiona respecto a cómo llegar a ser más efectivos y según esto ajusta su comportamiento.

1.5.2 Metodología de desarrollo de software RUP

Rational Unified Process (Proceso Unificado de Desarrollo), en adelante RUP, es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, mediante el Lenguaje Unificado de Modelado, en lo adelante UML, y el trabajo con otras metodologías. Al ser un proceso, en su modelación se definen los elementos de: actividades, artefactos, flujos de trabajo y trabajadores, donde sus principales características son (22):

- **Guiado por casos de uso:** Los casos de uso muestran lo que los usuarios necesitan y desean. Los casos de uso le dan una dirección al proceso de desarrollo basada en la perspectiva del usuario. Los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- **Iterativo e incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra procesos de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Es práctico dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.
- **Centrado en la arquitectura:** La arquitectura representa la visión común del sistema completo en la que el cual la sincronización del equipo de proyecto es de vital importancia, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para entenderlo, producirlo y desarrollarlo económicamente.

1.5.3 Metodología de desarrollo AUP

AUP (*Agil Unified Process*) es una metodología de desarrollo de software, siendo una versión simplificada de la metodología de desarrollo RUP. Describe de fácil y simple el desarrollo de software de negocio, empleando técnicas ágiles y conceptos que se mantienen válidos en RUP. Aplica técnicas como:

- Desarrollo dirigido por pruebas.
- Modelado ágil.
- Gestión de cambios ágil.
- Refactorización de base de datos para mejorar la productividad.

AUP propone que los elementos de alto riesgo sean de prioridad en el desarrollo y sean abordados en etapas tempranas del mismo.

Se establece cuatro fases que transcurren de manera consecutiva:

- **Inicio:** El objetivo de esta fase es obtener una comprensión común entre el cliente y el equipo de desarrollo, del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo
- **Elaboración:** El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y validar la arquitectura.
- **Construcción:** Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- **Transición:** El sistema se lleva a entornos donde se somete a pruebas de validación y aceptación; finalmente se despliega en los sistemas de producción.

La Universidad de las Ciencias Informáticas (UCI) desarrolló una versión de la metodología de desarrollo de software AUP, con el fin de crear una metodología que se adapte al ciclo de vida definido por la actividad productiva de la misma. La versión de la metodología desarrollada por la universidad decide mantener para el ciclo de vida de los proyectos la fase de Inicio, pero modificando el objetivo de la misma y se unifican las restantes fases de la metodología de desarrollo de software AUP en una sola, nombrada Ejecución y agregándose también una nueva fase llamada Cierre (23).

Fases AUP	Fases Validación AUP-UCI	Objetivos de las fases (Variación AUP-UCI)
Inicio	Inicio	Durante el inicio del proyecto se llevan a cabo actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
Elaboración	Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del
Construcción		
Transición		

		proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
	Cierre	En esta fase se analizan tanto los resultados del proyecto como la ejecución y se realizan las actividades formales de cierre del proyecto.

Para el desarrollo de la solución se empleará la metodología de desarrollo de software AUP-UCI escenario cuatro, debido a que el proyecto no es extenso y ya tiene evaluado al negocio a informatizar, dando como resultado la obtención de un negocio bien definido. Además, el cliente estará en constante intercambio con el equipo de desarrollo, con el objetivo de convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos.

1.6 Herramienta y tecnologías a utilizar.

Según Somerville, para muchas personas el software es solo un programa de computadora. Sin embargo, comenta que son todos aquellos documentos asociados a la configuración de datos que se necesitan para hacer que estos programas operen de manera adecuada. Estos productos de software se desarrollan para algún cliente en particular o para un mercado en general. Para el diseño y desarrollo de proyectos de software se aplican metodologías, modelos y técnicas que permiten resolver el problema (24).

1.6.1 Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado en adelante UML es un lenguaje de modelado visual que se usa para especificar, construir, visualizar y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. UML incluye conceptos semánticos, notación, y principios generales. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código, así como generadores de informes. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientado a objetos. El lenguaje de modelado capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguajes de programación, así como construir modelos por ingeniería inversa a partir de programas existentes (25).

Herramienta de modelado: Visual Paradigm

Se entiende por herramientas de Ingeniería de Software Asistida por Ordenador, en *inglés Computer Aided Software Engineering* en lo adelante CASE, como el “conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software” (26).

Visual Paradigm es una herramienta que sirve para realizar modelado usando el lenguaje UML. Presenta características gráficas muy cómodas que facilitan la realización de los diagramas siguientes: diagramas de clases, casos de uso, comunicación, secuencia, estado, entidad-interrelación, actividades y componentes. Permite la exportación de imágenes a los formatos JPG, PNG y SVG. Puede integrarse con diversos entornos de desarrollo integrado como NetBeans, JDeveloper, Eclipse, JBuilder (27).

Visual Paradigm se escogió para el proceso de modelado porque es fácil de emplear y cuenta con todas las herramientas para modelar los diagramas que se necesitan durante el desarrollo del proyecto. El equipo se siente identificado y tiene en un ambiente de trabajo favorable, debido a que los objetos se pueden modelar sin mucha dificultad.

1.6.2 Lenguaje de programación a utilizar

Java 8

Para la implementación de la propuesta de solución se ha decidido utilizar como lenguaje de programación Java 8 debido a sus características las cuales beneficiarán al desarrollo del sistema en cuestión, dichas características son:

- Realiza varias comprobaciones en compilación y en tiempo de ejecución, Libera al desarrollador de tener que realizar la liberación de memoria.
- Es el lenguaje nativo y más utilizado para el desarrollo de aplicaciones Android.
- Cuenta con numerosas librerías desarrolladas por terceros las cuales aumentan sus funcionalidades.
- Es un lenguaje orientado a objetos, multiplataforma, de buena seguridad y gran rendimiento.
- Posee una amplia documentación digital y varias comunidades de desarrollo.

1.6.3 Entorno de desarrollo integrado

Android Studio

Entorno dedicado en exclusiva a la programación de aplicaciones para dispositivos Android, proporcionando a Google un mayor control sobre el proceso de producción. El presente IDE de desarrollo es el seleccionado para llevar a cabo el desarrollo del sistema en cuestión, debido a las características (28):

- Utiliza ProGuard para optimizar y reducir el código del proyecto al exportar a APK (muy útil para dispositivos de gama baja con limitaciones de memoria interna)
- Integración de la herramienta Gradle encargada de gestionar y automatizar la construcción de proyectos, como pueden ser las tareas de *testing*, compilación o empaquetado.
- Interfaz especificada para el desarrollo en Android.
- Posibilita el control de versiones accediendo a un repositorio desde el que poder descargar Mercurial, Git, Github o Subversion.
- Alertas en tiempo real de errores sintácticos, compatibilidad o rendimiento antes de compilar la aplicación.
- Vista previa en diferentes dispositivos y resoluciones.
- Editor de diseño que muestra una vista previa de los cambios realizados directamente en el archivo XML.

Tabla 2: Requerimientos del sistema

Windows	Mac OS	Linux
Microsoft Windows 8/7/Vista/2003 o superior (32 o 64 bit)	Mac OS X 10.8.5 o superior, hasta la10.9 (<i>Mavericks</i>)	GNOME o entorno de escritorio KDE
Mínimo de 2 GB de RAM, recomendado 4 GB de RAM		
400 MB de espacio en disco		
Necesita de al menos 1 GB para Android SDK, emulador de imágenes del sistema, y cachés		
Resolución mínima de pantalla de 1280 x 800		
Java Development Kit (JDK) 7 o superior		

Android Studio ha pasado a ser a ser el entorno recomendado para el desarrollo de aplicaciones en Android, al tratarse de un IDE oficial de Google en colaboración con JetBrains (compañía de desarrollo software especializada en diseño de IDEs). Permite la creación de nuevos módulos dentro de un mismo proyecto,

sin necesidad de estar cambiando de espacio de trabajo para el manejo de proyectos. Su forma de construir los paquetes.apk, mantiene el uso de Gradle, proporcionando una serie de ventajas más acorde a un proyecto Java:

- Facilita la distribución de código, y por lo tanto el trabajo en equipo.
- Reutilización de código y recursos.
- Permite compilar desde línea de comandos, para aquellas situaciones en las que no esté disponible un entorno de desarrollo.

1.6.4 Framework de desarrollo

En el desarrollo de Software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en el cual otro proyecto de software puede ser organizado y desarrollado. Los *frameworks* suelen incluir (28):

- Soporte de programas
- Bibliotecas
- Software para el desarrollo y unir diferentes componentes de un proyecto

Los *frameworks* permiten:

- Facilitar el desarrollo de software
- Enviar los detalles de bajo nivel, permitiendo conectar más esfuerzo y tiempo en identificar los requerimientos de software.

Para el desarrollo de la solución propuesta se utilizará Odo Mobile como framework de desarrollo, el cual es un framework de código abierto que permite integrar la aplicación Android con Odo. Con la ayuda del framework se puede desarrollar rápidamente casi todas las aplicaciones soportadas por Odo, más rápido que con las que se desarrollan en el propio Odo. Este marco contiene su propio ORM para manejar la base de datos local del móvil (SQLite), encargándose así de los datos provenientes del sistema en Odo.

La arquitectura de Odo Mobile cuenta con cuatro capas las cuales son:

Elementos de la primera capa:

- **Configuración de la aplicación:** Android Manifest y la configuración del Gradle se encuentran en esta capa.

- **Registro de *activities*:** Proporciona una vista la cual posibilita que el usuario pueda interactuar con la aplicación, donde cada actividad tiene una ventana con el objetivo de dibujar su interfaz de usuario.
- ***Fragment*:** Representa un comportamiento o una parte de la interfaz de usuario en una actividad. Donde se puede combinar múltiples fragmentos en una sola actividad para construir una interfaz de usuarios con varios paneles y reutilizar un fragmento en múltiples actividades.
- ***Navigation drawer*:** Es un panel que hace una transición desde el borde izquierdo de la pantalla y muestra las principales opciones de navegación de la aplicación.

Segunda capa: Contiene los modelos de base de datos, cargadores de datos, lógica de negocio para el complemento, cargadores para las listas (listViews), código para los fragmentos y las actividades.

Tercera capa: Permite la actualización con el servidor de Odoo, proporcionando un adaptador base para la sincronización con el framework. Odoo Mobile gestionará los datos offline. Comparará el último registro actualizado y sincronizará los datos locales con los del servidor. El framework trata de dar soporte de modo offline, pero en algunos casos es indispensable anular la sincronización por defecto y crear un mecanismo personalizado de sincronización.

Cuarta capa: Contiene todos los componentes principales utilizado por el framework. El ORM, cuenta con adaptadores de base de sincronización, proveedores de servicios básicos, proveedores de modelos de contenido base y algunas clases de utilidades. Esta capa también es responsable de administrar las cuentas de usuarios y configuraciones de sincronización.

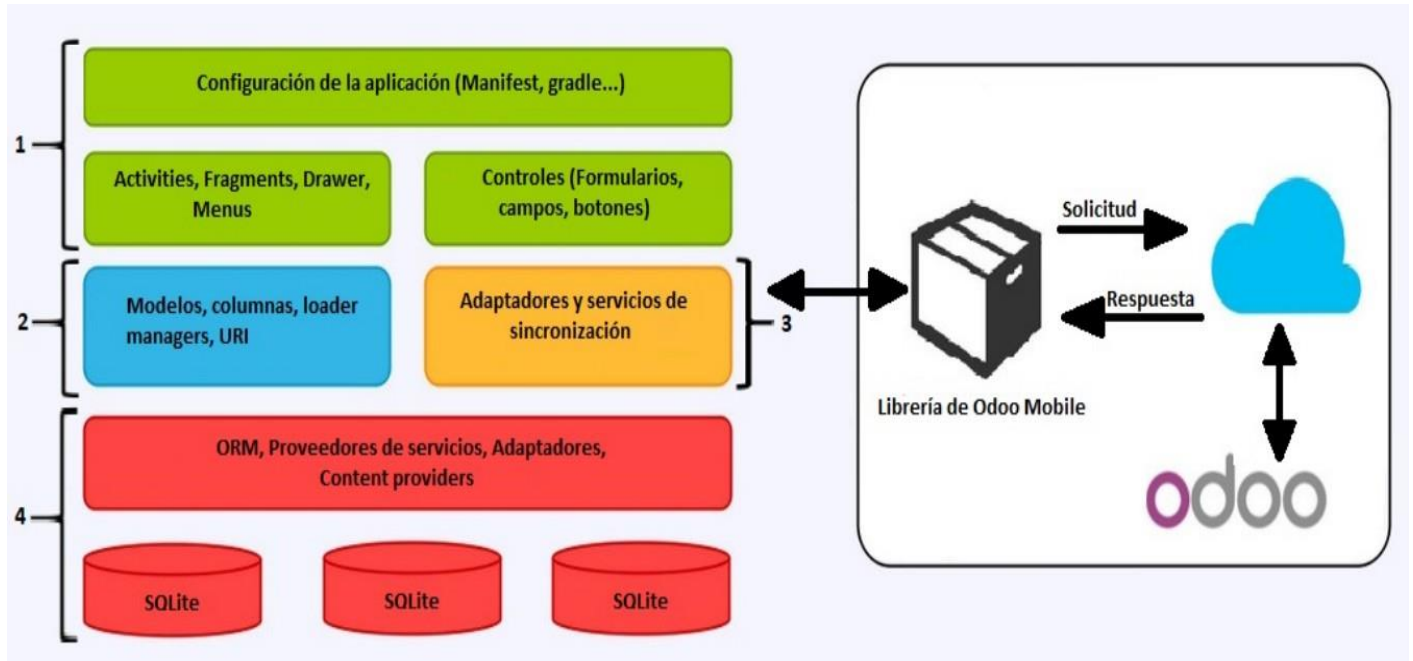


Figura 2: Arquitectura del framework Odoo Mobile

Fuente:(Odoo Mobile 2018)

1.7 Conclusiones parciales

Después de ser analizada la bibliografía especializada, conceptos asociados al dominio del problema y el estudio de soluciones similares, se concluye lo siguiente:

- Partiendo del estudio de las soluciones similares de aplicaciones ERP para móviles a nivel internacional, se pudo afirmar que los mismos no brindan una solución completa al problema existente, no obstante, las aplicaciones ERP móviles tratadas poseen funcionalidades comunes que serán tomadas en cuenta para el desarrollo de la aplicación.
- El estudio de los diferentes tipos de aplicaciones móviles permitió esclarecer que el tipo de aplicación nativa es la mejor opción para obtener un mejor diseño de la propuesta de solución.
- La investigación sobre los tipos de metodologías de desarrollo, demostró que el uso de la metodología de desarrollo AUP- UCI escenario cuatro es la idónea para guiar el proceso de desarrollo de la aplicación.
- El estudio de la herramienta a utilizar para el desarrollo de aplicación móvil demostró la ventaja de la facilidad del IDE Android Studio y el lenguaje Java para el desarrollo de la propuesta de solución.
- El estudio del *framework* de desarrollo a utilizar concreto que Odoo Mobile es la mejor opción para el desarrollo de la aplicación, ya que facilitará la sincronización de la aplicación con el sistema SIGAX.

CAPÍTULO 2. ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se describe la propuesta de solución. En la misma se lleva a cabo la fase de Ejecución establecida por la metodología empleada. Se detallan artefactos asociados al análisis y diseño; especificado, entre otros elementos, requisitos funcionales, no funcionales y patrones de diseño.

2.1 Propuesta de solución

La propuesta de solución tiene como objetivo la confección de una aplicación la cual permita mostrar los datos que aparecen en el sistema SIGAX con una interfaz amigable y almacenar dichos datos en base de datos del dispositivo, el mismo con un sistema operativo Android 4.0 o superior, logrando una correcta visualización de las principales funcionalidades del sistema, lográndose esto a través de la sincronización de la aplicación con el sistema SIGAX y almacenados los datos a la vez. Para llegar a dicha sincronización el dispositivo deberá estar conectado a una red inalámbrica donde esté proporcionando servicio el sistema SIGAX.

2.2 Modelo de dominio

Su utilidad radica en ser una forma de “inspiración” para el diseño de los objetos software, por lo que es el artefacto clave para el análisis orientado a objetos. Sirve como entrada para muchos de los artefactos que se construyen en un proceso software. El modelo de dominio muestra las clases conceptuales significativas en un dominio del problema, centrándose en las abstracciones relevantes, vocabulario del dominio e información del dominio. Por lo que se puede llegar a la conclusión que un modelo de dominio es una representación de las clases conceptuales del mundo real (29).

Definición de las clases

Usuario: Cliente registrado en el sistema SIGAX.

Sistema SIGAX: Sistema de gestión integral desarrollado en el centro FORTES para el Agente Transitario de Cargas TRANSCARGO.

Servicios Web: Servicios web brindados por el sistema SIGAX.

Datos: Conjunto de datos manejados por el sistema gestor.

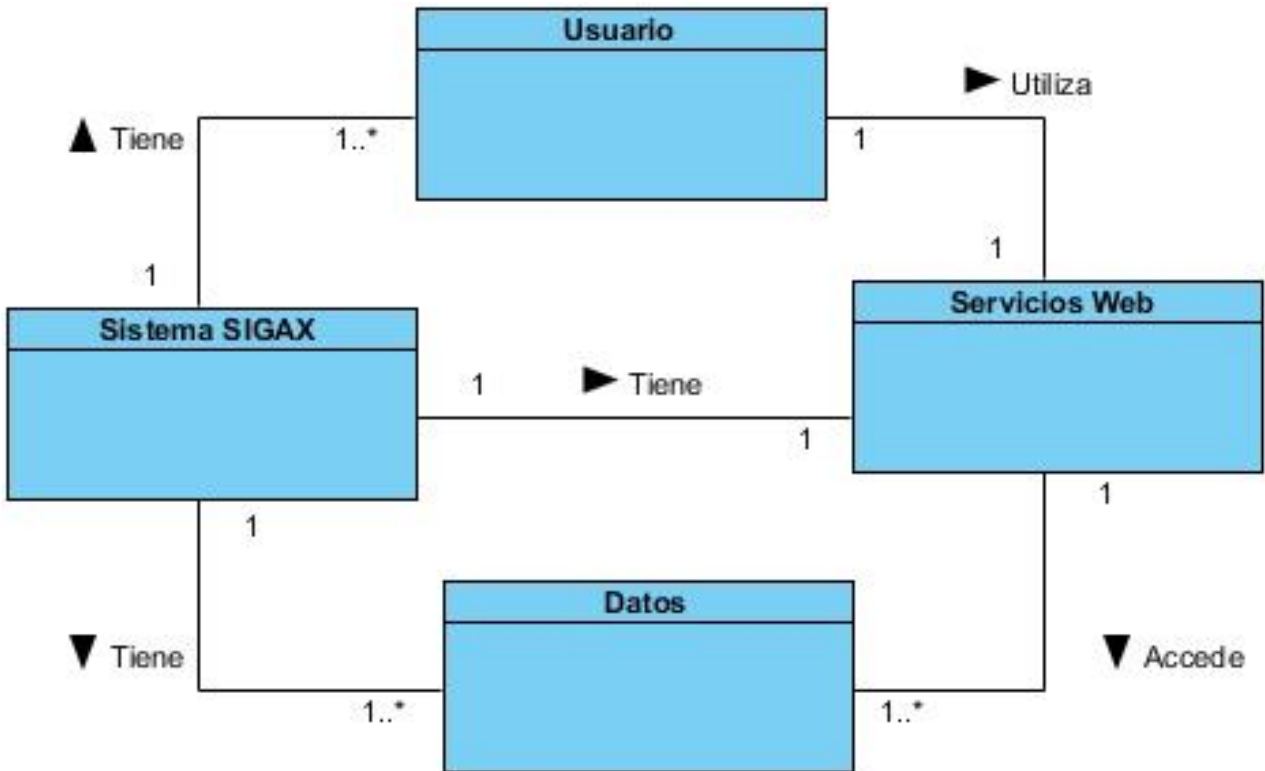


Figura 3. Modelo de dominio

2.3 Requisitos funcionales

Los requerimientos funcionales de una aplicación son las necesidades que se tiene de un producto. Por su parte, los requerimientos no funcionales son las características o condiciones con las que se desea que el sistema cumpla. El objetivo principal de este acápite es identificar los requerimientos funcionales y no funcionales de la propuesta de solución. Los requerimientos funcionales de un sistema describen lo que el sistema debe hacer. Estos requerimientos dependen del tipo de software que se desarrolle, de los posibles usuarios del software y del enfoque tomado por la organización al redactar dichos requerimientos (24). A continuación, se listarán los requisitos funcionales identificados para el desarrollo de la solución propuesta, estos obtenidos en una reunión con el cliente. A cada requisito funcional se le asignó un nivel de importancia o prioridad entre 1 (poco importante) y 5 (muy importante).

Tabla 3: Requisitos funcionales

	Funcionalidad	Importancia
1	Autenticar usuario	5
2	Listar conductores	5

3	Ver detalles de un conductor	5
4	Realizar busca rápida de conductores	2
5	Filtrar conductores	3
6	Listar vehículos	5
7	Ver detalles de un vehículo	5
8	Realizar búsqueda rápida de vehículos	2
9	Filtrar vehículos	3
10	Listar multas	5
11	Ver detalles de una multa	5
12	Realizar búsqueda rápida de multas	2
13	Filtrar multas	3
14	Listar contenedores disponibles	5
15	Ver detalles de un contenedor disponible	5
16	Realizar búsqueda rápida de contenedores disponibles	2
17	Filtrar contenedores disponibles	3
18	Listar contenedores arrendados	5
19	Ver detalles de un contenedor arrendado	5
20	Realizar búsqueda rápida de contenedores arrendados	2
21	Filtrar contenedores arrendados	3
22	Listar contenedores fuera de servicio	5
23	Ver detalles de un contenedor fuera de servicio	5
24	Realizar búsqueda rápida de contenedores fuera de servicio	2

25	Filtrar contenedores fuera de servicios	3
26	Listar declaraciones de mercancía	5
27	Ver detalles de una declaración de mercancía	5
28	Realizar búsqueda rápida de declaraciones de mercancías	2
29	Filtrar declaraciones de mercancías	3
30	Listar órdenes de pago	5
31	Ver detalles de un orden de pago	5
32	Realizar búsqueda rápida de órdenes de pago	2
33	Filtrar órdenes de pago	3
34	Listar facturaciones	5
35	Ver detalles de una facturación	5
36	Realizar búsqueda rápida de facturaciones	2
37	Filtrar facturaciones	3
38	Listar compradores	5
39	Realizar búsqueda rápida de compradores	2
40	Filtrar compradores	3
41	Listar contenedores de logística	5
42	Ver detalles de un contenedor de logística	5
42	Realizar búsqueda rápida de contenedores de logística	2
43	Filtrar contenedores de logística	3
44	Listar reportes de incidencias	5
45	Ver detalles de un reporte de incidencias	5
46	Realizar búsqueda rápida de reportes de	2

	incidencias	
47	Filtrar reportes de incidencias	3

2.4 Requisitos no funcionales

Los requerimientos no funcionales, como su nombre sugieren, son aquellos requerimientos que no se refieren a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, define las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema (24).

Tabla 4: Prestaciones mínimas de hardware y software de las terminales móviles donde se utilizará

Recurso	Numeración	Descripción
Hardware	RNF1	Se deberá disponer de algún dispositivo móvil, ya sea <i>Tablet</i> o celular, con al menos 4.0 pulgadas de pantalla
	RNF2	El dispositivo móvil debe tener 50 megabytes de espacio libre para la instalación de la aplicación y 100 megabytes para datos
Software	RNF3	El dispositivo móvil que se disponga deberá tener un sistema operativo Android con versión 4.0 o superior
Interfaz	RNF4	Para comodidad del usuario aplicación se utilizará en idioma español

2.5 Historias de usuarios

Una correcta descripción de los requisitos permitirá contar con un punto de vista más detallado en el momento de definir la arquitectura del sistema, estableciendo un punto de partida para las pruebas de funcionamiento y las futuras mejoras a incorporar. Las Historias de Usuario (HU) son el artefacto establecido por la metodología y el escenario que se emplea para la especificación o descripción de funcionalidades. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores

puedan implementarla en el tiempo estimado (30). Al realizar la discusión y el análisis de un requisito determinado se procede a realizar un prototipo de la interfaz de usuario a manera de apoyo gráfico, su objetivo será asegurar el total entendimiento por parte de los desarrolladores del requisito en cuestión.

A continuación, se muestra la historia de usuario para el requisito “Listar conductores”, el resto de historias se reflejarán en el Anexo 1.

HU_ 1: Listar conductores.

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
1	Nuevo	4	4	2	1	5
Nombre						
Listar Conductores						
Notas						
Descripción						
El objetivo de esta actividad es listar los conductores existentes en el sistema. Para ello el usuario seleccionará la opción “Conductor” del menú y se desplegará el listado con los datos más descriptivos de los conductores.						
Prototipo de Interfaz						
<p>El prototipo de interfaz de usuario para un smartphone muestra una pantalla con el título "Conductores". En la parte superior izquierda hay un ícono de menú lateral. La pantalla muestra una lista de conductores con los siguientes datos:</p> <ul style="list-style-type: none"> Licencia: g6t65665676, Nombre: Randy Muñoz Nuñez, CI: 94051729786 Licencia: k897877787, Nombre: Pepe Nillar, CI: 1111111111 Licencia: j3838393393, Nombre: Sandy Perez, CI: 94051793883 <p>El prototipo también incluye un botón flotante de búsqueda en la parte inferior derecha.</p>						

2.6 Modelo de datos

Diseño de una base de datos

El diseño de una base de datos es el proceso por cual se determina la organización de una base de datos, incluyendo su estructura, contenido y las aplicaciones que se han de desarrollar. Debido a la creciente aceptación de las bases de datos por parte de la industria y el gobierno en el plano comercial, el diseño de bases desempeña un papel central en el empleo de los recursos de datos en la mayoría de las organizaciones. El diseño de bases de datos ha pasado a constituir parte de la formación general de los informáticos, en el nivel que la capacidad de construir algoritmos usando un lenguaje de programación convencional (31).

Modelo de datos Entidad-Relación

Es una serie de conceptos que puede utilizarse para describir un conjunto de datos y las operaciones para manipularlos. Denominadas por sus siglas E-R, el modelo de entidad – relación representa a la realidad a través de las entidades, que son objetos que se existen y se distinguen de otros por sus características. Las características de las entidades en la base de datos se llaman atributos, por ejemplo, el nombre, dirección, carnet de identidad, identificador, entre otros; estos son atributos de la entidad empleada. A su vez una entidad se puede relacionar o asociar con más entidades a través de relaciones (31).

La aplicación que se propone desarrollar hace uso de la base datos del sistema SIGAX, debido a que la información que se pretende consultar y mostrar se encuentra en la misma. A continuación, se muestra una parte del modelo de base de datos del sistema SIGAX, mostrando solo las entidades y atributos que son relevantes para el desarrollo del sistema propuesto.

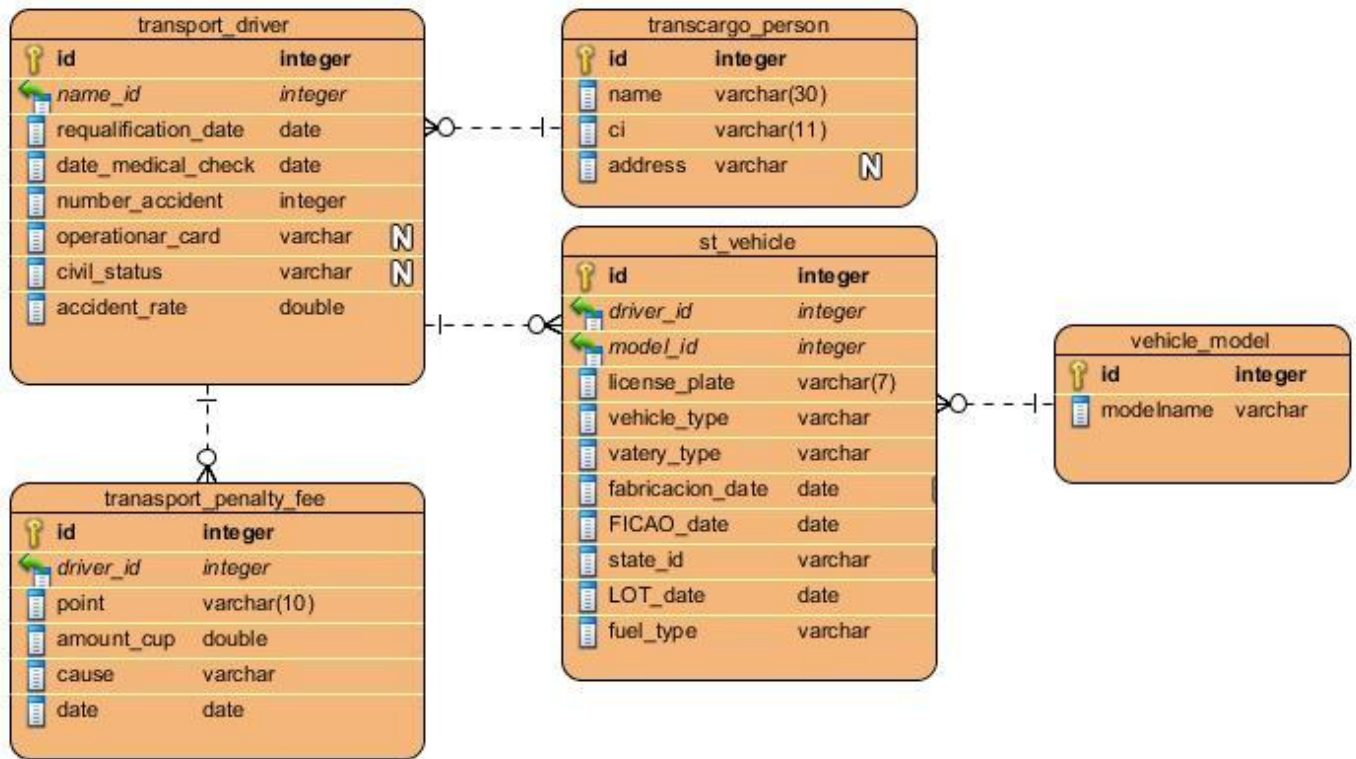


Figura 3: Modelo de datos (Entidades conductor, multa y persona)

Fuente: Elaboración propia

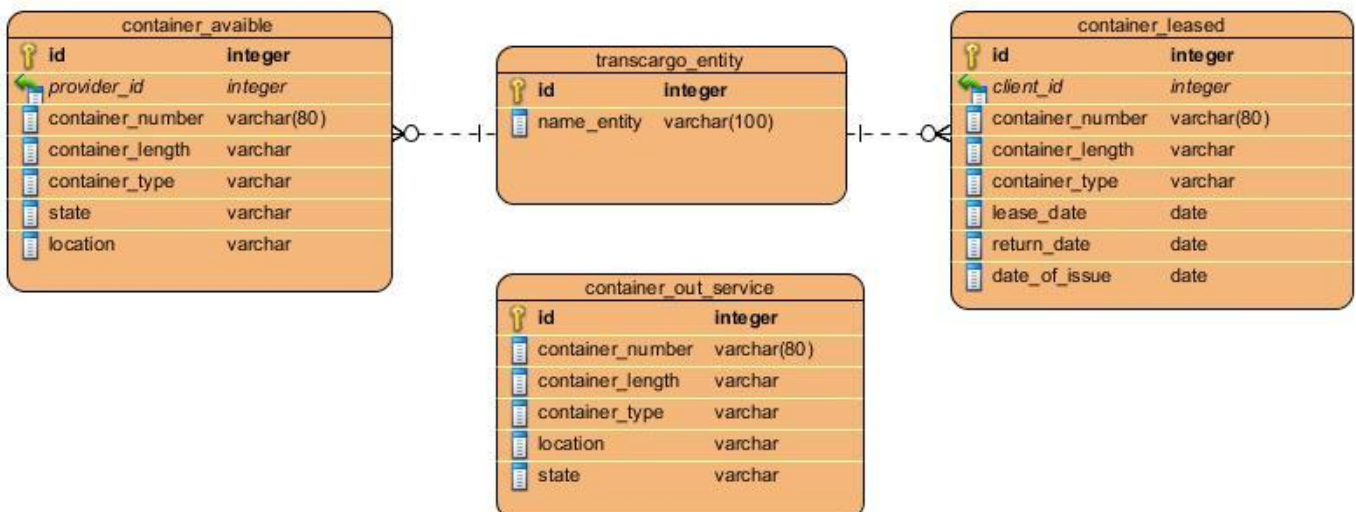


Figura 4: Modelo de datos (Entidad contenedor disponible, contenedor arrendado y contenedor fuera de servicio)

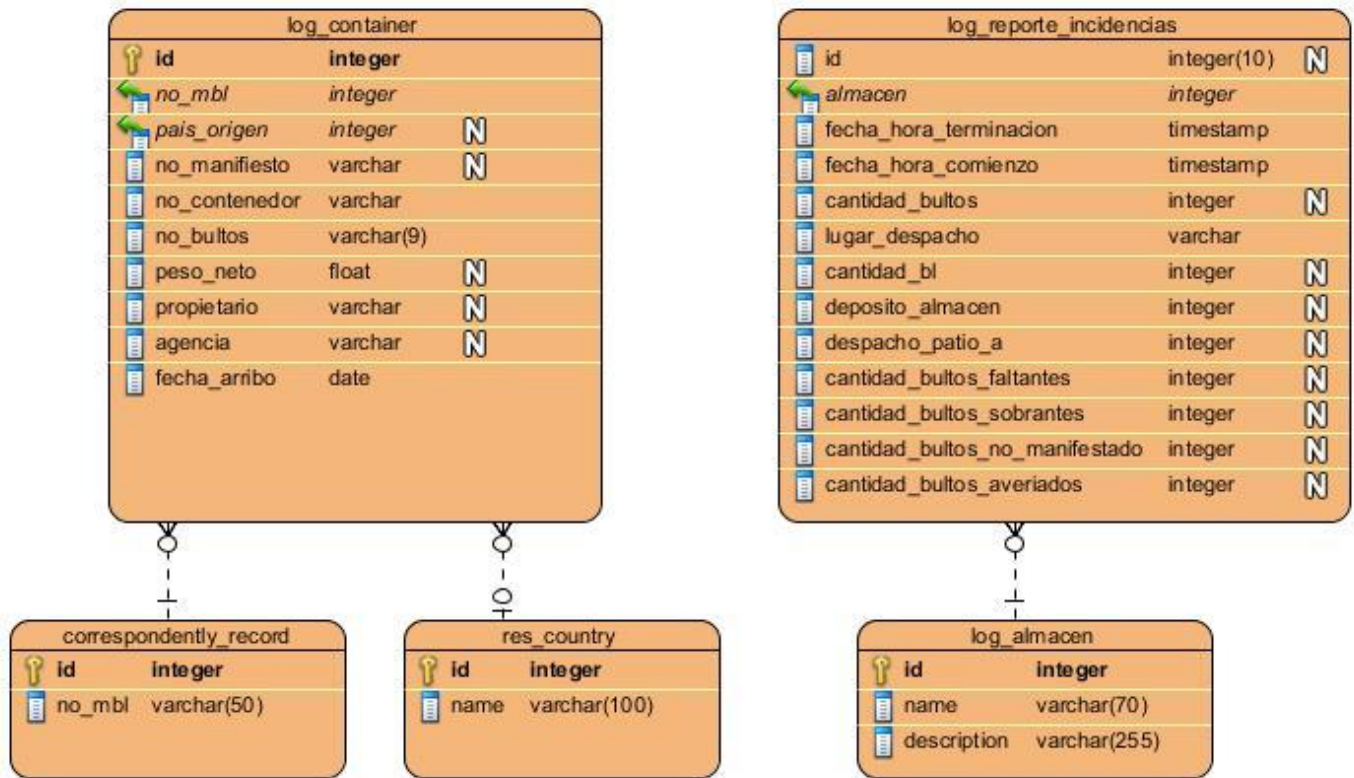


Figura 5: Modelo de datos (Entidades de contenedores y de reporte de incidencias)

2.7 Patrón Arquitectónico

Descripción abstracta de una arquitectura de software que se ensayó y puso a prueba en algunos sistemas de software distintos. La descripción del patrón incluye información acerca de dónde es adecuado usar el patrón y la organización de los componentes de la arquitectura (24).

Modelo-vista-controlador

Separa presentación e interacción de los datos del sistema. El sistema se estructura en tres componentes lógicos que interactúan entre sí. El componente Modelo maneja los datos del sistema y las operaciones asociadas a esos datos. El componente Vista define y gestiona como se presentan los datos al usuario. El componente Controlador dirige la interacción del usuario y pasa estas interacciones a Vista y Modelo (24). La solución propuesta constará de tres capas. Dentro de la capa Vista se encontrarán los archivos visuales con los que el usuario interactuará directamente, introduciendo las configuraciones pertinentes y resolviendo sus necesidades, esto se logra a través de los *layouts* (Lenguaje XML en Android). En la capa Modelo se tendrán las clases que interactúan con las tablas de base de datos. La capa Controlador está diseñada para poder responder a los distintos eventos originados por los objetos de la vista y gestionar el flujo de datos entre los objetos del modelo de datos y la capa de visualización.

2.8 Patrones de diseños

Se podría definir patrón como una solución a un problema en un contexto. Los patrones ayudan a capturar conocimiento y a crear un vocabulario técnico, hacen el diseño orientado a objetos más flexible, elegante y en algunos casos reusable, son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Proporcionan una estructura conocida por todos los programadores, de manera que la forma de trabajar no resulte distinta entre los mismos. Un *framework* simplifica el desarrollo de una aplicación mediante la automatización de algunos patrones utilizados para resolver las tareas comunes. Los patrones de diseño son modelos que proporcionan respuestas a un conjunto de problemas similares. (32).

2.8.1 Patrones GRAPS

General Responsibility Assignment Software Patterns (GRASP) nos da unos principios generales para asignar responsabilidades a objetos software y se utiliza sobre todo en la realización de diagramas de interacción en la programación orientada a objetos. Básicamente las responsabilidades de un objeto son conocer y hacer. Dichas responsabilidades implicarán más o menos métodos y clases (33). En la aplicación se evidencia el uso de los patrones: Patrón Alta cohesión, Patrón Bajo acoplamiento y Patrón creador, los cuales se describen a continuación.

Patrón Alta cohesión: Este patrón define lo relacionadas que están las responsabilidades de una clase, o una clase con responsabilidades altamente relacionadas y que no lleva a cabo gran cantidad de trabajo (33). Para el desarrollo de la propuesta de solución fueron creados varios fragments encargados de trabajar con un módulo específico del sistema; de esta manera se evita que una clase sea la única responsable de muchas tareas en áreas funcionales muy heterogéneas. La clase "ContainerList" es ejemplo de la utilización de este patrón. Es la encargada de listar los elementos de la entidad Contenedor (*Container*), permitiendo la sincronización en tiempo real con la base de datos y la realización de búsquedas sobre el listado.

Patrón Bajo acoplamiento: Este patrón se utiliza con el objetivo de disminuir la dependencia entre clases e incrementar la reutilización de código. Acoplamiento indica que tan fuertemente está conectada una clase con otra, tiene conocimiento de, o influye sobre otra clase, una clase con bajo acoplamiento no depende de otras clases. Esto se logra asignando responsabilidades de tal manera que el acoplamiento sea el menor posible, logrando mejor comprensión de las clases aisladas, facilitando la reutilización del código y no afectan los cambios en otros componentes (33). El diseño propuesto respeta este patrón al evidenciar clases con el menor número de clases relacionadas posible, ejemplo de ello son los servicios de sincronización. La clase "CorrespondentlyRecordSyncService" se encarga, solamente, de realizar la sincronización con la base de datos de la entidad Expediente (Record). La creación de estas clases independientes provoca que

los cambios realizados en otros servicios de sincronización no influyan en el funcionamiento de un servicio específico.

Patrón Creador: Responsable de crear una nueva instancia de una clase, donde la clase B tiene la responsabilidad de crear una instancia de la clase A si:

- B agregación objetivos de A.
- B contiene objetos de A.
- B usa objetos de A.
- B tiene los datos necesarios para inicializar A cuando este es creado.

Esto conlleva a que soporta el bajo acoplamiento, es decir, este no se incrementa. Este patrón se ve reflejado en la clase “AuthorizeList”, la cual tiene la responsabilidad de crear una instancia de la clase “Authorize” con el objetivo de acceder a sus atributos y mostrarlos en una lista.

2.8.2 Patrones GoF

Patrones publicados por Gamma, Helm, Johnson y Vlossodes en 1995: patrones de la pandilla de los cuatro (gang of four). Esta serie de patrones permiten ampliar el lenguaje, aprender nuevos estilos de diseño y además se introducirá más notación UML. Existen 23 patrones GoF de los que 15 se utilizan con frecuencia. Se clasifican según su propósito en los patrones: de creación (Creación de objetos), estructurales (Composición de clases y objetos) y de comportamiento (Modo en que las clases y objetos interactúan y se reparten las responsabilidades) (33). En la aplicación se evidencian los patrones de Singleton y el Templated Method, los cuales se describen a continuación.

Singleton: Este patrón asegura que una clase tenga una única instancia y que esta sea fácilmente accesible. Esto se logra definiendo un método estático de la clase que devuelve el singleton, permitiendo el manejo de objetos únicos y que sean accesibles por otros objetos (33). El patrón se ve reflejado en la clase “OdoAccountManager” la cual maneja la información relacionada con el usuario activo en la aplicación. Con este patrón la clase será la responsable de crear su propia instancia única la cual es devuelta con el método “getActiveUser”.

Templated Method: Este patrón se ve reflejado en la clase “Query” la cual posee el método “generateQuery”; este método se encarga de recibir una lista de condiciones y generar una consulta SQL para realizar la búsqueda filtrada de elementos. La implementación del método “generateQuery” varía en cada una de las subclases de “Query” (“DriverQuery”, “PenaltyFeeQuery”, “VehicleQuery” entre otras) por lo que este patrón tiene como objetivo permitir que las subclases redefinan ciertos pasos del método sin cambiar su estructura, por lo que se evita que exista código duplicado en la aplicación.

2.8.3 Diagrama de clases

Los diagramas de clases representan las relaciones existentes entre las clases de un sistema. Su empleo consiste en modelar la vista estática del diseño de un sistema, especificar, visualizar, especificar y documentar modelos estructurales, y para construir sistemas ejecutables a través de ingeniería directa o inversa (34).

La estructura que guiará el desarrollo de la aplicación para la consulta y almacenamiento de datos a la aplicación SIGAX en sistemas operativos Android tendrá como base el siguiente diagrama de clases, la cual representa uno de los módulos del sistema:

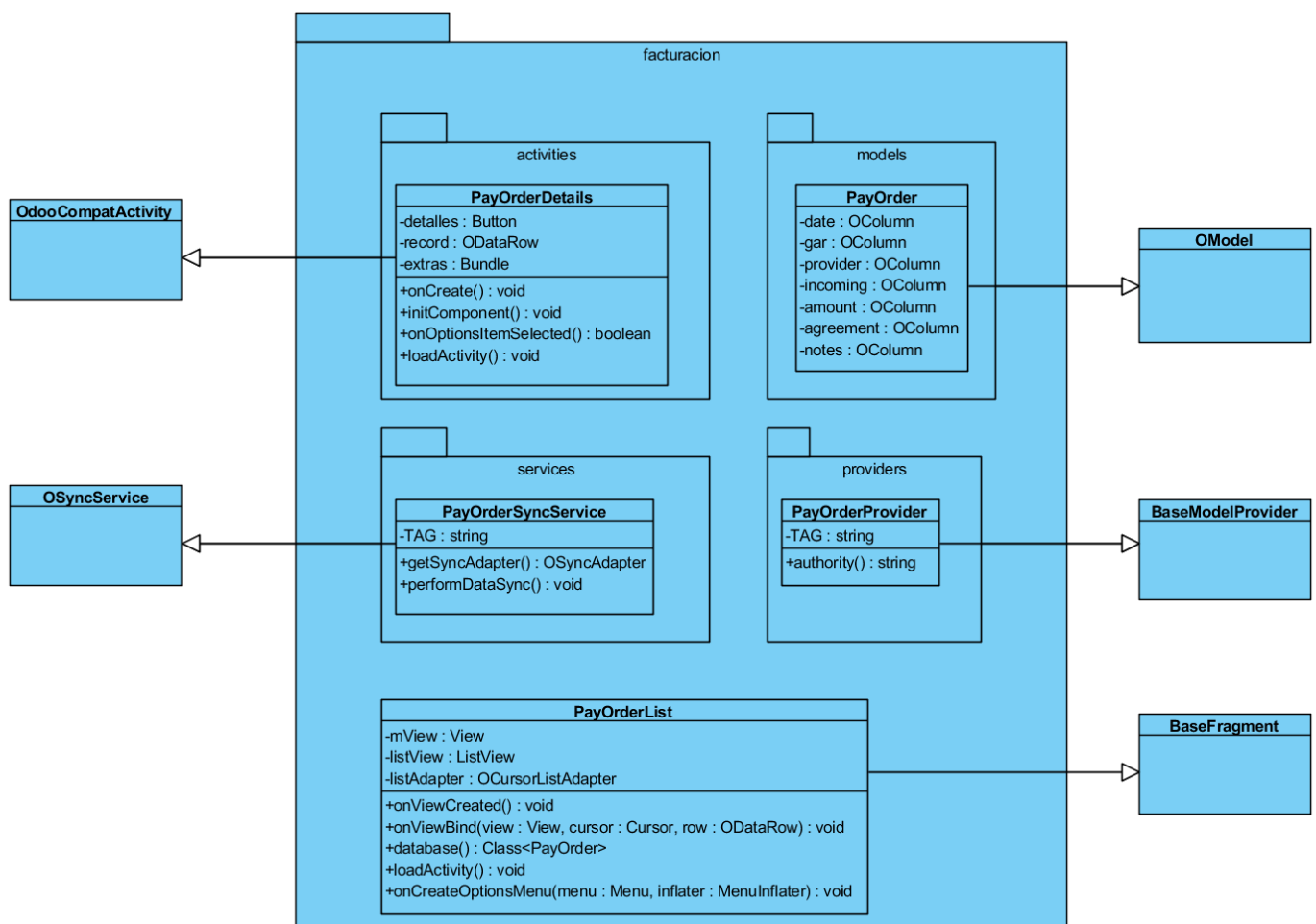


Figura 6: Diagrama de clases del módulo facturación.

2.9 Conclusiones parciales

Después de circular por los periodos de exploración e inicialización del proyecto referidas en la metodología AUP-UCI se concluye lo siguiente:

- Con la representación modelo de dominio se logró llegar a comprender las diferentes clases que componen al sistema.
- La definición de las historias de usuario, juntas a sus respectivas interfaces, dio la posibilidad de comprender los requisitos funcionales.
- La definición de la arquitectura basada en Modelo-Vista-Controlador junto al uso de patrones de diseño garantizó un mejor entendimiento de la solución propuesta, donde se sentaron las bases para su implementación.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

En el presente capítulo se exponen aspectos asociados a la implementación de la aplicación, en el cual se propone: definir la organización del código, implementar las clases y objetos en forma de componentes, probar los componentes desarrollados e integrarlos a un sistema ejecutable. Se registran los resultados obtenidos de la aplicación de pruebas al sistema, con el fin de validar el cumplimiento de los requerimientos establecidos. Se presentan las pautas de codificación y los resultados obtenidos de las pruebas en busca de errores en la aplicación.

Una vez definida las HU y concluido el diseño de la aplicación tiene lugar a la solución propuesta, cuyos objetivos están encaminados a desarrollar de manera iterativa e incremental un producto completo y listo para despliegue.

3.1 Modelo de implementación

El modelo de implementación como los elementos del modelo de diseño, se implementan en términos de componentes, como ficheros de código fuente y ejecutables. El modelo describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación utilizado, y como dependen los componentes uno de los otros (35).

3.1.1 Pautas de codificación

Los estándares de codificación garantizan la comunicación entre los programadores, permitiendo la reutilización y mantenimiento de los sistemas. Durante el proceso de desarrollo de la propuesta de solución se emplea el estándar *CamelCase*² para la definición de variables y métodos con el fin de lograr un buen entendimiento del código, se divide en los estándares *lowerCamelCase*³ y *UpperCamelCase*⁴.

A continuación, se describe los estándares de codificación que registran la implementación de las funcionalidades.

- **Clases:** Los nombres de las clases deben ser sustantivos, la primera letra del nombre debe empezar con mayúscula y las demás letras con minúsculas excepto la inicial de las siguientes palabras, se escriben con mayúscula.

² *CamelCase*: Estándar de codificación que define el comienzo con la letra minúscula y las siguientes iniciales de palabras con mayúsculas.

³ *lowerCamelCase*: Estándar de codificación *CamelCase* que comienza con la letra minúscula y el resto de las palabras las comienzan con mayúscula.

⁴ *UpperCamelCase*: Estándar de codificación *CamelCase* en el que todas las palabras las comienzan con la letra mayúscula.

- **Paquetes:** Los nombres de los paquetes se escriben con minúsculas y en caso de tener más de una palabra se dividirán con un guion bajo entre las palabras.
- **Métodos:** Los nombres de los métodos deben ser verbos y de carácter obligatorio comenzar con mayúsculas, en caso de tener más palabras el nombre, la primera letra de las demás palabras deberán comenzar con mayúscula.
- **Sentencias:** Se debe tener una sentencia por línea de código. Las sentencias deben estar entre signos de llaves, aunque sea una sola sentencia. En caso de que una línea de código sea demasiado grande se dividirá en varias líneas después de una coma u operador.
- **Variables:** Deben comenzar con una letra minúscula, al cambiar la palabra se usará letra mayúscula. Los nombres de las variables no deben comenzar con el carácter bajo o con el signo de peso, aunque el lenguaje Java lo permite.
- **Comentarios en las funciones:** Las funciones deben tener un comentario antes de ser declaradas, explicando el funcionamiento de las mismas.

3.1.2 Diagrama de Componentes

Un diagrama de componentes modela la vista estática del sistema y son usados para estructurar el modelo de implementación en términos de subsistema de implementación. Muestra la organización y dependencias entre un conjunto de componentes. A continuación, serán expuestos los diferentes usos del diagrama de componentes:

- Muestra relaciones de derivación entre ficheros de código fuente y ficheros que son resultados de compilación.
- Muestra la estructura de alto nivel del modelo de implementación, especificando los subsistemas de implementación y sus dependencias.

A continuación, se muestra el Diagrama de Componentes del sistema:

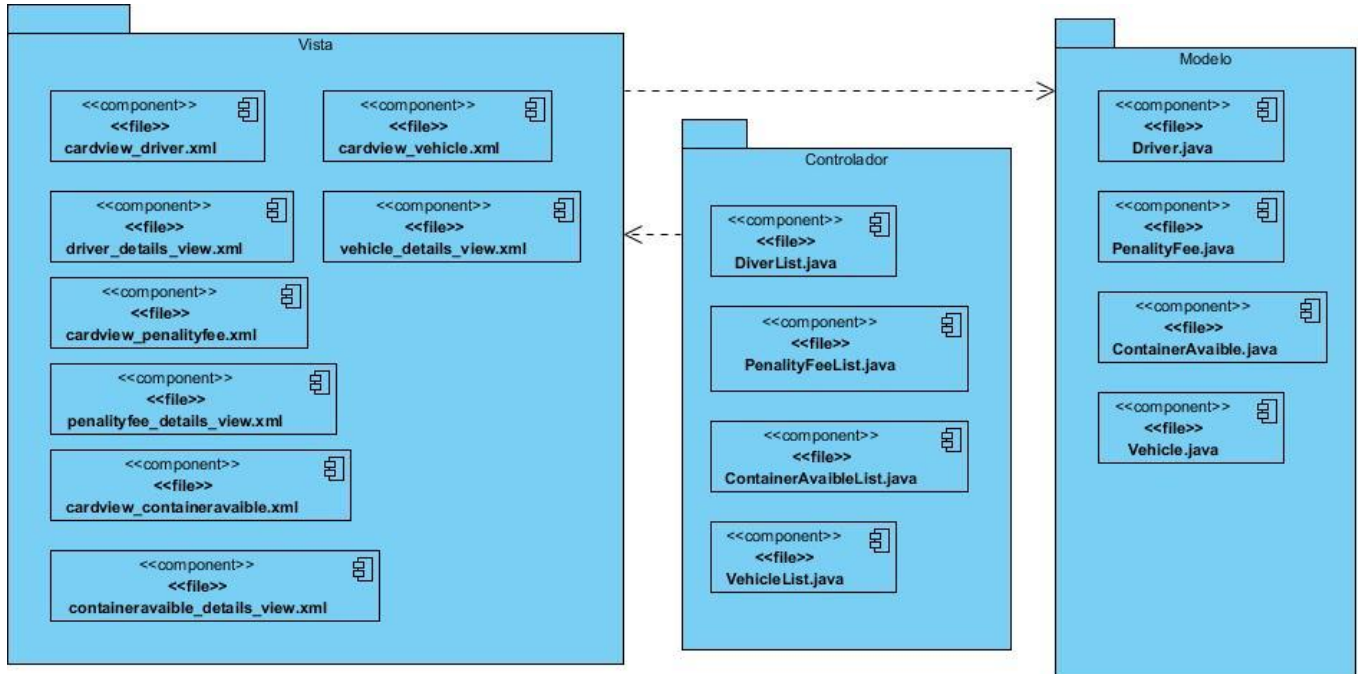


Figura 7: Diagrama de componentes

3.2 Pruebas y validación del software

Una forma adecuada para determinar la calidad de un software es el proceso de pruebas. En este proceso se establecen pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con la finalidad de medir el grado con el que cumple con los requerimientos. Además, las pruebas son capaces de detectar los defectos del software y de verificar la integración correcta de los componentes.

La metodología empleada establece la realización de una serie de pruebas. De ellas se aplicarán las pruebas unitarias y de aceptación a la solución implementada.

3.2.1 Pruebas funcionales

Las pruebas funcionales tienen como objetivo validar cuando el comportamiento observado del *software* probado cumple o no con sus especificaciones. Lo que aseguran el apropiado trabajo de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados. Las metas de estas pruebas son verificar apropiada aceptación de datos y verificar el procesamiento, recuperación e implementación adecuada de las reglas del negocio (36).

Para la realización de las pruebas de aceptación se utilizó el método de caja negra. Este método se centra principalmente en los requisitos funcionales del software. La misma no es una alternativa al método de caja blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados utilizando el método de caja blanca. El método de caja negra permite obtener un conjunto de condiciones

de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En este método se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos. Muchos autores consideran que este método permite encontrar (38):

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

3.2.2 Resultados de las pruebas funcionales

Luego de haber realizado las pruebas definidas para la aplicación fueron obtenidas un conjunto de no conformidades, en adelante (NC). Estas NC fueron clasificadas de acuerdo con su prioridad en altas, medias y bajas.

Las pruebas funcionales se realizaron en 4 iteraciones, en la primera iteración se detectaron 8 NC (2 altas, 2 medias y 4 bajas). En la segunda iteración se detectaron 5 NC (1 alta, 1 medias y 3 bajas). En la tercera iteración se detectó 1 NC (1 baja). Para la cuarta iteración no se detectó NC. Los datos anteriormente expuestos aparecen reflejados en el gráfico siguiente:



Figure 8: Pruebas funcionales

3.2.3 Pruebas de aceptación

Las pruebas de aceptación comprueban los errores a través del análisis de los requisitos, proponen realizar numerosas operaciones sobre la interfaz visual del software con el fin de saber si se cumplen todos los requisitos (37).

Los casos de pruebas de aceptación pretenden demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma correcta.
- Se produce una salida correcta.
- La integridad de la información interna se mantiene.

Se pretende encontrar con este tipo de pruebas los siguientes errores:

- Errores en la interfaz.
- Errores en estructuras de datos o en acceso a base de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación

Según la documentación oficial de la metodología de desarrollo para la actividad productiva de la UCI, el objetivo de esta prueba es verificar que el software esté listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software es construido.

Las pruebas de aceptación se realizaron usando el método de caja negra. Durante la etapa de prueba se realizan la recolección de los resultados de las pruebas definidas en la documentación y su ejecución para obtener posibles errores. Cada caso de prueba fue representado en forma de tabla con los siguientes campos: Número y nombre de la historia a que pertenece, fecha en que fue creado, fecha en que se ejecutó, si pasó la prueba, entradas y resultados esperados. En adelante, se presenta una de las pruebas de aceptación generadas para la historia de usuario "Listar conductores".

Tabla 5: Prueba de aceptación para la historia "Listar conductores"

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO:	HISTORIA DE USUARIO: HU_5
DESCRIPCIÓN: prueba la funcionalidad de listar conductores.	

<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un conductor. • El usuario debe poseer los permisos de transporte.
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Una vez autenticado el usuario se selecciona la opción “Conductores”</p>
<p>RESULTADO ESPERADO:</p> <p>Aparece un listado con todos los conductores registrados en el sistema. En el listado se deben mostrar los siguientes datos:</p> <ul style="list-style-type: none"> • Licencia • Nombre y apellidos • CI (Carnet de identidad) <p>De no existir ningún conductor en el sistema se mostrará un mensaje.</p>
<p>HEVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

3.2.4 Resultados de las pruebas de aceptación

Después de haber realizado las pruebas definidas para la aplicación fueron obtenidas un conjunto de NC. Estas NC fueron clasificadas de acuerdo con su prioridad en altas, medias y bajas.

En las pruebas de aceptación que se realizaron fueron ejecutadas un total de 47 casos de pruebas. En la primera iteración se detectaron 11 NC (3 altas, 4 medias y 4 bajas). En la segunda iteración se detectaron 5 NC (1 alta, 2 medias y 2 bajas). En la tercera iteración se detectó 1 NC (1 media). Para la cuarta iteración no se detectó NC. Los datos anteriormente expuestos aparecen reflejados en el gráfico siguiente:

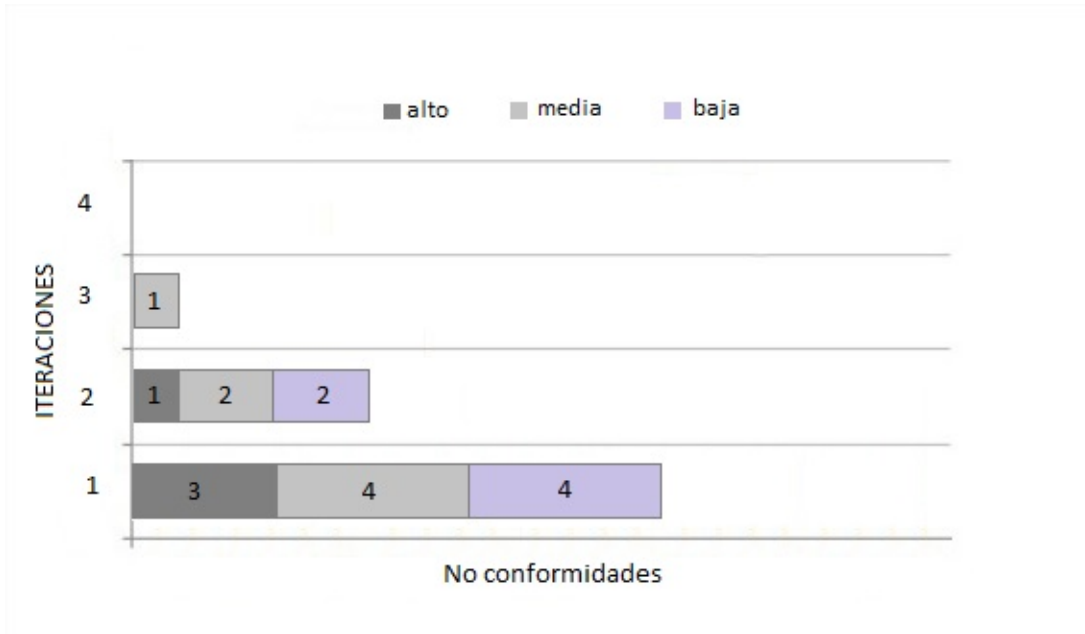


Figura 9: Pruebas de aceptación

3.2.5 Pruebas unitarias

Las pruebas unitarias se concentran en la lógica del procesamiento interno y en las estructuras de datos dentro de los límites de un componente. Se enfocan además en el esfuerzo de la verificación de las unidades más pequeñas del diseño de software, las cuales son los componentes o los módulos (39). Desarrollar estas pruebas permite garantizar que las operaciones se ajustan a las especificaciones y que todos los componentes se han probado de forma adecuada.

En las pruebas unitarias se realiza un examen minucioso de los detalles procedimentales, comprobando los caminos lógicos del programa, comprobando los bucles y condiciones, y comprobar el estado del programa en varios puntos. Es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar los casos de pruebas (37).

Garantizan que:

- Se ejecutan al menos una vez todos los caminos independientes de cada módulo.
- Se utilizan sus decisiones en su parte verdadera y en su parte falsa.
- Se utilizan todas las estructuras de datos internas.
- Se ejecutan todos sus bucles en sus límites.

El método utilizado para las pruebas unitarias fue el método de caja blanca (También conocido como métodos de caja de cristal o pruebas estructurales) se centran en métodos procedimentales del *software*,

por lo que su diseño está fuertemente ligado al código fuente. Mediante al método de caja blanca el desarrollador del *software* puede obtener casos de prueba que (40):

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Para la realización de las pruebas unitarias a la aplicación se utilizó fue utilizado el *framework* JUnit en la versión 4.12 y el *framework* Mockito en su versión 2.8.9.

JUnit es una herramienta simple, de código abierto que se ha transformado en el estándar para probar clases en Java de forma unitaria al ser fácil de configurar y utilizar. Esta herramienta ejecuta casos de pruebas contra un conjunto de objetos y provee formas de inicializar y configurar cada caso. Al integrar datos de entrada y comparar con los resultados esperados, JUnit devolverá si un método determinado pasó la prueba exitosamente o mostrará un mensaje de error (41).

Mockito es un *framework* de simulación de pruebas unitarias para Java. El cual permite la simulación de la creación y verificación de objetos y dependencias externas. Utiliza para ello la técnica conocida como *mocking*. El *mocking* es una técnica utilizada en las pruebas de software donde los componentes reales son reemplazados por objetos que poseen un comportamiento predefinido solamente para la prueba, de esta forma se configuran los objetos para que devuelvan un valor específico sin ejecutar ninguna acción (42).

Estas pruebas unitarias fueron realizadas durante todo el proceso de implementación de las funcionalidades. A continuación, se muestra dos ejemplos de pruebas unitarias realizadas a la clase “DriverQuery”, centrándose en el método “getQuery”:

```

4  ...
5
6  @Test
7  public void paramsHasThreeElements() {
8      String param = "param1";
9      Pair<String, List<String> > datos = driver.getQuery(param);
10     Assert.assertEquals(3, datos.second.size());
11 }
12
13 ...

```

```
4 ...
5
6 @Test
7 public void testQueryIfNull() {
8     Pair<String, List<String> > datos = driver.getQuery(null);
9     Assert.assertEquals("", datos.first);
10 }
11
12 ...|
13
```

3.3 Conclusiones parciales

Luego de transitar por las fases de producto, estabilización y prueba-corrección definidas en la metodología AUP, se concluye lo siguiente:

- El establecimiento de las pautas de codificación permitió el desarrollo de un código reutilizable, legible y específico para el programador.
- El uso vinculado de las pruebas unitarias y de aceptación permitió identificar las no conformidades en la propuesta de solución, realizando 66 pruebas de aceptación y 20 unitarias.
- Gracias al proceso llevado a cabo y las pruebas realizadas al sistema permitió validar y comprobar que la aplicación cumple con los requisitos funcionales presentados para la implementación de la misma.

Conclusiones

Entre los principales resultados obtenidos durante la investigación y desarrollo del producto final se puede destacar que:

- El estudio de las soluciones similares, en específico, las aplicaciones ERP móviles demostró la ausencia de las soluciones adaptables al problema de la investigación, sin embargo, ayudó a identificar las posibles funcionalidades a implementar en la propuesta de solución.
- El empleo de las herramientas, metodologías y tecnologías seleccionadas permitieron analizar y describir la solución propuesta: una aplicación móvil nativa para dispositivos móviles con el sistema operativo Android utilizando el *framework* de desarrollo Odo Mobile.
- La implementación de la aplicación móvil del sistema SIGAX, luego de ser validada a partir de las pruebas definidas, permitió el correcto funcionamiento de la visualización de los datos del sistema SIGAX, a través de la sincronización con la aplicación, logrando a su vez guardar dichos datos y dando la posibilidad de consultarlos de manera *offline*.

Recomendaciones

Para perfeccionar la solución propuesta se realizan las siguientes recomendaciones:

- Actualizar las funcionalidades de la solución a medida que se realicen actualizaciones en el sistema SIGAX.
- Implementar otras funcionalidades con la que cuenta el sistema SIGAX como: Crear, Eliminar y Editar.

Referencias Bibliográficas

1. **Ramón, Costa.** *El uso de las TIC en las organizaciones.* 2012.
2. **Rueda Cruz, Andrea Araceli.** *Planificación de Recursos Empresariales (ERP).* 2017.
3. **Prieto Blázquez, Josep.** *Introducción a los sistemas de comunicación inalámbricos.* 2014.
4. **Merchán Riera , Jorge Misael , y otros.** *Impacto de las Aplicaciones Móviles en la Gestión Empresarial en Latinoamérica.* 2017.
5. **Muñiz, Luis.** *ERP, Guía práctica para la Selección e Implementación.* 2004.
6. **TRANSCARGO.** *Sitio oficial de TRANSCARGO.* [En línea] 2018. www.transcarga.transnet.cu.
7. **Daniel y Sol Llaven, Daniel.** *Android.* s.l. : Patria, 2015.
8. **Banz Alonso, Arturo, y otros.** *Dispositivos móviles.* 2011.
9. **Santos Macías, Francisco Javier.** *Tecnologías inalámbricas para la comunicación.* 2009.
10. **Software de Gestión - a3 | Wolters Kluwer.** [En línea] 2017. [Citado el: 27 de 11 de 2017.] <https://a3.wolterskluwer.es/>.
11. **plc, Sage Group.** *Enterprise Management.* [En línea] 2018. [Citado el: 3 de 2 de 2018.] <http://www.sagex3.com>.
12. **Odoo.** *Odoo.* [En línea] 2017. [Citado el: 15 de 11 de 2017.] https://www.odoo.com/es_ES/.
13. **MyOdoo.** *MyOdoo - your bussiness at your fingertips. The power of Odoo in one mobile App.* [En línea] 2018. [Citado el: 5 de 2 de 2018.] <https://www.myodoo.com/>.
14. **Technologies, JPC.** *modoo.* [En línea] 2018. [Citado el: 8 de 2 de 2018.] <http://www.jpctechnologies.net/solutions/modoo>.
15. **Odoo.** *Odoo Mobile - Mobile apps for Android and IOS | Odoo.* [En línea] 2018. [Citado el: 6 de 2 de 2018.]
16. **Pimienta, Pedro.** *Tipos de aplicaciones móviles y sus características.* [En línea] deldeaApp, 2014. [Citado el: 28 de 05 de 2018.] <https://deideaaapp.org/tipos-de-aplicaciones-moviles-y-sus-caracteristicas/>.
17. **Fernandez Gutierrez, Alfonso y Chessa, Salvatore.** *Blogthinkbig.com. Aplicaciones web vs. aplicaciones nativas vs. aplicaciones híbridas.* [En línea] 2013. [Citado el: 5 de 10 de 2017.] <https://blogthinkbig.com/aplicaciones-web-nativas-hibridas>.

18. Tomás Gironés, Jesús. *El gran libro de Android*. 2013.
19. Porras, Eve. Ingeniería de sistemas. *Sistemas operativos móviles: iOS* . [En línea] 2012. [Citado el: 25 de 11 de 2017.] <http://eve-ingsistemas-u.blogspot.com/2012/04/sistemas-operativos-moviles-ios.html>.
20. CONCEPTODEFINICION.DE. Definición Windows Phone. [En línea] 2017. [Citado el: 25 de 11 de 2017.] <http://conceptodefinicion.de/windows-phone/>.
21. Kirschbaum, Ricardo. Clarín TECNOLOGÍA. [En línea] 2017. [Citado el: 26 de 11 de 2017.] https://www.clarin.com/tecnologia/android-vencio-windows-convertirse-sistema-operativo-usado-internet_0_Hk3xQ-xTg.html.
22. EcuRed. [En línea] [Citado el: 4 de Febrero de 2014.] http://www.ecured.cu/index.php/Proceso_Unificado_de_Desarrollo..
23. Sánchez, Tamara Rodríguez . *Metodología de Desarrollo para la Actividad Productiva*. La Habana : s.n.
24. Somerville, Ian. *Software Engineering*. 2011.
25. Rumbaugh, James, Jacobson, Ivar y Booch, Grady. *El Lenguaje Unificado de Modelado. Manual de Referencia*.
26. Informática.inei., Sub-Jefatura de. [En línea] Noviembre de 1999. [Citado el: 10 de Febrero de 2014.] <http://www.inei.gob.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf..>
27. Valdez, Altamirano Alfonso. [En línea] [Citado el: 18 de Febrero de 2014.] <http://docs.google.com/viewer?a=v&q=cache:vo4SGq-b8gkJ:ubicuos.com/wp-content/uploads/2009/05/comparativoides.pdf+visual+paradigm%2Bcaracteristicas&hl=es-419&gl=cu&pid=bl&srcid=A>.
28. Alegs. ALEGSA.com.ar. *Definición de Framework de desarrollo (Informática)*. [En línea] 2014. [Citado el: 25 de 11 de 2017.] <http://www.alegsa.com.ar/Dic/framework.php>.
29. Garcia Peñalvo, Francisco José y García Holgado, Alicia. *Dominio de modelado*. 2017.
30. Patricio, Letelier. *Metodologías ágiles para el desarrollo de software: eXtreme Programming*. 2006.
31. University, Atlantic Internarional. AIU. *AIU- Atlantic Internarional University*. [En línea] 2018. [Citado el: 22 de 5 de 2018.] <https://www.aiu.edu/cursos/base%20de%20datos/pdf%20leccion%202/lecci%C3%B3n%202.pdf>.


32. Fabien Potencier, François Zaninotto. *Symfony, la guía definitiva*. Apress : s.n., 2007.
33. Patrones de diseño. *Ingeniería de Software II*. Madrid : s.n., 2011.
34. Aria, Miriela Velazquez. *Componente de medición de la calidad de imágenes de huellas dactilares*. La Habana : s.n., 2014.
35. I. Jacobson, G. Booch, J. Rumbaugh. *El proceso unificado de desarrollo de software*. Madrid : s.n., 2000.
36. Reoger, Pressman. *Ingeniería del Software:Un Enfoque Práctico*. 2010.
37. *Técnicas de pruebas*. 2015.
38. EcuRed. [En línea] [Citado el: 2018 de 05 de 06.] http://www.ecured.cu/index.php/Pruebas_de_caja_negra.
39. Pressman, Roger. *Ingeniería del Software. Un enfoque Práctico*. 2005. ISBN: 9701054733.
40. EcuRed. [En línea] 6 de 05 de 2018. http://www.ecured.cu/index.php/Pruebas_de_caja_blanca.
41. Rodríguez, Jorge. *Pruebas unitarias*. 2006.
42. Grzejszczak, Marcin. *Mockito Cookbook*. s.l. : Packt Publishing Ltd, 2014.

Glosario de términos


- **Framework:** En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software.
- **TIC:** Tecnologías de la Información y las Comunicaciones. Conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones.
- **UML:** Unified Modeling Language (Lenguaje Unificado de Modelado). Lenguaje de modelado de sistemas de software más conocido en la actualidad.
- **XML:** eXtensible Markup Language (Lenguaje de Marca Extensible). Metalenguaje extensible de etiquetas desarrollado, de marcado sencillo, similar al HTML.
- **Integración:** Conjunto de elementos relacionados o que interactúan que permiten implantar y alcanzar la política y los objetivos de una organización, en lo que se refiere a aspectos diversos como pueden ser los de seguridad. Es el proceso de combinar y probar múltiples componentes juntos. El objetivo es tomar los componentes probados en unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño. Se llama integración incremental cuando el programa se construye y se prueba en pequeños segmentos en los que los errores son más fáciles de aislar y corregir, es más probable que se pueda probar completamente las interfaces y se puede aplicar un enfoque de prueba sistemática.
- **IEEE/LOM (Institute of Electrical and Electronics Engineers/Learning Technology Standards Committee):** Asociación internacional, cuya misión es promover los procesos ingenieriles para la creación, desarrollo, integración, compartición y aplicaciones del conocimiento sobre tecnologías eléctricas y de información

Anexo 1: Historias de usuario

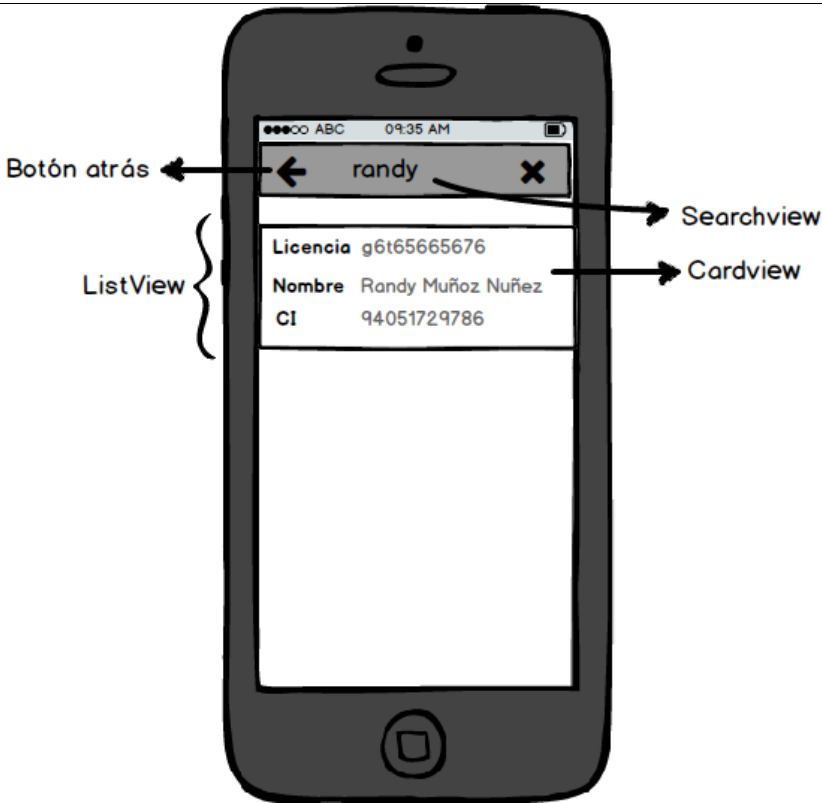
HU_ 2: Listar conductores.

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
1	Nuevo	4	4	2	1	5
Nombre						
Listar Conductores						
Notas						
Descripción						
El objetivo de esta actividad es listar los conductores existentes en el sistema. Para ello el usuario seleccionará la opción "Conductor" del menú y se desplegará el listado con los datos más descriptivos de los conductores.						
Prototipo de Interfaz						
 <p>The image shows a mobile application prototype for 'Conductores'. The screen displays a list of three driver cards. Each card contains the following information: 'Licencia' (License), 'Nombre' (Name), and 'CI' (C.I.D.). The first card shows license 'g6t65665676', name 'Randy Muñoz Nuñez', and CI '94051729786'. The second card shows license 'k897877787', name 'Pepe Nillar', and CI '1111111111'. The third card shows license 'j3838393393', name 'Sandy Perez', and CI '94051793883'. Annotations with arrows point to: 'Ícono del menú lateral' (hamburger menu icon), 'Cardview' (one of the driver cards), 'Botón flotante' (a floating search button), and 'ListView' (the entire list of cards).</p>						

HU_ 3: Ver detalles de un conductor.

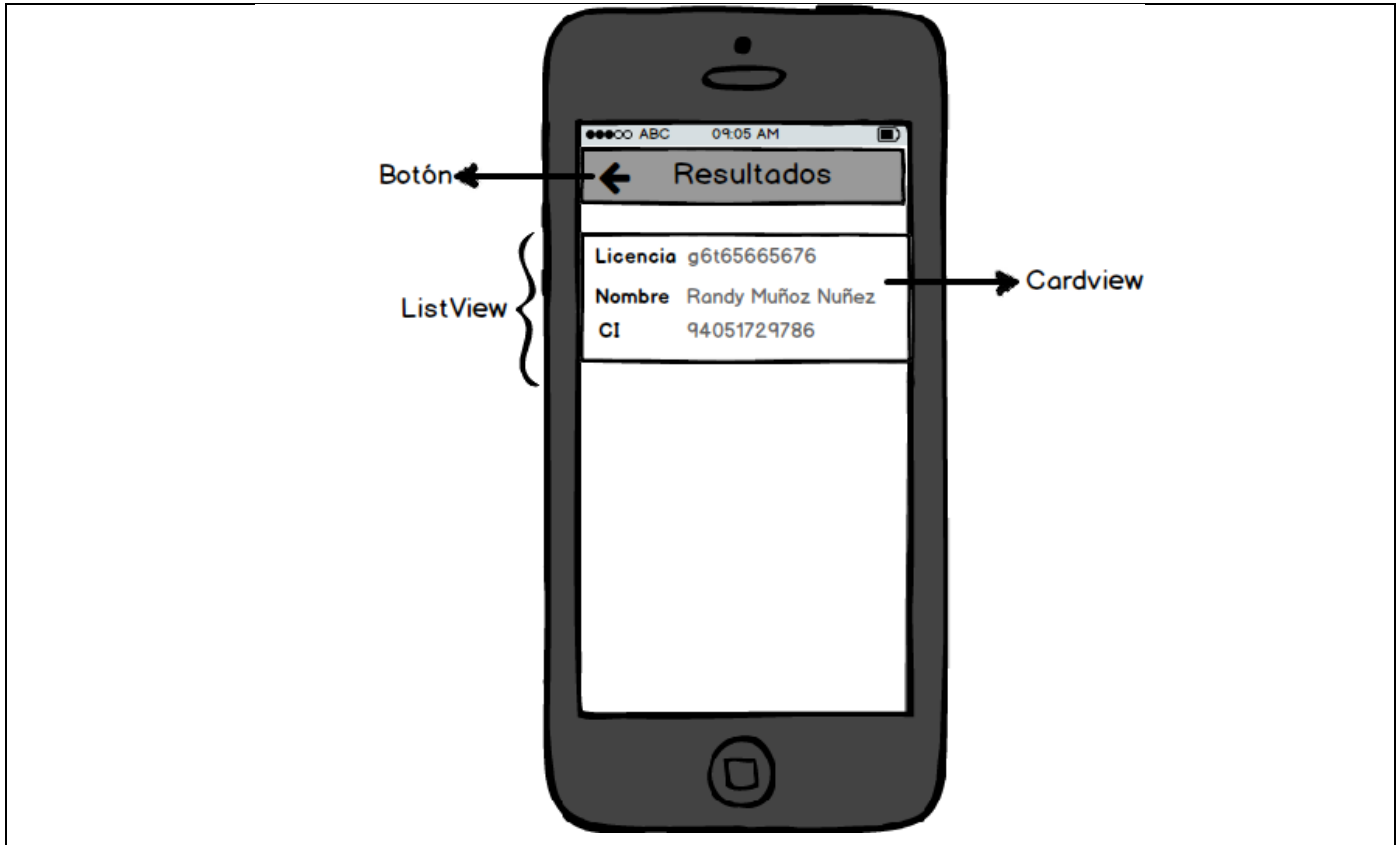
Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
2	Nuevo	4	4	1	0.5	5
Nombre						
Ver detalles de un conductor						
Notas						
Descripción						
El objetivo de esta actividad permitir ver los detalles de un conductor existente en el sistema. Inicialmente se muestra una vista con los conductores que han sido incluidos en el sistema. Al dar clic sobre uno de esos conductores se mostrará una vista con los detalles del conductor seleccionado.						
Prototipo de Interfaz						
 <p>The image shows a mobile application prototype on a smartphone. The screen displays a 'Datos del chofer' (Driver Data) screen. At the top, there is a status bar with 'ABC' and '06:27 AM'. Below the status bar is a header with a back arrow and the text 'Datos del chofer'. The main content area lists the following information: LICENCIA k897877787, NOMBRE Y APELLIDOS Randy Muñoz Nuñez, CI 94051793883, CARTA OPERATIVA PRUEBA, and NÚMERO DE ACCIDENTES 0. An arrow points from the text 'Botón atrás' to the back arrow in the header.</p>						

HU_ 4: Realizar búsqueda rápida de conductores.

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
3	Nuevo	3	3	0.5	0.5	2
Nombre						
Realizar búsqueda rápida de conductores						
Notas						
Descripción						
El objetivo de esta actividad permitir al usuario realizar una búsqueda rápida sobre el listado de conductores. Para ello se introducirá un texto y se mostrará un listado con los conductores que coincidan con el criterio.						
Prototipo de Interfaz						
 <p>The diagram shows a mobile application interface on a smartphone. At the top, there is a search bar containing the text 'randy'. To the left of the search bar is a back arrow labeled 'Botón atrás'. To the right is a close button labeled 'X'. Below the search bar, a list of search results is displayed, labeled as 'ListView'. The first result is a 'Cardview' containing the following information: 'Licencia g6t65665676', 'Nombre Randy Muñoz Nuñez', and 'CI 94051729786'. Arrows from the labels 'Searchview' and 'Cardview' point to the search bar and the first result card, respectively.</p>						

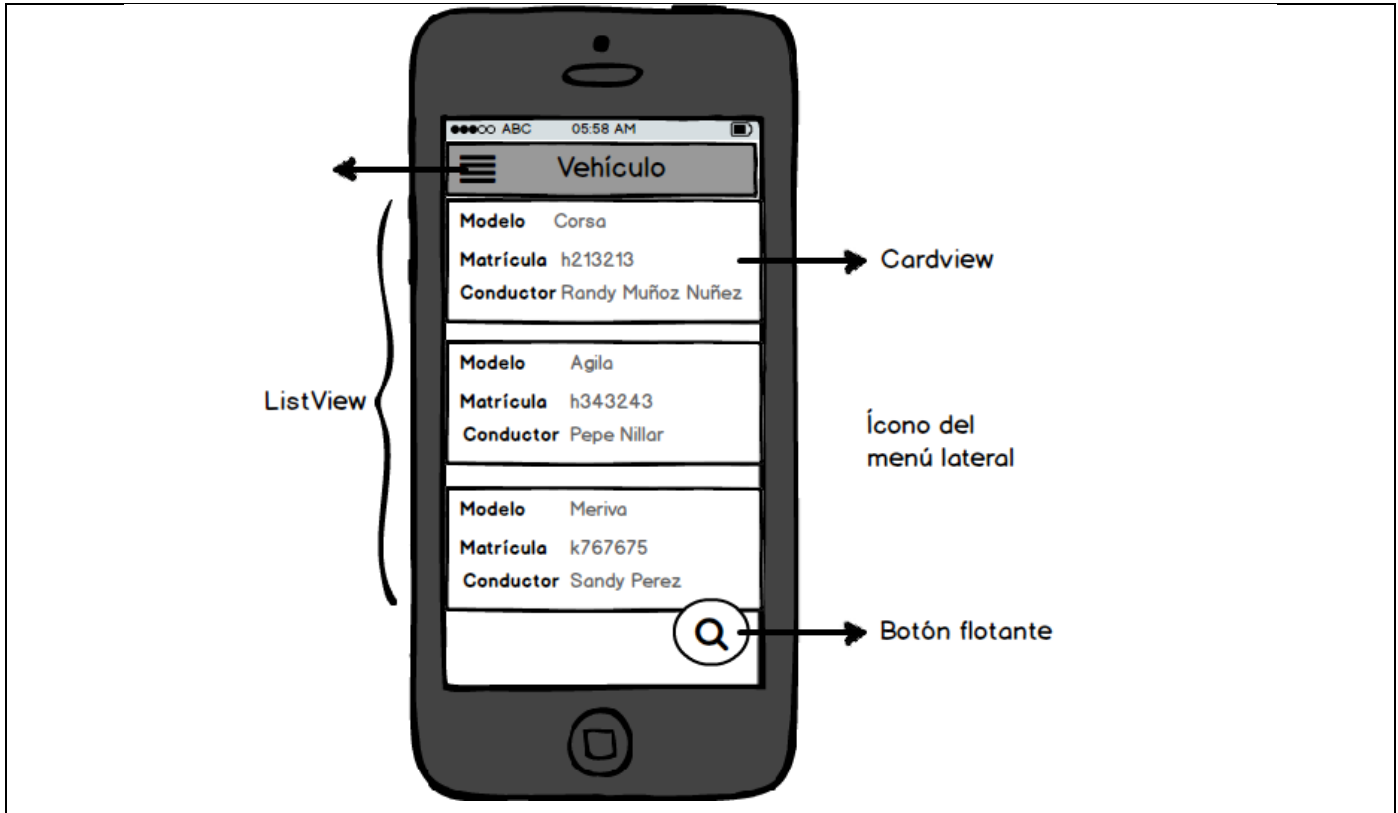
HU_ 5: Filtrar conductores

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
4	Nuevo	4	4	2	2	3
Nombre						
Filtrar conductores						
Notas						
Descripción						
<p>El objetivo de esta actividad es filtrar el listado de conductores existentes en el sistema. Para ello el usuario adicionará uno o varios criterios de búsqueda y presionar el botón "Buscar", se mostrará un listado con los resultados obtenidos.</p>						
Prototipo de Interfaz: Adicionar criterio						
<p> Botón atrás ← → Botón Lista desplegable ← → Botón Lista desplegable ← → Campo de texto </p>						
Prototipo de Interfaz: Mostrar resultado						



HU_ 6: Listar vehículos

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
5	Nuevo	4	4	2	1	5
Nombre						
Listar Vehículos						
Notas						
Descripción						
El objetivo de esta actividad es listar los vehículos existentes en el sistema. Para ello el usuario seleccionará la opción "Vehículo" del menú y se aparecerá un listado con los datos más descriptivos de los vehículos.						
Prototipo de Interfaz						



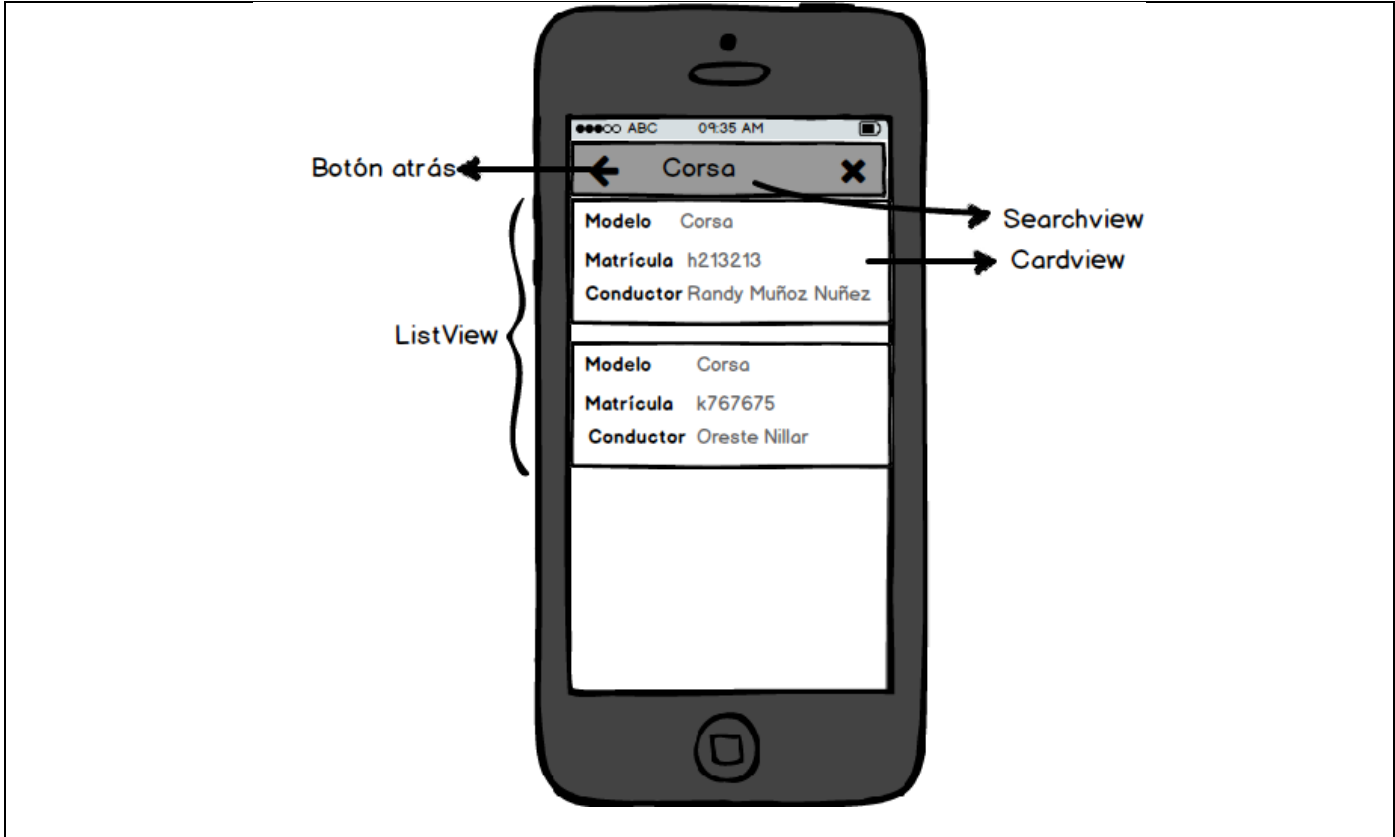
HU_ 7: Ver detalles de un vehículo

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
6	Nuevo	4	4	2	1	5
Nombre						
Ver detalles de un vehículo						
Notas						
Descripción						
El objetivo de esta actividad permitir ver los detalles de un vehículo existente en el sistema. Inicialmente se muestra una vista con los vehículos que han sido incluidos en el sistema. Al dar clic sobre uno de esos vehículos se mostrará una vista con los detalles del vehículo seleccionado.						
Prototipo de Interfaz						

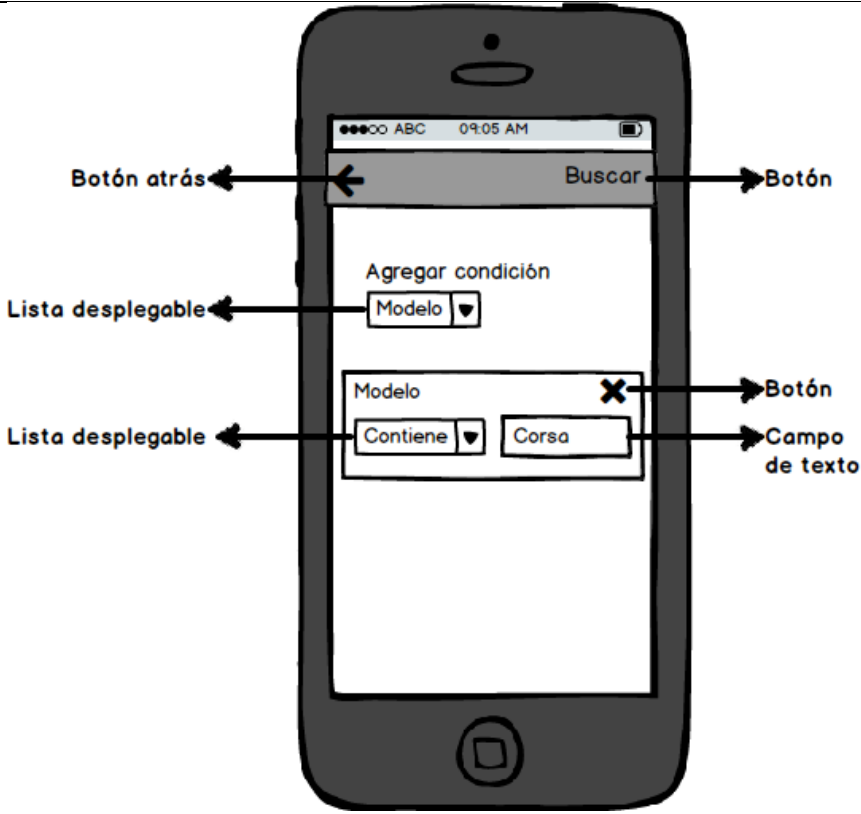


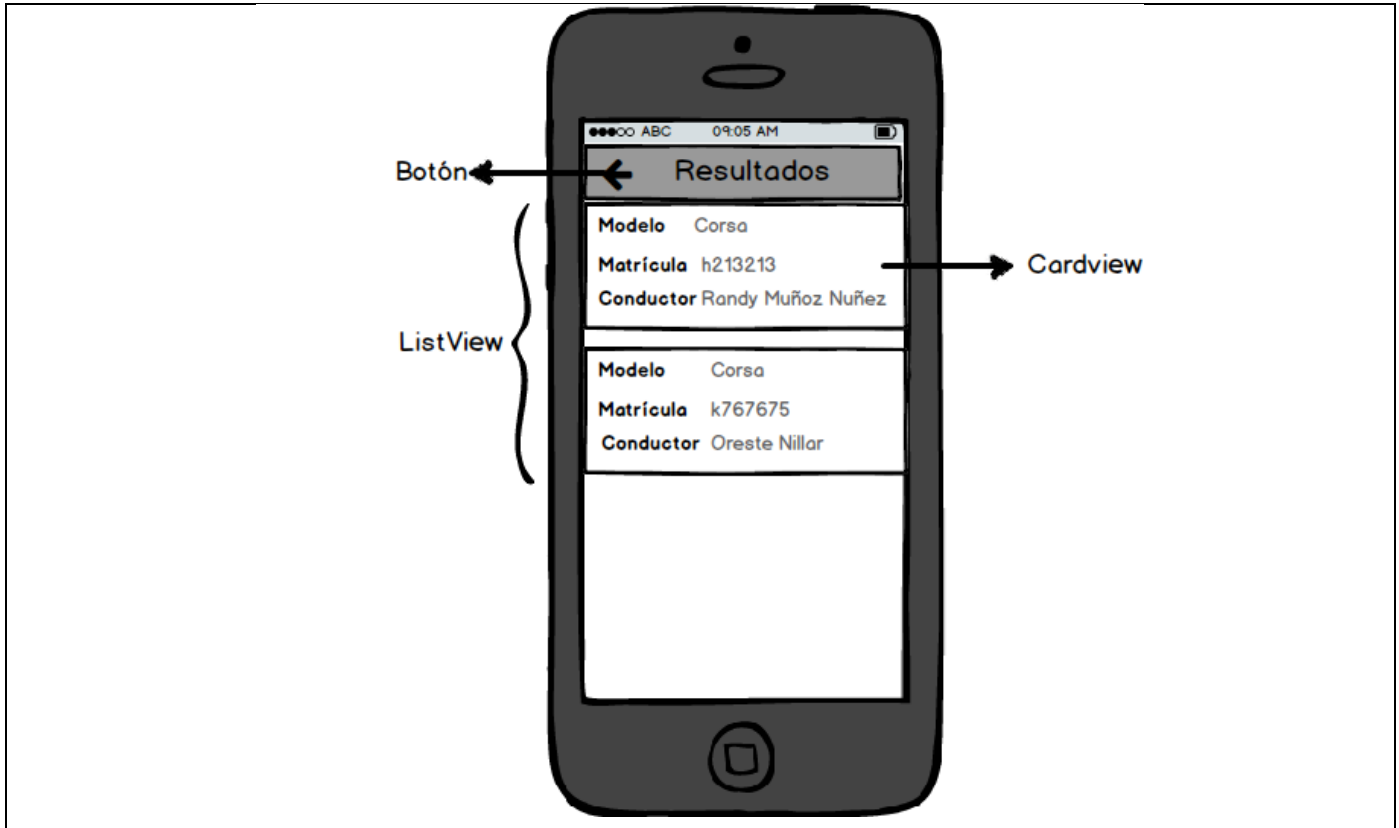
HU_ 8: Realizar búsqueda rápida de vehículos

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
7	Nuevo	3	3	0.5	0.5	2
Nombre						
Realizar búsqueda rápida de vehículos						
Notas						
Descripción						
El objetivo de esta actividad permitir al usuario realizar una búsqueda rápida sobre el listado de vehículos. Para ello se introducirá un texto y se mostrará un listado con los vehículos que coincidan con el criterio.						
Prototipo de Interfaz						



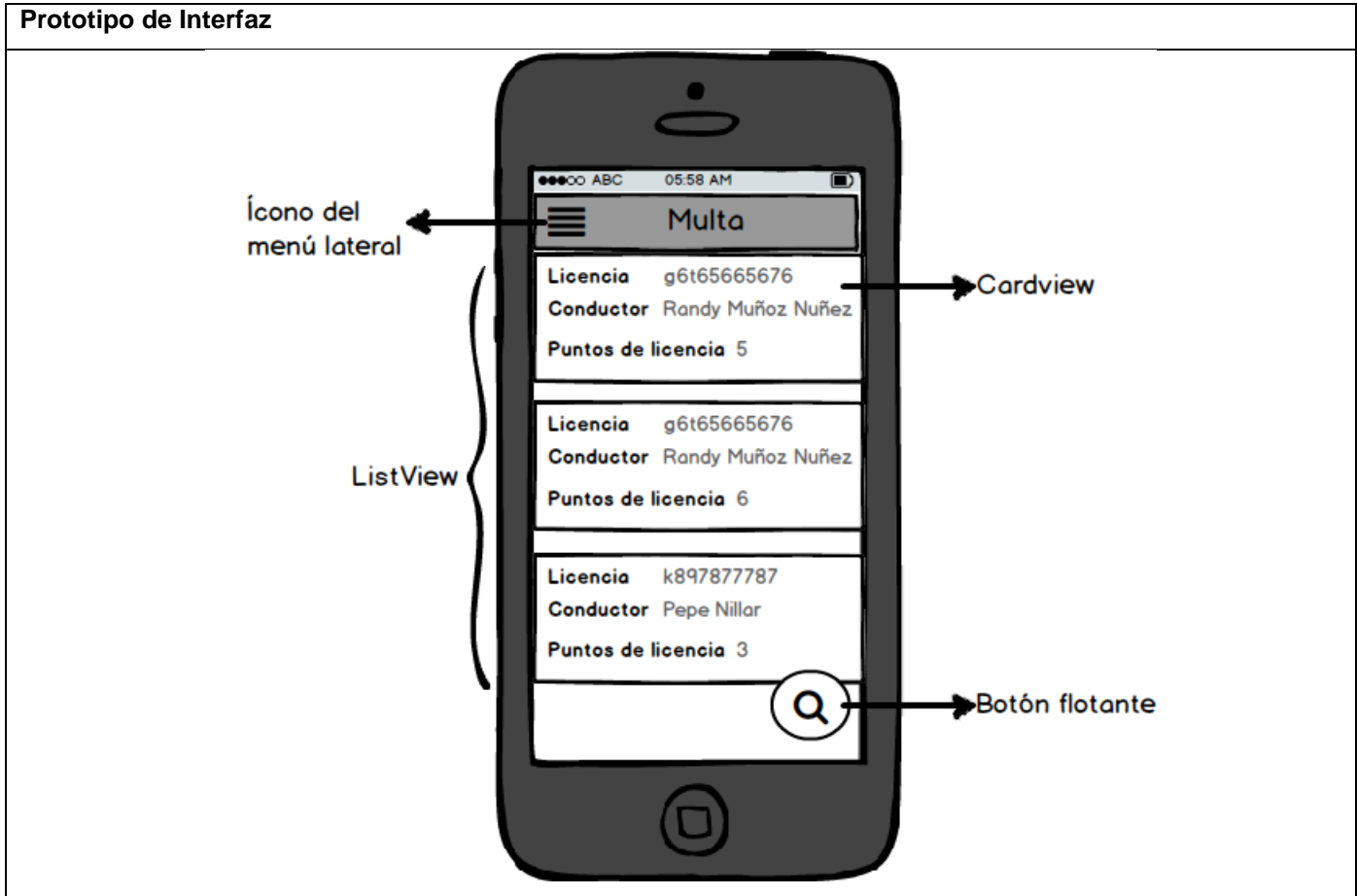
HU_ 9: Filtrar vehículos

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
8	Nuevo	4	4	2	2	3
Nombre						
Filtrar vehículos						
Notas						
Descripción						
El objetivo de esta actividad es filtrar el listado de los vehículos existentes en el sistema. Para ello el usuario adicionará uno o varios criterios de búsqueda y presionar el botón “Buscar”, se mostrará un listado con los resultados obtenidos.						
Prototipo de Interfaz: Adicionar criterio						
						
Prototipo de Interfaz: Mostrar resultados						



HU_ 10: Listar multas

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
9	Nuevo	4	4	2	1	5
Nombre						
Listar multas						
Notas						
Descripción						
El objetivo de esta actividad es listar las multas existentes en el sistema. Para ello el usuario seleccionará la opción "Multas" del menú y se aparecerá un listado con los datos más descriptivos de las multas.						



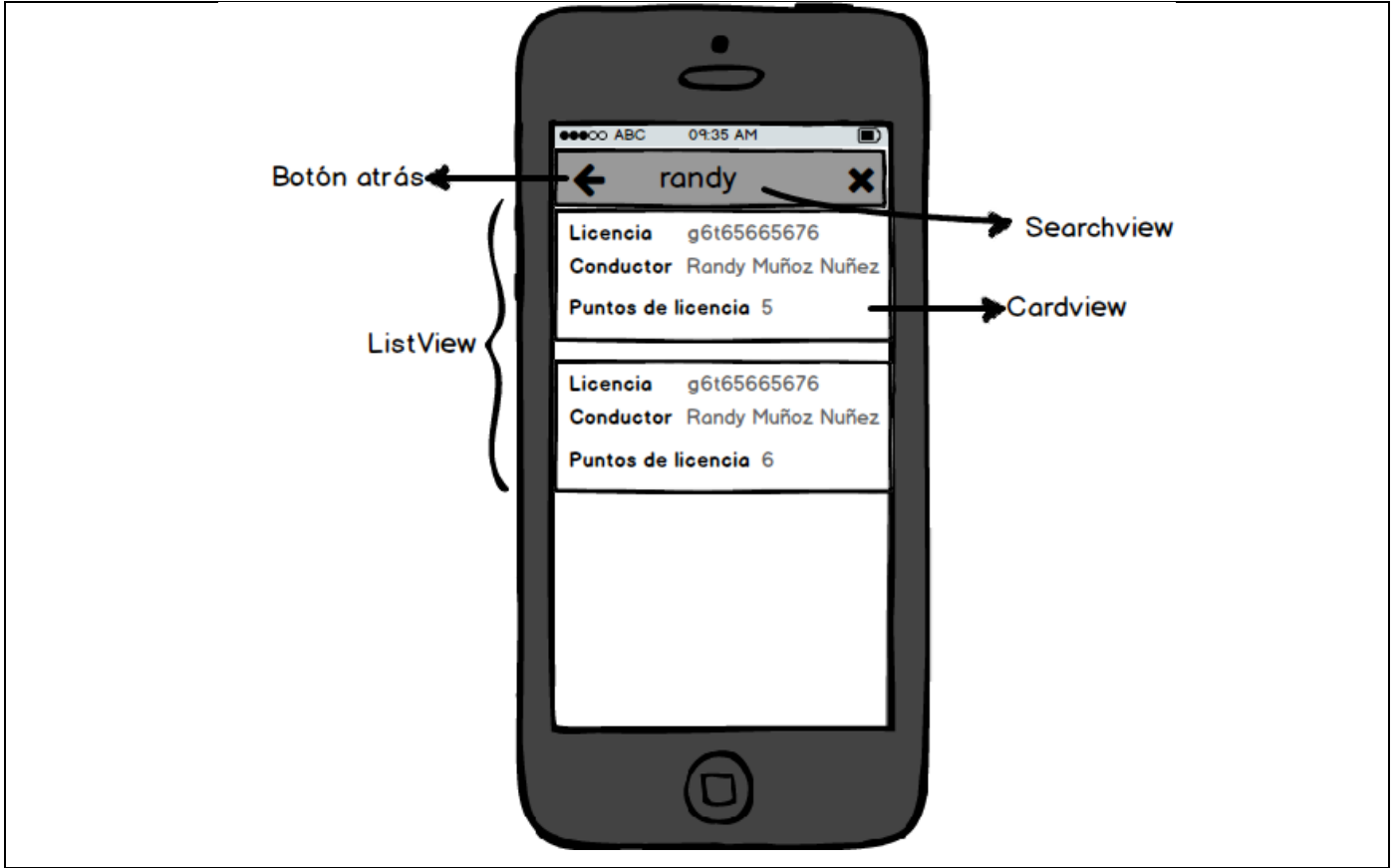
HU_ 11: Ver detalles de una multa

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
10	Nuevo	4	4	2	1	5
Nombre						
Ver detalles de una multa						
Notas						
Descripción						
El objetivo de esta actividad permitir ver los detalles de una multa existente en el sistema. Inicialmente se muestra una vista con las multas que han sido incluidos en el sistema. Al dar clic sobre uno de esas multas se mostrará una vista con los detalles de la multa seleccionada.						
Prototipo de Interfaz						



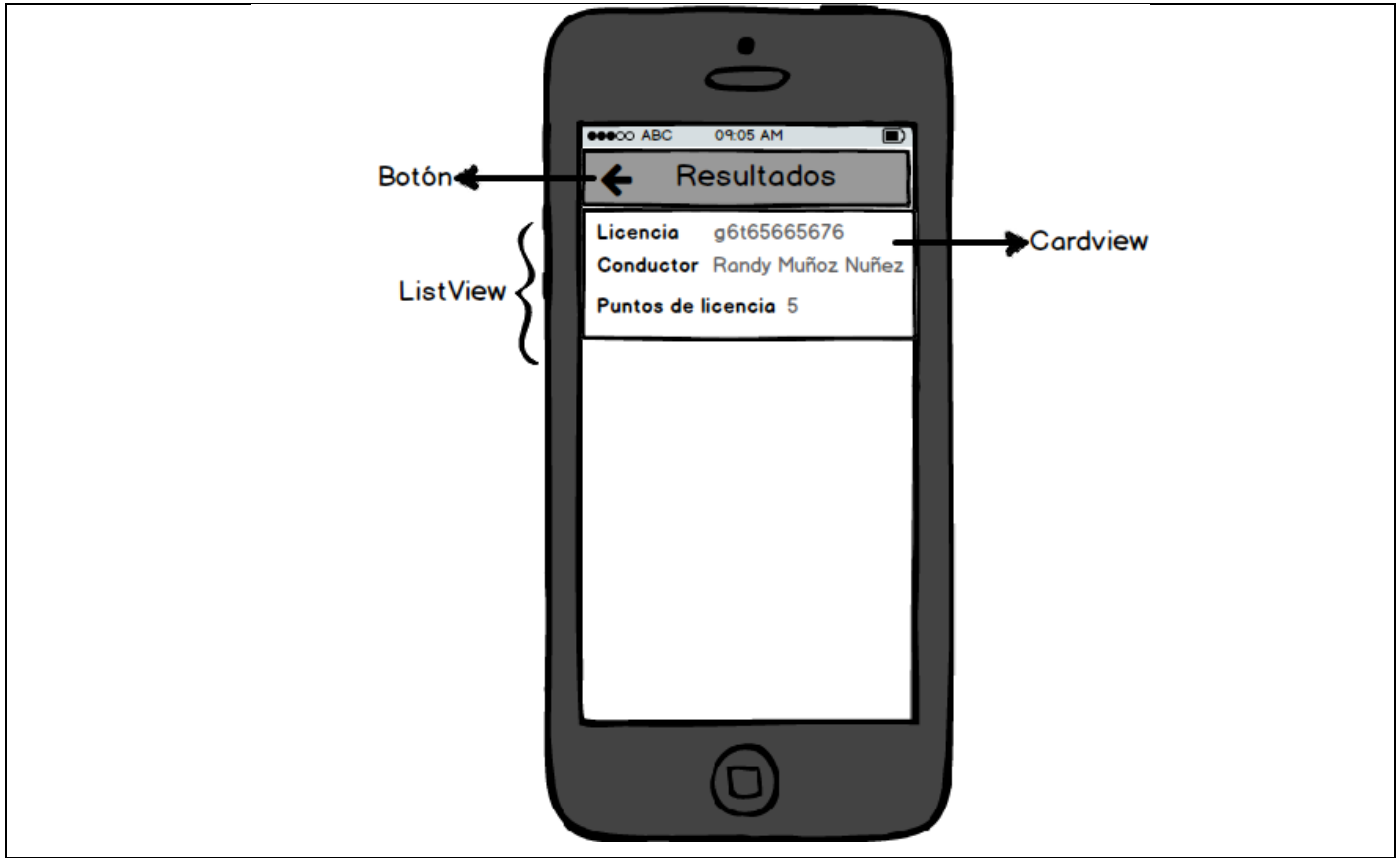
HU_ 12: Realizar búsqueda rápida de multas

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
11	Nuevo	3	3	0.5	0.5	2
Nombre						
Realizar búsqueda rápida de multas						
Notas						
Descripción						
El objetivo de esta actividad permitir al usuario realizar una búsqueda rápida sobre el listado de multas. Para ello se introducirá un texto y se mostrará un listado con las multas que coincidan con el criterio.						
Prototipo de Interfaz						



HU_ 13: Filtrar multas

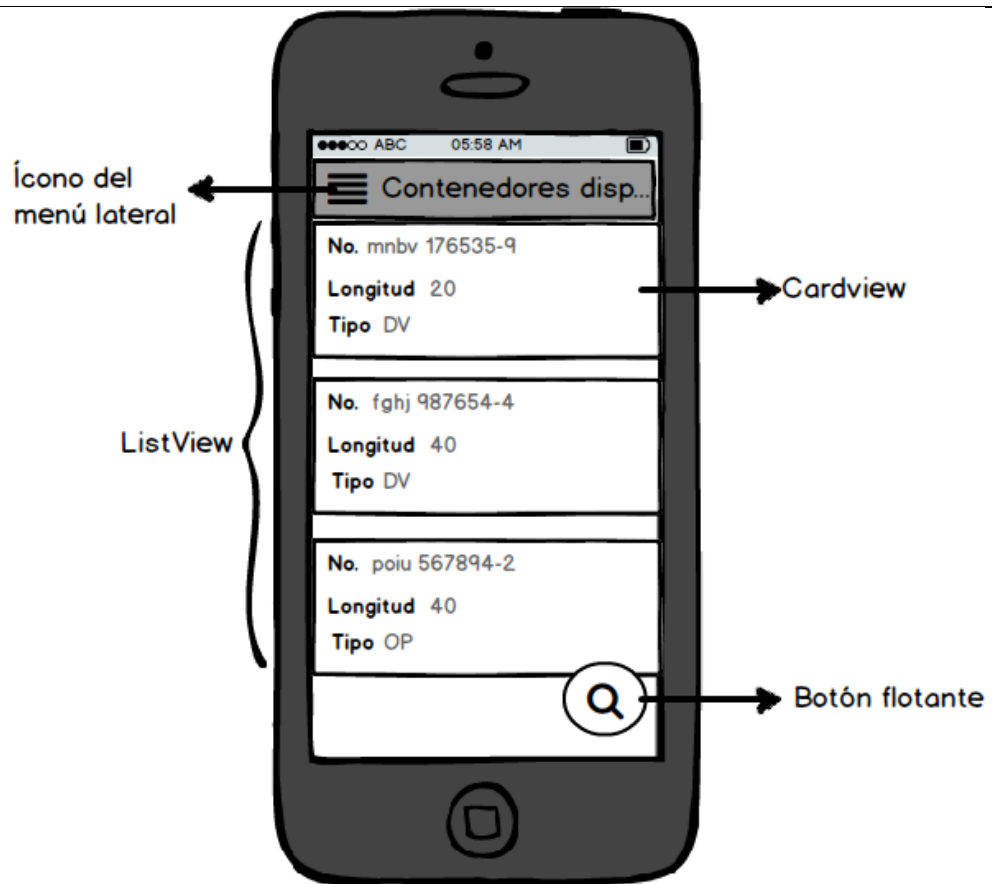
Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
12	Nuevo	4	4	2	2	3
Nombre						
Filtrar multas						
Notas						
Descripción						
El objetivo de esta actividad es filtrar el listado de las multas existentes en el sistema. Para ello el usuario adicionará uno o varios criterios de búsqueda y presionar el botón “Buscar”, se mostrará un listado con los resultados obtenidos.						
Prototipo de Interfaz: Adicionar criterio						
Prototipo de Interfaz: Mostrar resultados						




HU_ 14: Listar contenedores disponibles

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
13	Nuevo	4	4	2	1	5
Nombre						
Listar contenedores disponibles						
Notas						
Descripción						
El objetivo de esta actividad es listar los contenedores disponibles existentes en el sistema. Para ello el usuario seleccionará la opción "Contenedores disponibles" del menú y se aparecerá un listado con los datos más descriptivos de los contenedores disponibles.						

Prototipo de Interfaz



HU_ 14: Ver detalles de un contenedor disponible

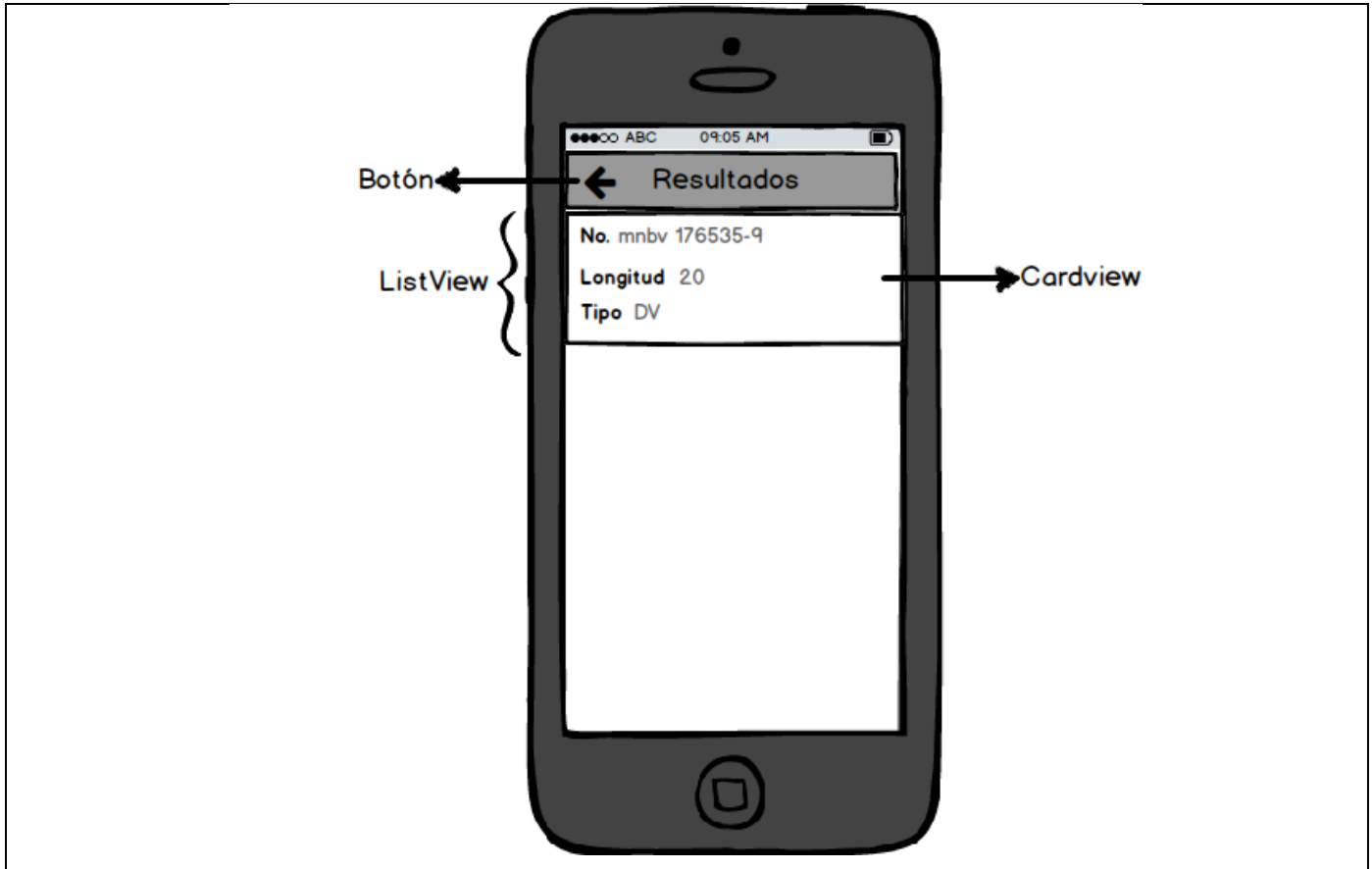
Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
14	Nuevo	4	4	2	1	5
Nombre						
Ver detalles de un contenedor disponible						
Notas						
Descripción						
El objetivo de esta actividad permitir ver los detalles de un contenedor disponible existente en el sistema. Inicialmente se muestra una vista con los contenedores disponibles que han sido incluidos en el sistema. Al dar clic sobre uno de esos contenedores disponibles se mostrará una vista con los detalles del contenedor disponible seleccionado.						
Prototipo de Interfaz						
						

HU_ 15: Realizar búsqueda rápida de contenedores disponibles

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
15	Nuevo	3	3	0.5	0.5	2
Nombre						
Realizar búsqueda rápida de contenedores disponibles						
Notas						
Descripción						
El objetivo de esta actividad permitir al usuario realizar una búsqueda rápida sobre el listado de contenedores disponibles. Para ello se introducirá un texto y se mostrará un listado con los contenedores disponibles que coincidan con el criterio.						
Prototipo de Interfaz						

HU_ 16: Filtrar contenedores disponibles

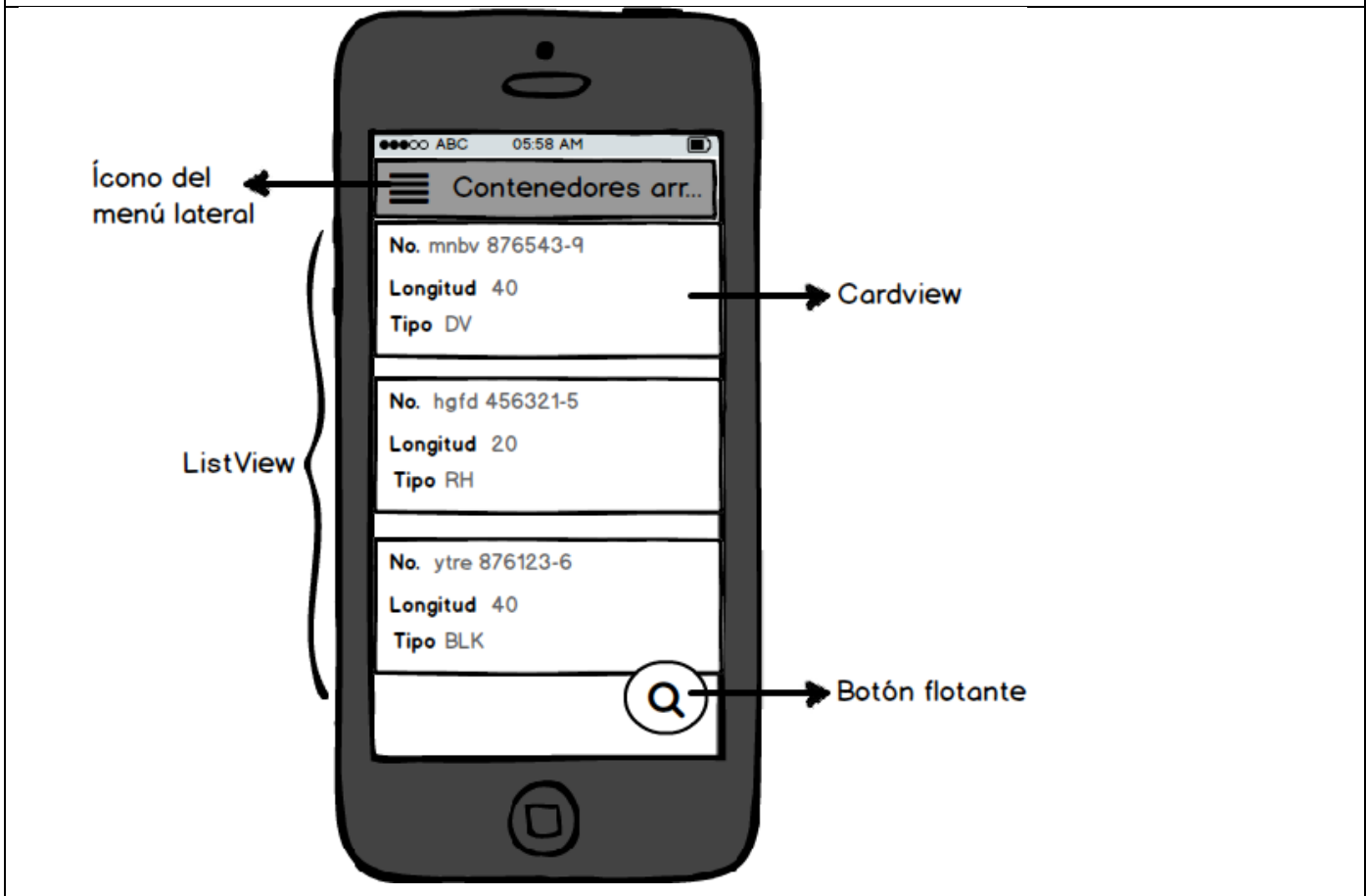
Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
16	Nuevo	4	4	2	2	3
Nombre						
Filtrar contenedores disponibles						
Notas						
Descripción						
El objetivo de esta actividad es filtrar el listado de los contenedores disponibles existentes en el sistema. Para ello el usuario adicionará uno o varios criterios de búsqueda y presionar el botón "Buscar", se mostrará un listado con los resultados obtenidos.						
Prototipo de Interfaz: Adicionar criterio						
<p>Botón atrás ←</p> <p>→ Botón</p> <p>Lista desplegable ←</p> <p>→ Botón</p> <p>Lista desplegable ←</p> <p>→ Campo de texto</p>						
Prototipo de Interfaz: Mostrar resultados						



HU_17: Listar contenedores arrendados

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
17	Nuevo	4	4	2	1	5
Nombre						
Listar contenedores arrendados						
Notas						
Descripción						
El objetivo de esta actividad es listar los contenedores arrendados existentes en el sistema. Para ello el usuario seleccionará la opción "Contenedores arrendados" del menú y se aparecerá un listado con los datos más descriptivos de los contenedores arrendados.						

Prototipo de Interfaz

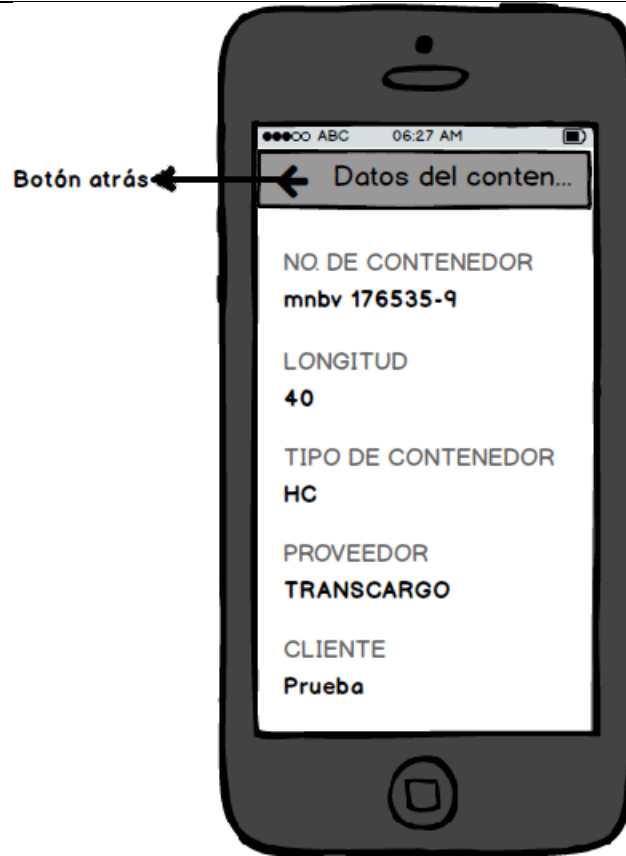


HU_ 18: Ver detalles de un contenedor arrendados

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
18	Nuevo	4	4	2	1	5
Nombre						
Ver detalles de un contenedor arrendado						
Notas						
Descripción						

El objetivo de esta actividad permitir ver los detalles de un contenedor arrendado existente en el sistema. Inicialmente se muestra una vista con los contenedores arrendados que han sido incluidos en el sistema. Al dar clic sobre uno de esos contenedores arrendados se mostrará una vista con los detalles del contenedor arrendado seleccionado.

Prototipo de Interfaz



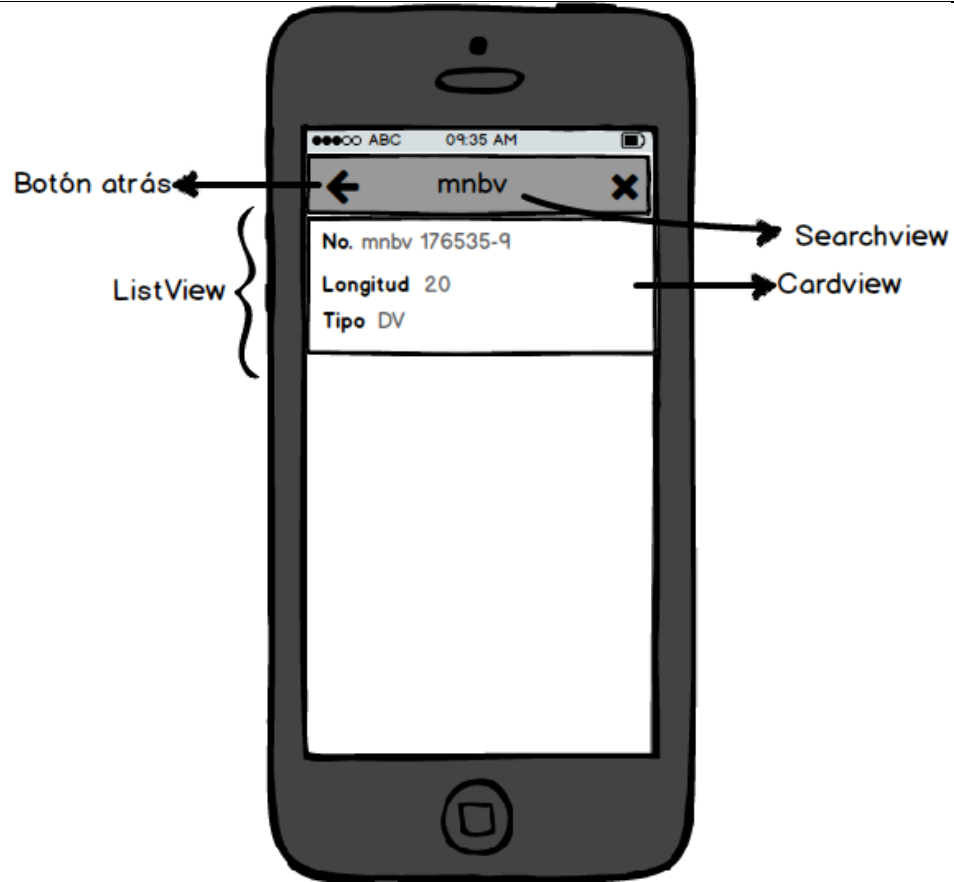
HU_ 19: Realizar búsqueda rápida de contenedores arrendados

Historia de usuario

Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
19	Nuevo	3	3	0.5	0.5	2
Nombre						
Realizar búsqueda rápida de contenedores arrendados						
Notas						
Descripción						

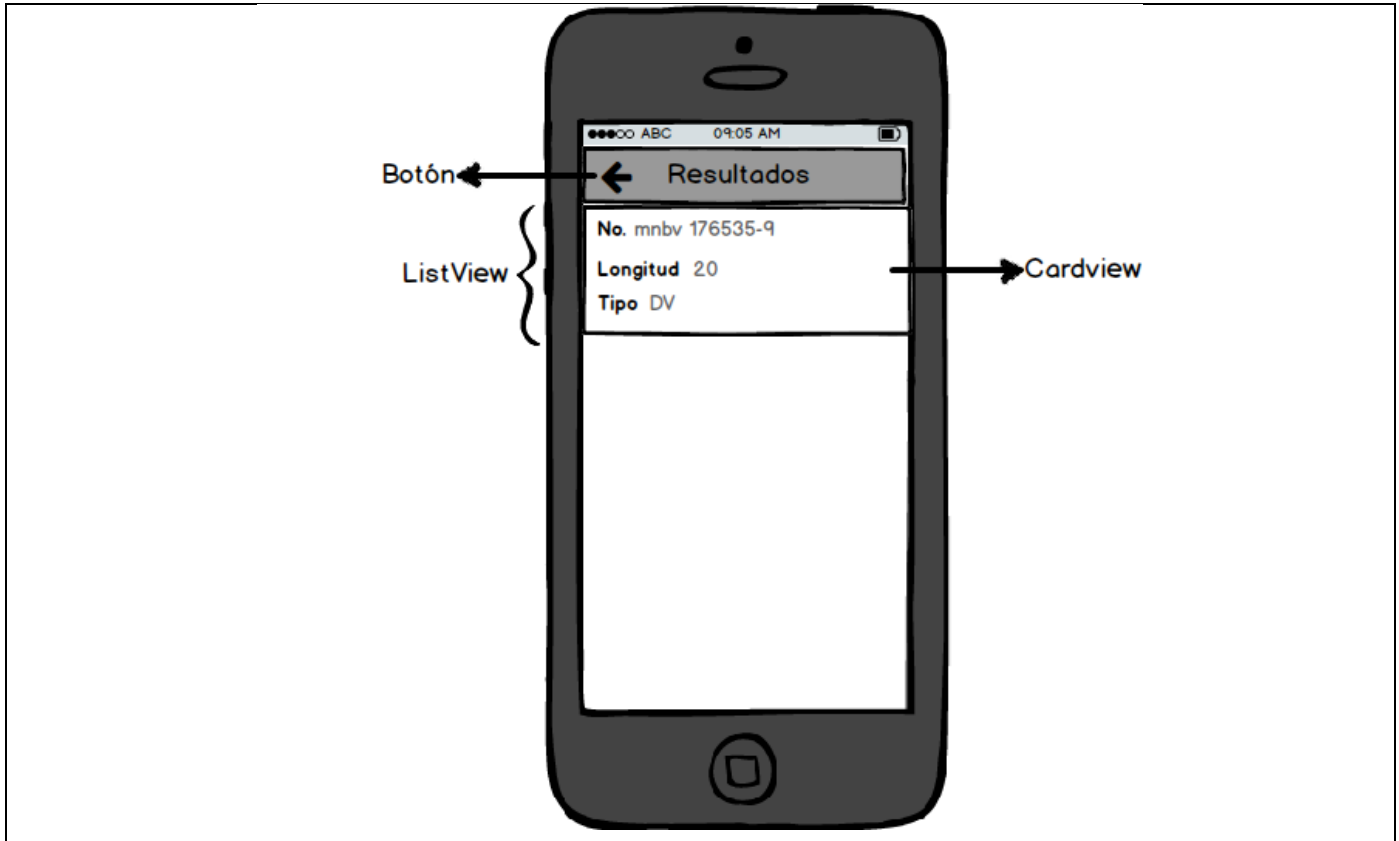
El objetivo de esta actividad permitir al usuario realizar una búsqueda rápida sobre el listado de contenedores arrendados. Para ello se introducirá un texto y se mostrará un listado con los contenedores arrendados que coincidan con el criterio.

Prototipo de Interfaz



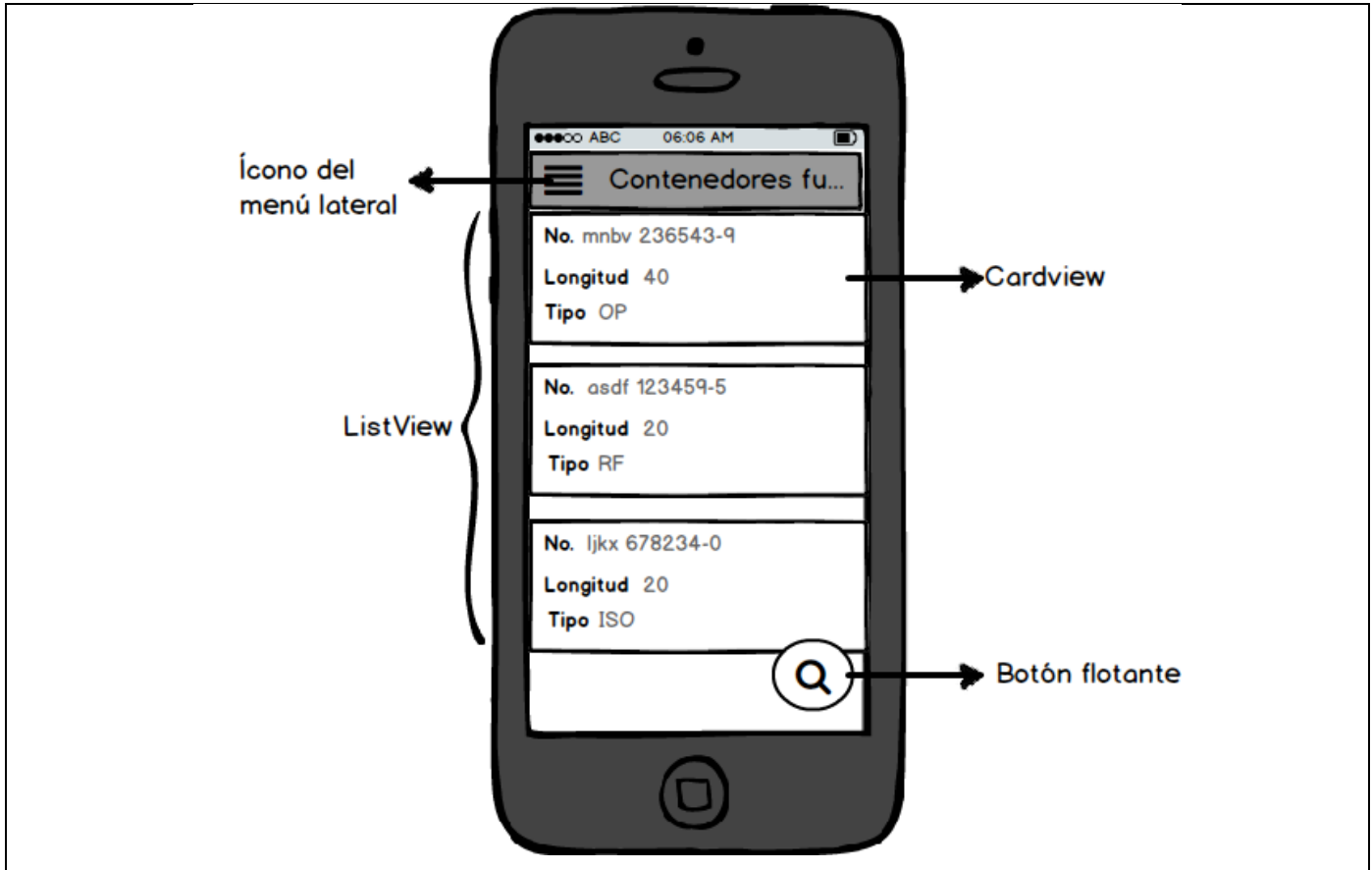
HU_ 20: Filtrar contenedores arrendados

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
20	Nuevo	4	4	2	2	3
Nombre						
Filtrar contenedores arrendados						
Notas						
Descripción						
El objetivo de esta actividad es filtrar el listado de los contenedores arrendados existentes en el sistema. Para ello el usuario adicionará uno o varios criterios de búsqueda y presionar el botón “Buscar”, se mostrará un listado con los resultados obtenidos.						
Prototipo de Interfaz: Adicionar criterio						
<p>Diagrama de un prototipo de interfaz de usuario en un teléfono inteligente. El formulario muestra un botón 'Buscar' en la parte superior derecha, un botón de retroceso en la parte superior izquierda, y un campo 'Agregar condición' con una lista desplegable 'No. de contenedor'. Abajo hay un campo de texto con un botón de eliminación 'X', un campo de texto 'mnbv 17', y una lista desplegable 'Contiene'.</p>						
Prototipo de Interfaz: Mostrar resultados						



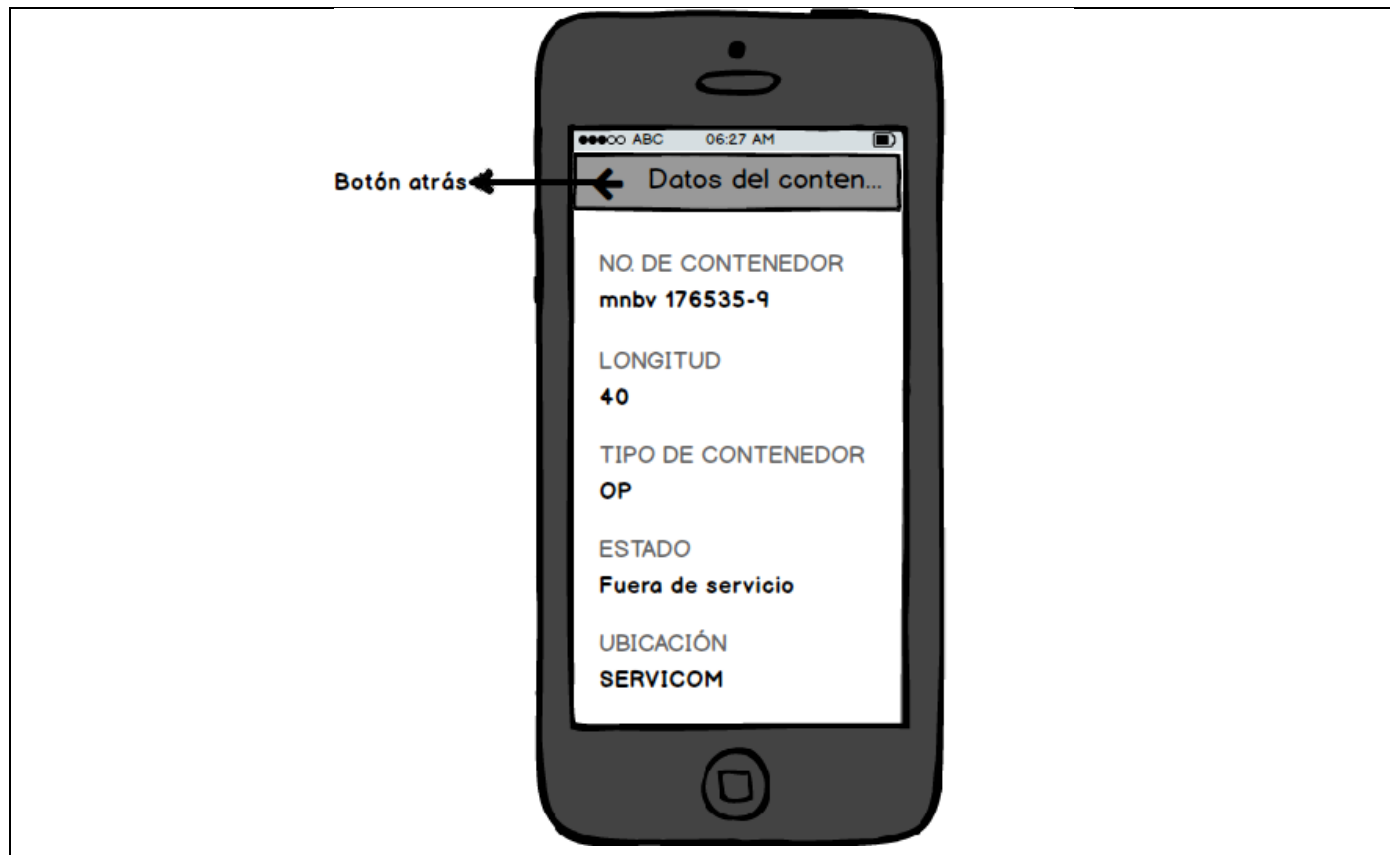
HU_21: Listar contenedores fuera de servicio

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
21	Nuevo	4	4	2	1	5
Nombre						
Listar contenedores fuera de servicio						
Notas						
Descripción						
El objetivo de esta actividad es listar los contenedores fuera de servicio existentes en el sistema. Para ello el usuario seleccionará la opción "Contenedores fuera de servicio" del menú y se aparecerá un listado con los datos más descriptivos de los contenedores fuera de servicio.						
Prototipo de Interfaz						



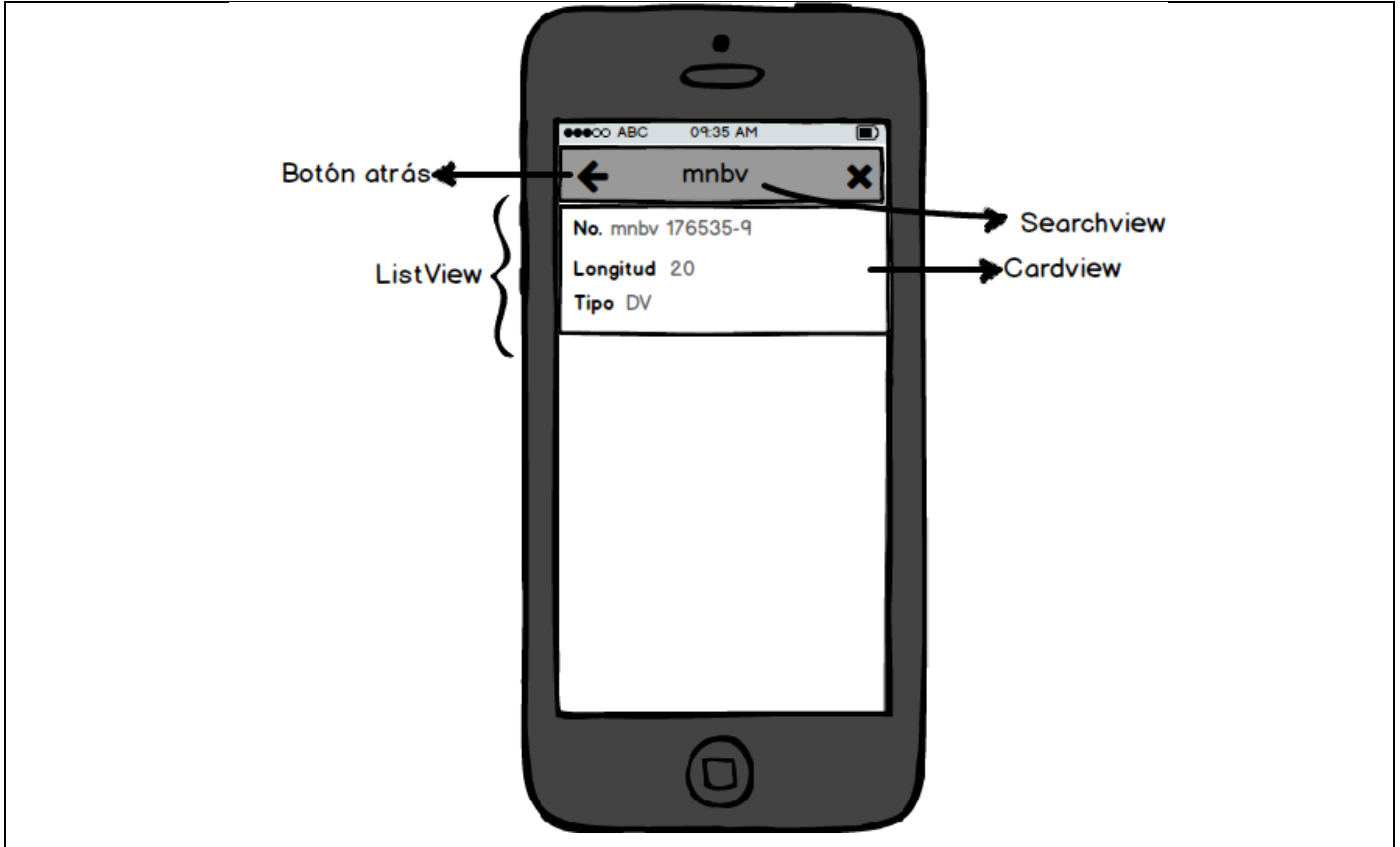
HU_ 22: Ver detalles de un contenedor fuera de servicio

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
22	Nuevo	4	4	2	1	5
Nombre						
Ver detalles de un contenedor fuera de servicio						
Notas						
Descripción						
El objetivo de esta actividad permitir ver los detalles de un contenedor fuera de servicio existente en el sistema. Inicialmente se muestra una vista con los contenedores fuera de servicio que han sido incluidos en el sistema. Al dar clic sobre uno de esos contenedores fuera de servicio se mostrará una vista con los detalles del contenedor fuera de servicio seleccionado.						
Prototipo de Interfaz						



HU_ 23: Realizar búsqueda rápida de contenedores fuera de servicio

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
23	Nuevo	3	3	0.5	0.5	2
Nombre						
Realizar búsqueda rápida de contenedores fuera de servicio						
Notas						
Descripción						
El objetivo de esta actividad permitir al usuario realizar una búsqueda rápida sobre el listado de contenedores fuera de servicio. Para ello se introducirá un texto y se mostrará un listado con los contenedores fuera de servicio que coincidan con el criterio.						
Prototipo de Interfaz						

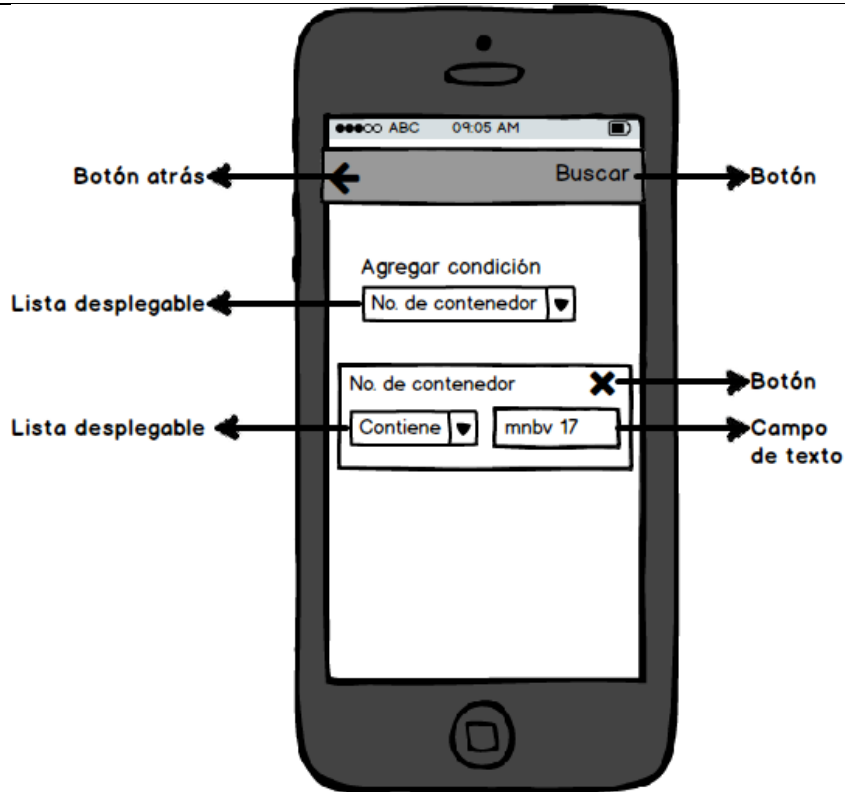


HU_ 24: Filtrar contenedores fuera de servicio

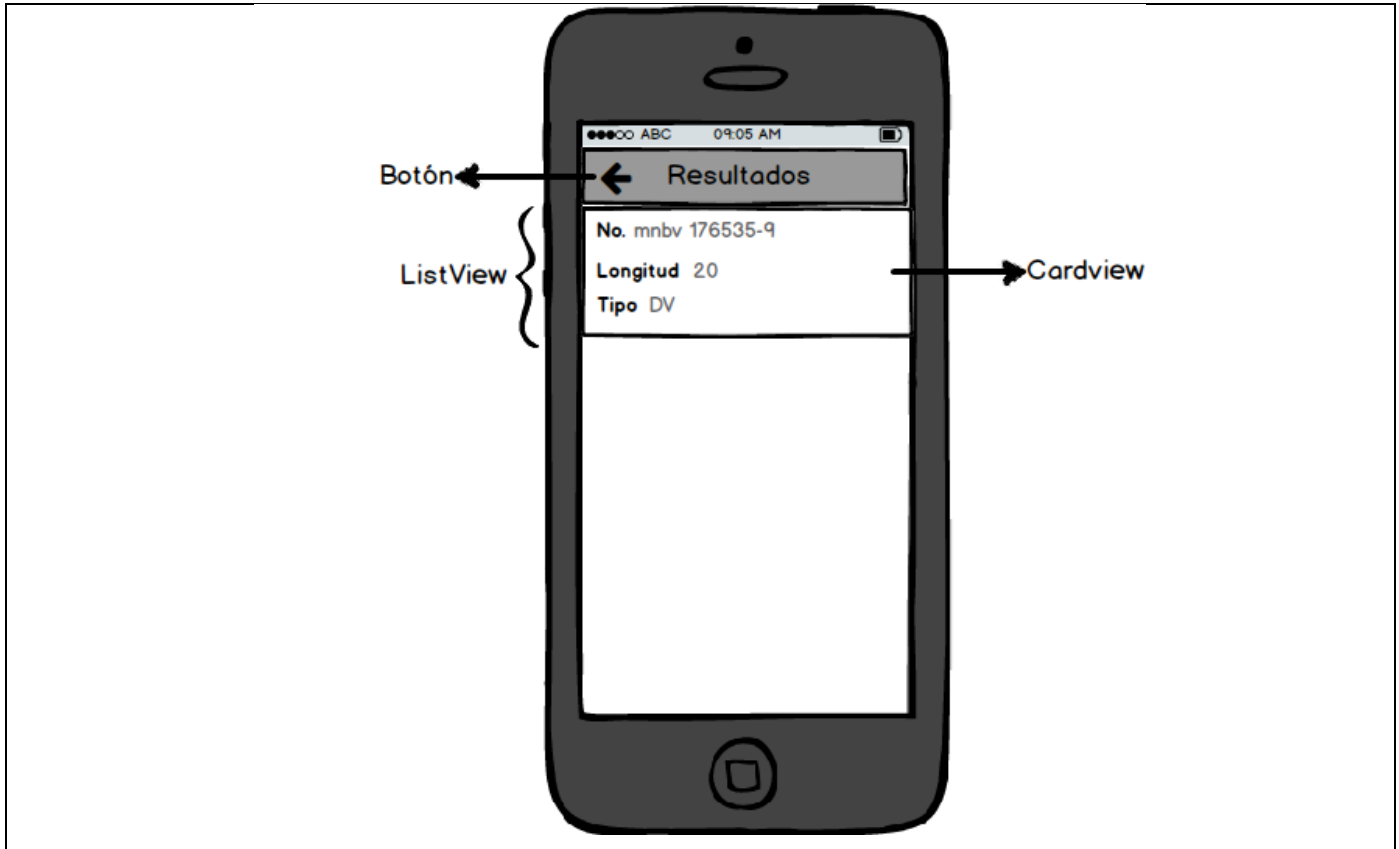
Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
24	Nuevo	4	4	2	2	3
Nombre						
Filtrar contenedores fuera servicio						
Notas						
Descripción						

El objetivo de esta actividad es filtrar el listado de los contenedores fuera servicios existentes en el sistema. Para ello el usuario adicionará uno o varios criterios de búsqueda y presionar el botón "Buscar", se mostrará un listado con los resultados obtenidos.

Prototipo de Interfaz: Adicionar criterio

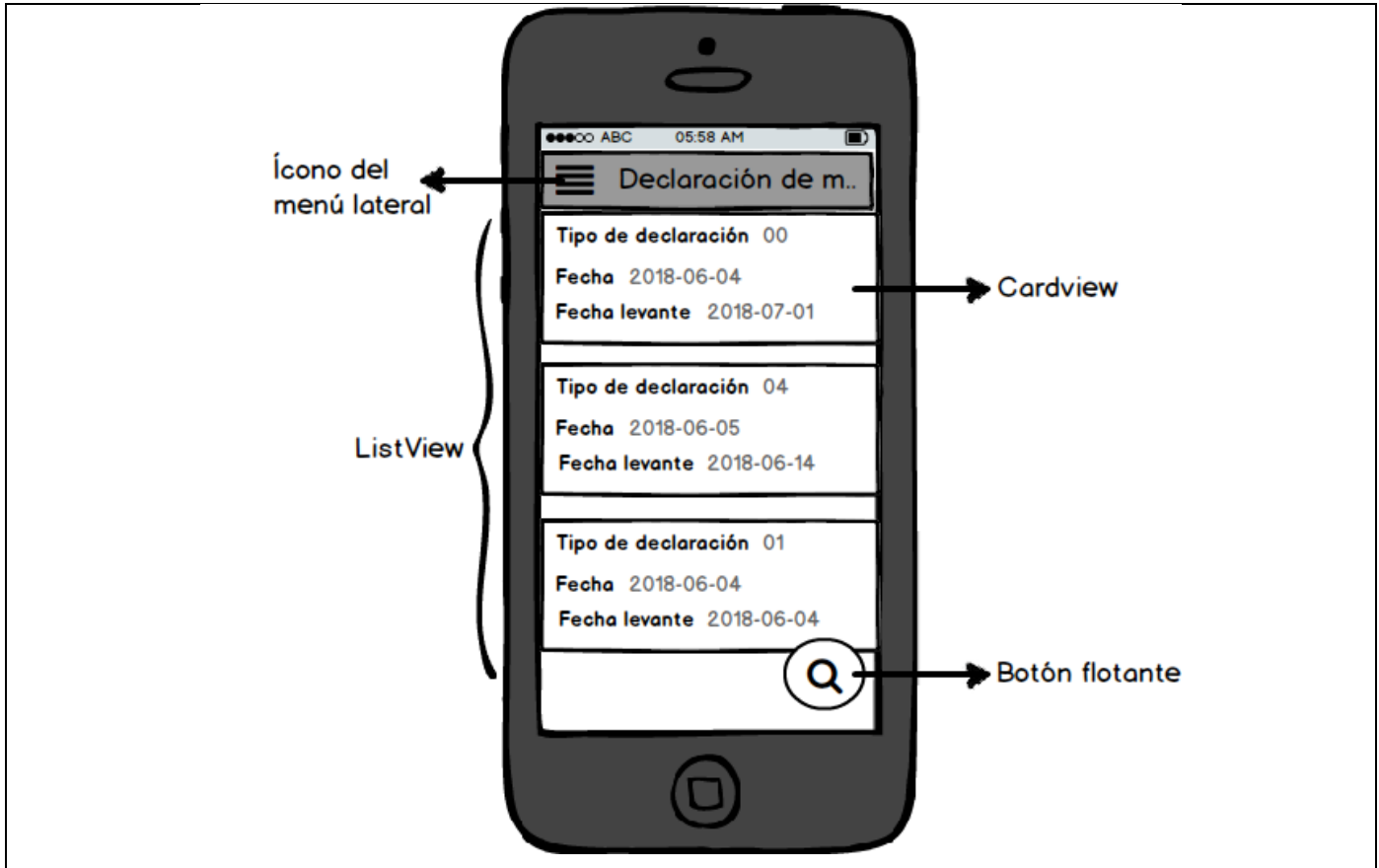


Prototipo de Interfaz: Mostrar resultados



HU_25: Listar declaraciones de mercancía

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
25	Nuevo	4	4	2	1	5
Nombre						
Listar declaraciones de mercancía						
Notas						
Descripción						
El objetivo de esta actividad es listar las declaraciones de mercancía existentes en el sistema. Para ello el usuario seleccionará la opción “Declaración de mercancía” del menú y se aparecerá un listado con los datos más descriptivos de las declaraciones de mercancía.						
Prototipo de Interfaz						



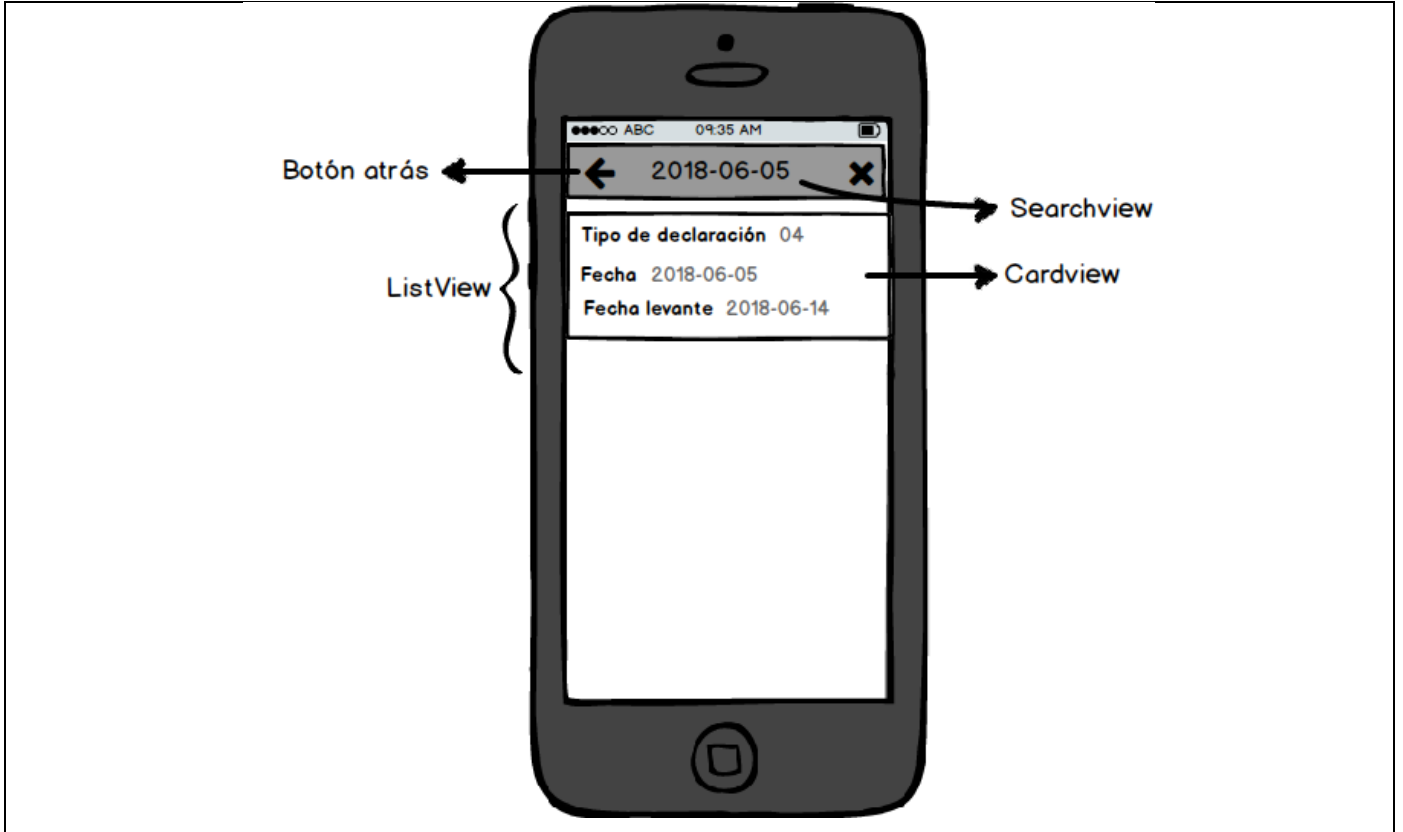
HU_ 26: Ver detalles de una declaración de mercancía

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
26	Nuevo	4	4	2	1	5
Nombre						
Ver detalles de una declaración de mercancía						
Notas						
Descripción						
El objetivo de esta actividad permitir ver los detalles de una declaración de mercancía existente en el sistema. Inicialmente se muestra una vista con las declaraciones de mercancía que han sido incluidos en el sistema. Al dar clic sobre uno de esas declaraciones de mercancía se mostrará una vista con los detalles de la declaración de mercancía seleccionada.						



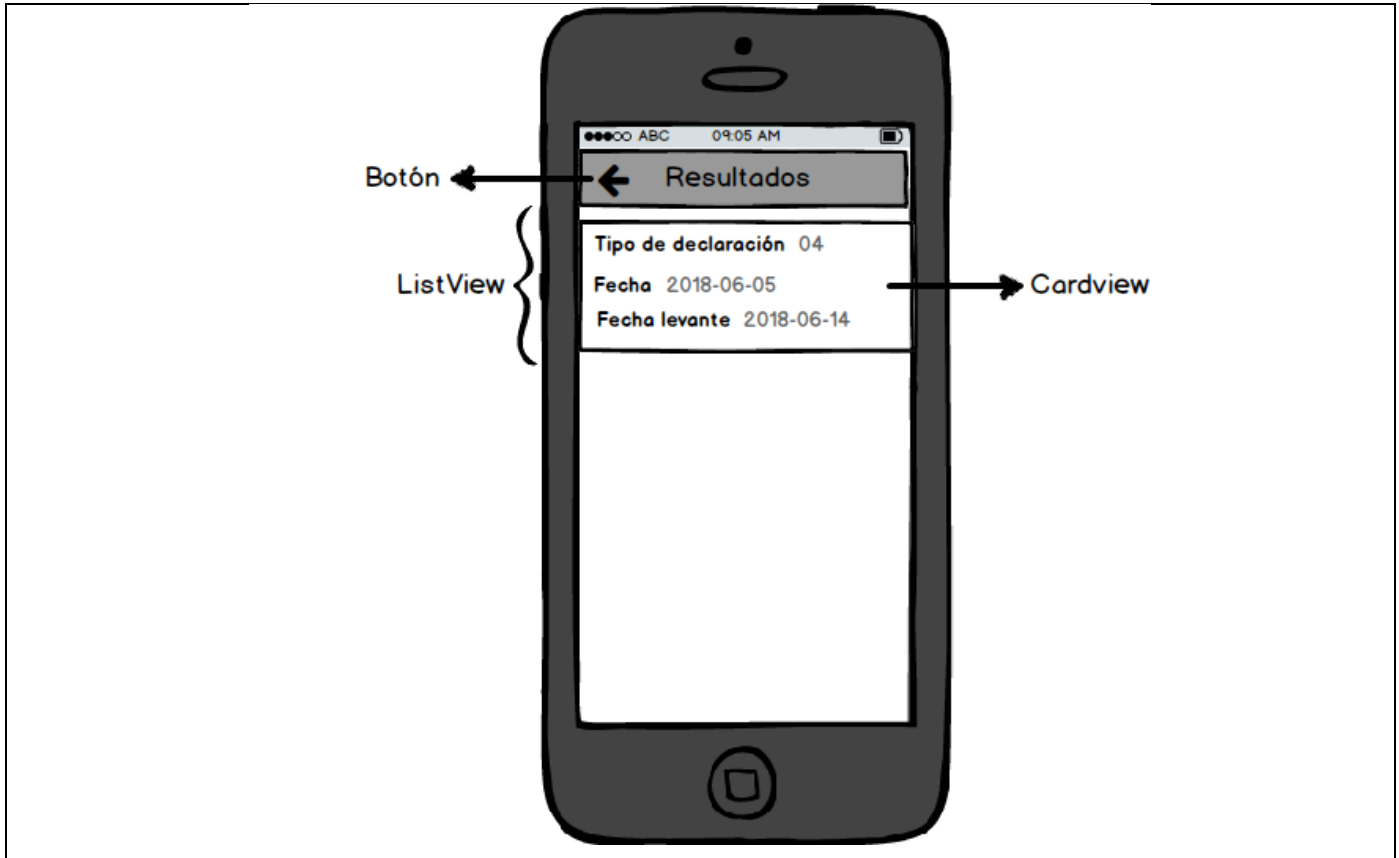
HU_ 27: Realizar búsqueda rápida de declaraciones de mercancía

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
27	Nuevo	3	3	0.5	0.5	2
Nombre						
Realizar búsqueda rápida de declaraciones de mercancía						
Notas						
Descripción						
El objetivo de esta actividad permitir al usuario realizar una búsqueda rápida sobre el listado de declaraciones de mercancía. Para ello se introducirá un texto y se mostrará un listado con las declaraciones de mercancía que coincidan con el criterio.						
Prototipo de Interfaz						



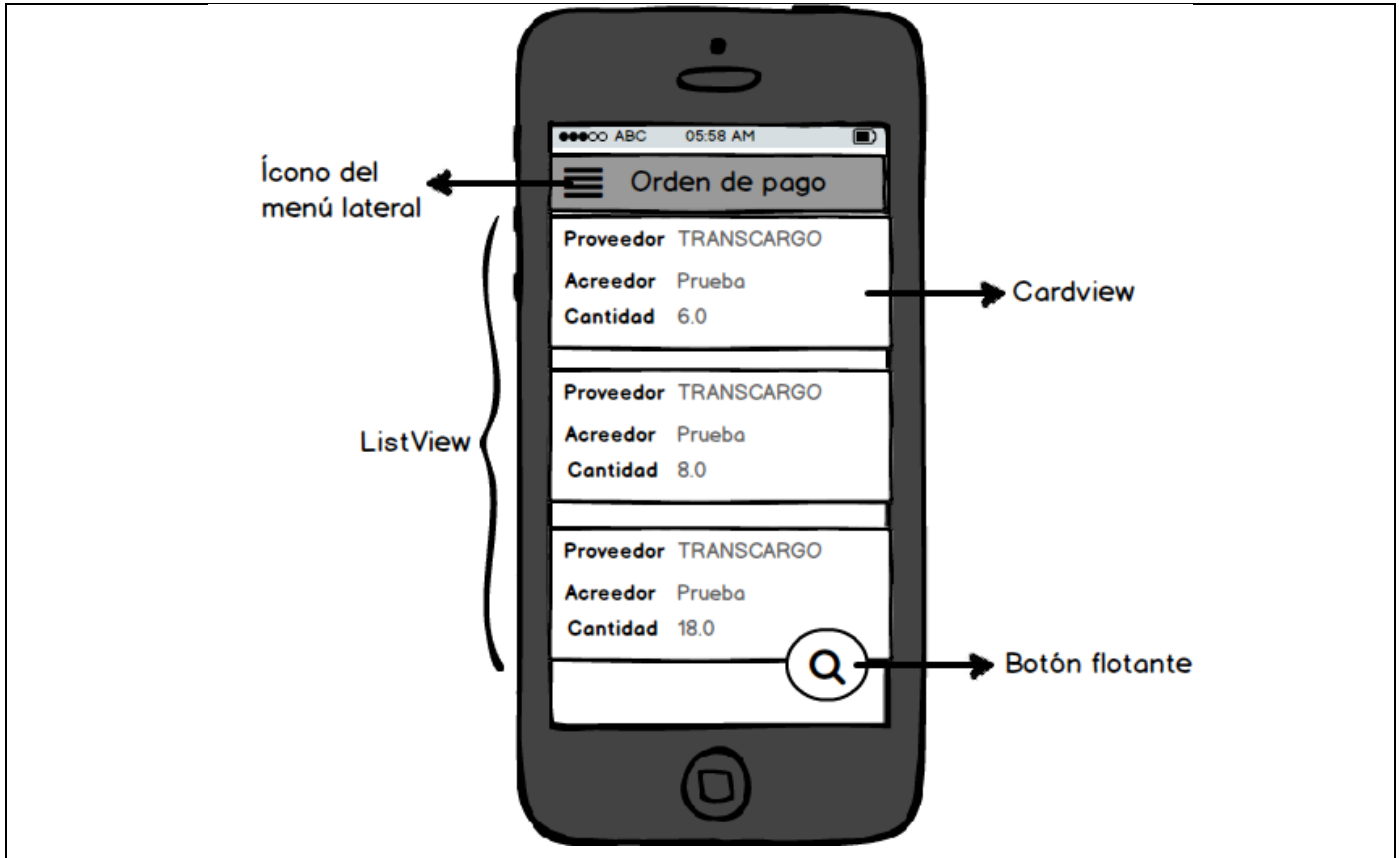
HU_ 28: Filtrar declaraciones de mercancía

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
28	Nuevo	4	4	2	2	3
Nombre						
Filtrar declaraciones de mercancía						
Notas						
Descripción						
El objetivo de esta actividad es filtrar el listado de las declaraciones de mercancía existentes en el sistema. Para ello el usuario adicionará uno o varios criterios de búsqueda y presionar el botón "Buscar", se mostrará un listado con los resultados obtenidos.						
Prototipo de Interfaz: Adicionar criterio						
Prototipo de Interfaz: Mostrar resultados						



HU_29: Listar órdenes de pago

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
29	Nuevo	4	4	2	1	5
Nombre						
Listar órdenes de pago						
Notas						
Descripción						
El objetivo de esta actividad es listar las órdenes de pago existentes en el sistema. Para ello el usuario seleccionará la opción "Orden de pago" del menú y se aparecerá un listado con los datos más descriptivos de las órdenes de pago.						
Prototipo de Interfaz						



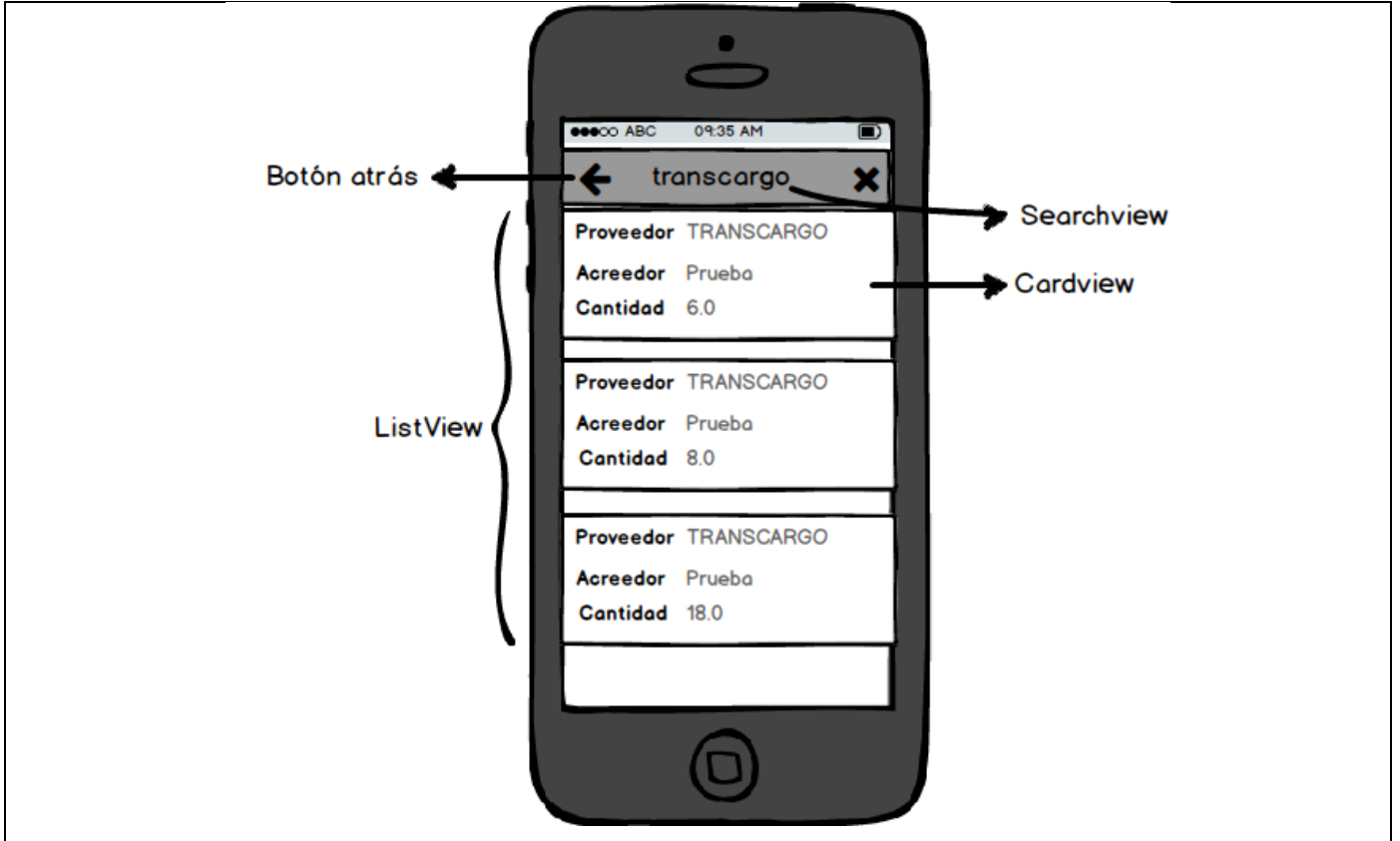
HU_ 30: Ver detalles de una orden de pago

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
30	Nuevo	4	4	2	1	5
Nombre						
Ver detalles de una orden de pago						
Notas						
Descripción						
El objetivo de esta actividad permitir ver los detalles de una orden de pago existente en el sistema. Inicialmente se muestra una vista con las órdenes de pago que han sido incluidos en el sistema. Al dar clic sobre uno de esas órdenes de pago se mostrará una vista con los detalles del orden de pago seleccionado.						
Prototipo de Interfaz						



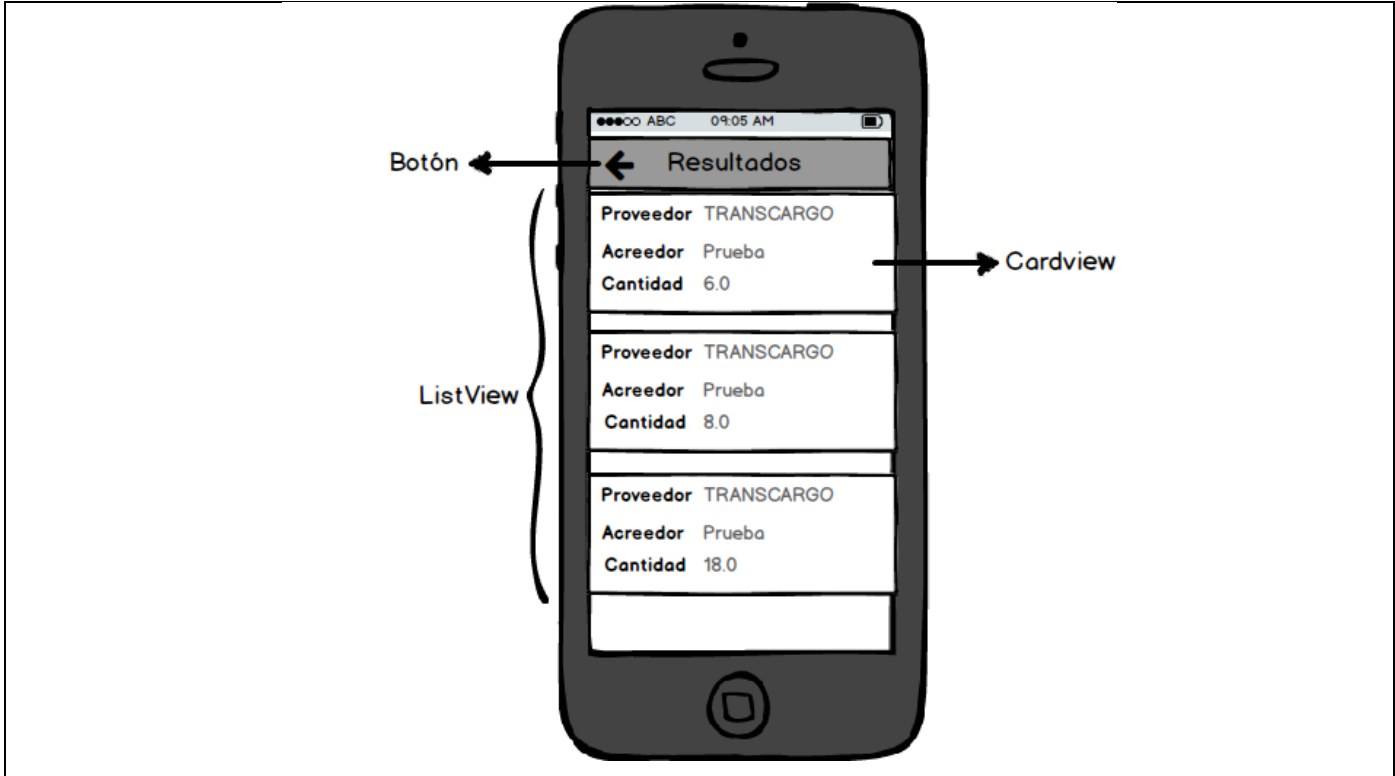
HU_ 31: Realizar búsqueda rápida de órdenes de pago

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
31	Nuevo	3	3	0.5	0.5	2
Nombre						
Realizar búsqueda rápida de órdenes de pago						
Notas						
Descripción						
El objetivo de esta actividad permitir al usuario realizar una búsqueda rápida sobre el listado de órdenes de pago. Para ello se introducirá un texto y se mostrará un listado con las órdenes de pago que coincidan con el criterio.						
Prototipo de Interfaz						



HU_ 32: Filtrar órdenes de pago

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
32	Nuevo	4	4	2	2	3
Nombre						
Filtrar órdenes de pago						
Notas						
Descripción						
El objetivo de esta actividad es filtrar el listado de las órdenes de pago existentes en el sistema. Para ello el usuario adicionará uno o varios criterios de búsqueda y presionar el botón "Buscar", se mostrará un listado con los resultados obtenidos.						
Prototipo de Interfaz: Adicionar condición						
<p>El prototipo de interfaz muestra un formulario en un dispositivo móvil. En la parte superior hay un botón 'Buscar' y un botón de atrás a la izquierda. Debajo hay un campo 'Agregar condición' con una lista desplegable 'Proveedor'. En la parte inferior hay un campo 'Proveedor' con una lista desplegable 'Contiene' que muestra el valor 'transcargo'. Hay un botón de eliminación 'X' a la derecha del campo 'Proveedor' y un botón de texto a la derecha del campo 'Contiene'.</p>						
Prototipo de Interfaz: Mostrar resultados						

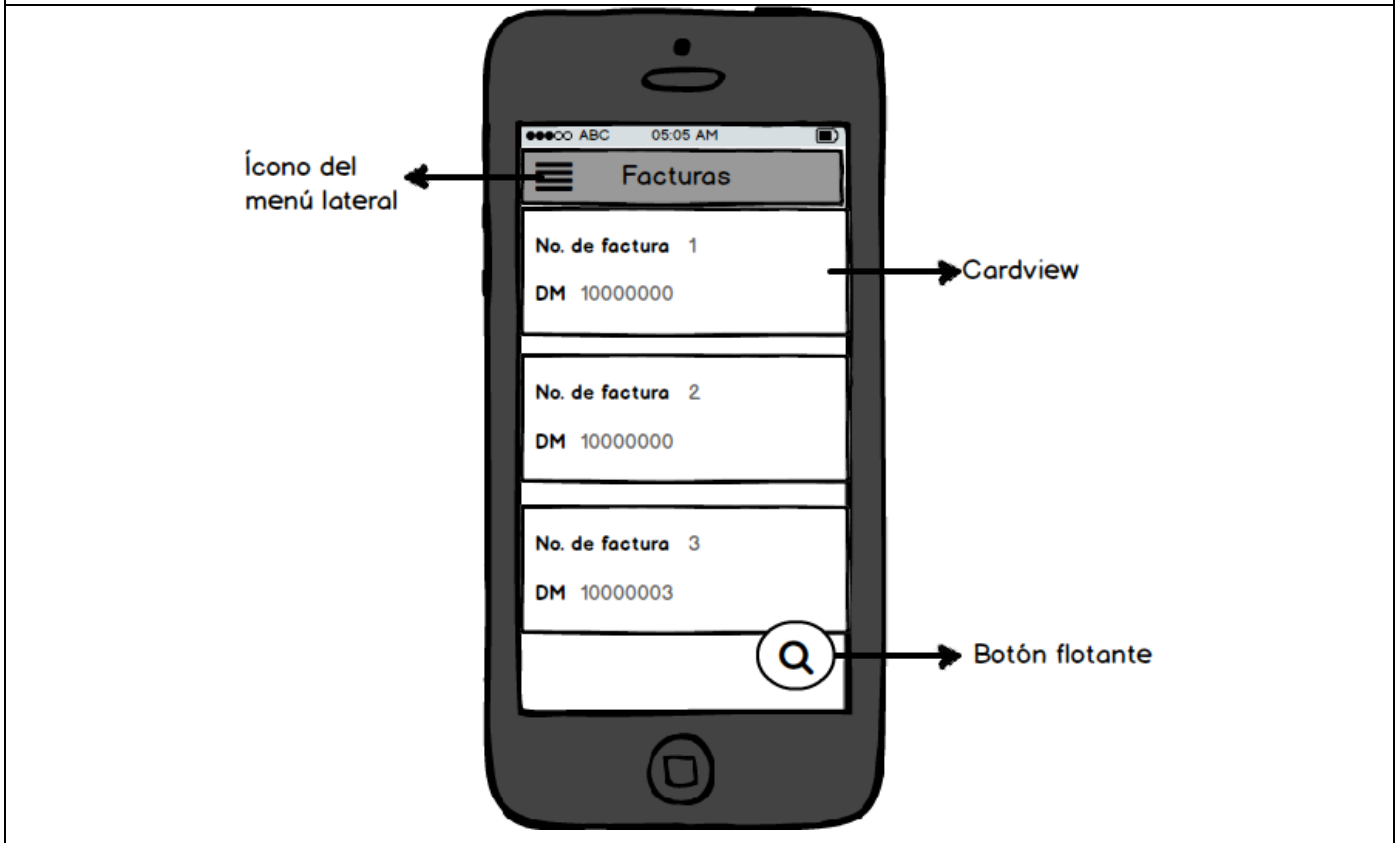


HU_33: Listar facturas

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
33	Nuevo	4	4	2	1	5
Nombre						
Listar facturas						
Notas						
Descripción						

El objetivo de esta actividad es listar las facturas existentes en el sistema. Para ello el usuario seleccionará la opción "Facturas" del menú y se aparecerá un listado con los datos más descriptivos de las facturas.

Prototipo de Interfaz

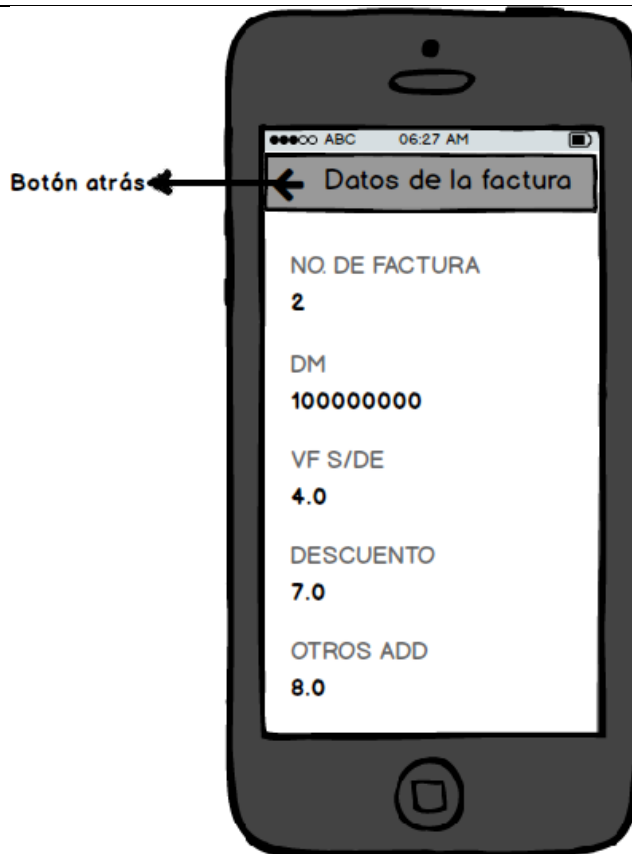


HU_ 34: Ver detalles una factura

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
34	Nuevo	4	4	2	1	5
Nombre						
Ver detalles de una factura						
Notas						
Descripción						

El objetivo de esta actividad permitir ver los detalles de una factura existente en el sistema. Inicialmente se muestra una vista con las facturas que han sido incluidos en el sistema. Al dar clic sobre uno de esas facturas se mostrará una vista con los detalles de la factura seleccionado.

Prototipo de Interfaz

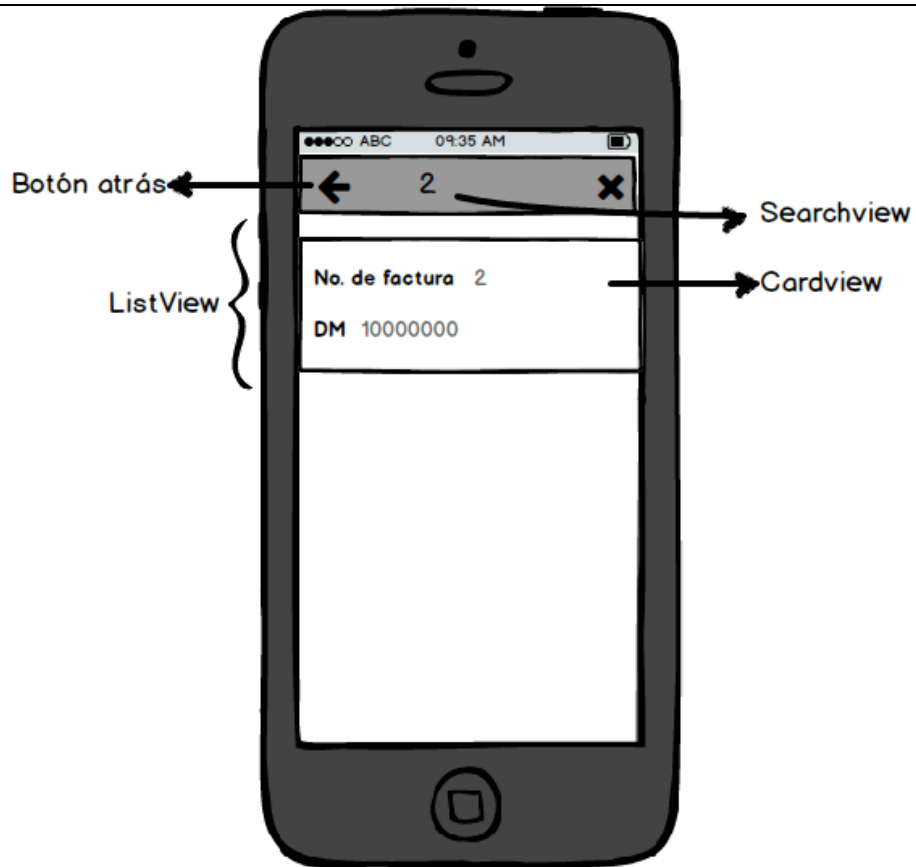


HU_ 35: Realizar búsqueda rápida de facturas

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
35	Nuevo	3	3	0.5	0.5	2
Nombre						
Realizar búsqueda rápida de facturas						
Notas						
Descripción						

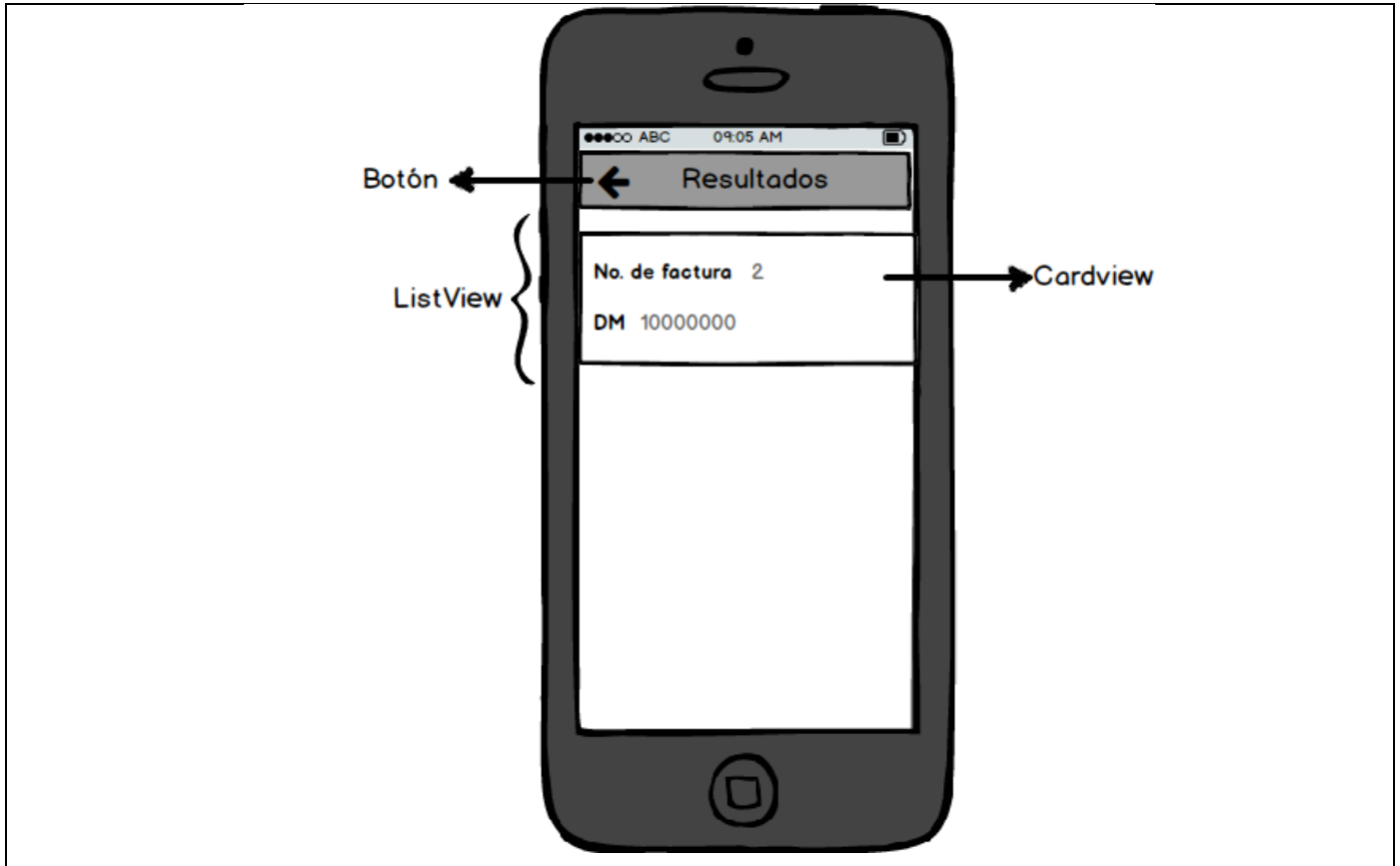
El objetivo de esta actividad permitir al usuario realizar una búsqueda rápida sobre el listado de facturas. Para ello se introducirá un texto y se mostrará un listado con las facturas que coincidan con el criterio.

Prototipo de Interfaz



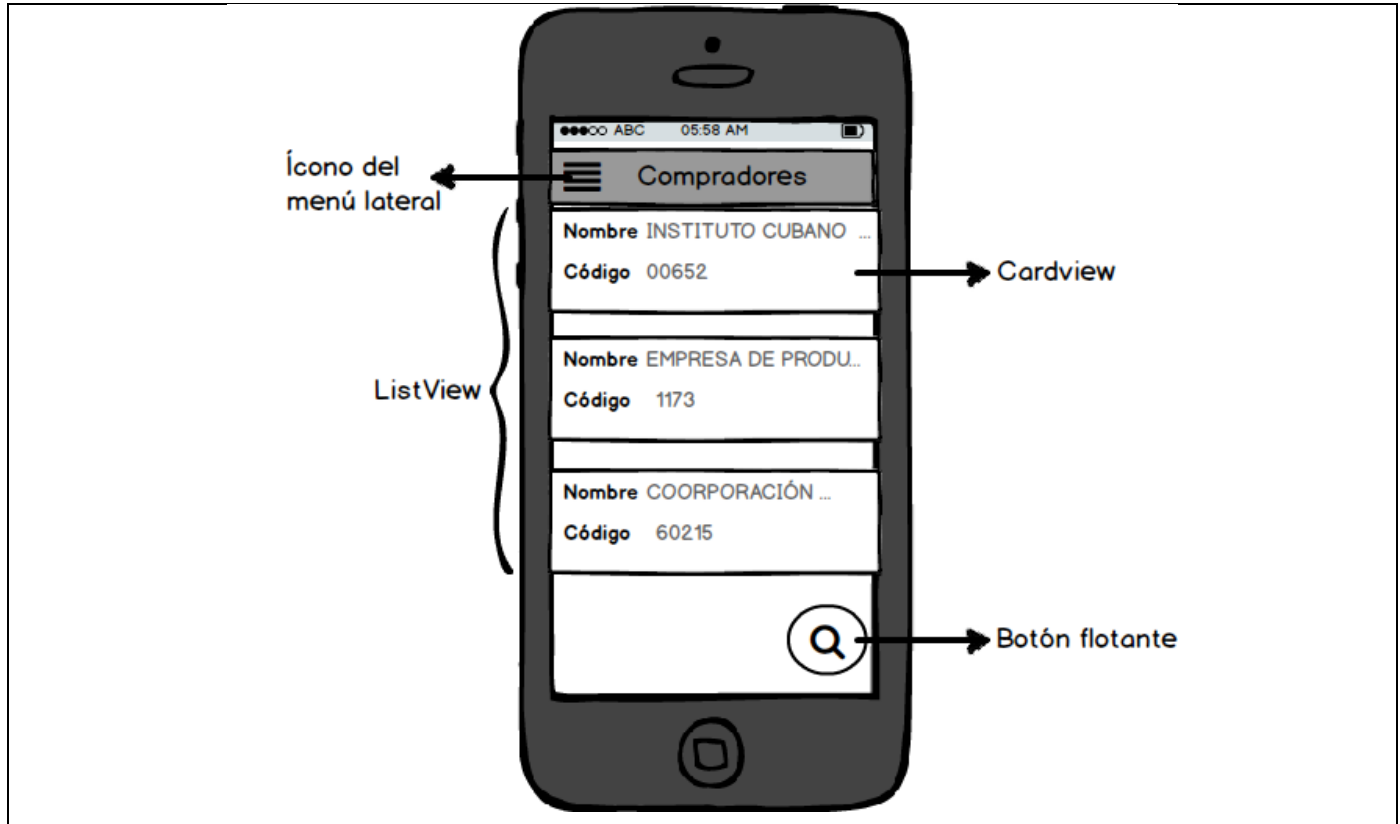
HU_ 36: Filtrar facturas

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
36	Nuevo	4	4	2	2	3
Nombre						
Filtrar facturas						
Notas						
Descripción						
El objetivo de esta actividad es filtrar el listado de facturas existentes en el sistema. Para ello el usuario adicionará uno o varios criterios de búsqueda y presionar el botón “Buscar”, se mostrará un listado con los resultados obtenidos.						
Prototipo de Interfaz: Agregar criterio						
<p>El prototipo de interfaz muestra un teléfono móvil con la siguiente estructura de elementos:</p> <ul style="list-style-type: none"> Botón atrás: Representado por una flecha hacia la izquierda en la barra superior. Botón: Representado por el texto "Buscar" en la barra superior. Lista desplegable: Un menú desplegable con el texto "Agregar condición" y "No. de factura" y una flecha hacia abajo. Botón: Un botón con un símbolo "X" que permite eliminar el criterio. Lista desplegable: Un menú desplegable con el texto "es" y una flecha hacia abajo. Campo de texto: Un campo de entrada con el número "2". 						
Prototipo de Interfaz: Mostrar resultados						



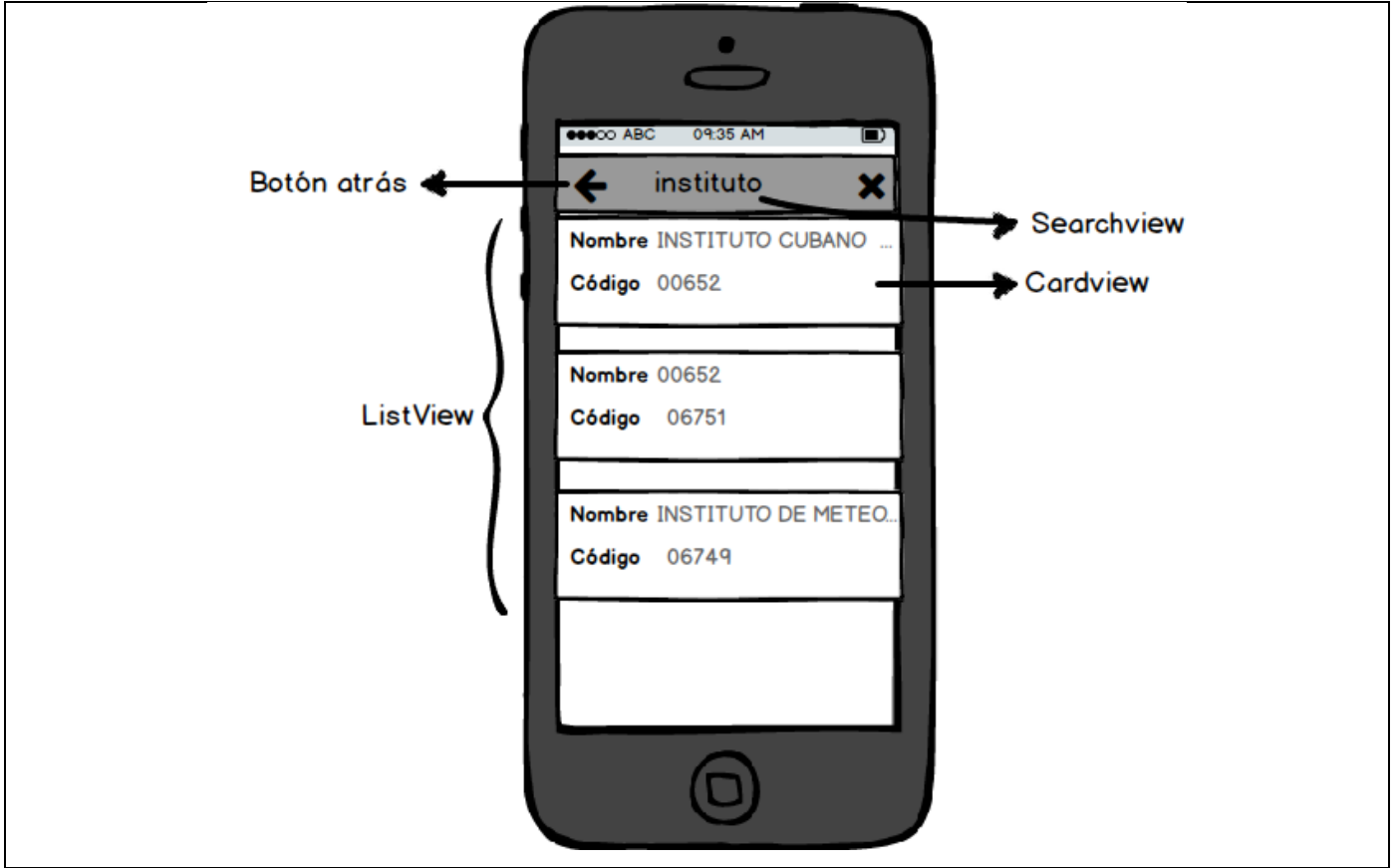
HU_37: Listar compradores

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
37	Nuevo	4	4	2	1	5
Nombre						
Listar compradores						
Notas						
Descripción						
El objetivo de esta actividad es listar los compradores existentes en el sistema. Para ello el usuario seleccionará la opción "Compradores" del menú y se aparecerá un listado con los datos más descriptivos de los compradores.						
Prototipo de Interfaz						



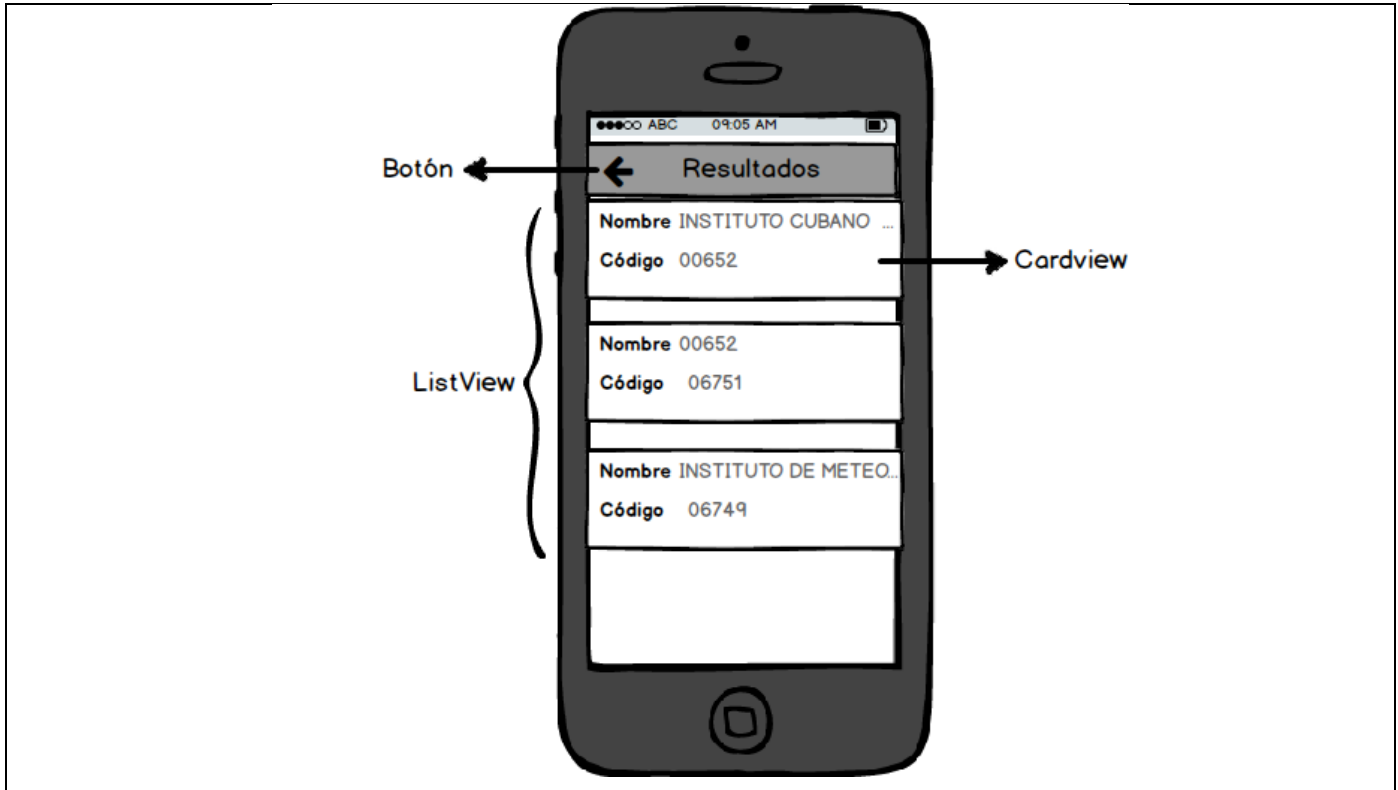
HU_ 38: Realizar búsqueda rápida de compradores

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
38	Nuevo	3	3	0.5	0.5	2
Nombre						
Realizar búsqueda rápida de compradores						
Notas						
Descripción						
El objetivo de esta actividad permitir al usuario realizar una búsqueda rápida sobre el listado de los compradores. Para ello se introducirá un texto y se mostrará un listado con los compradores que coincidan con el criterio.						
Prototipo de Interfaz						



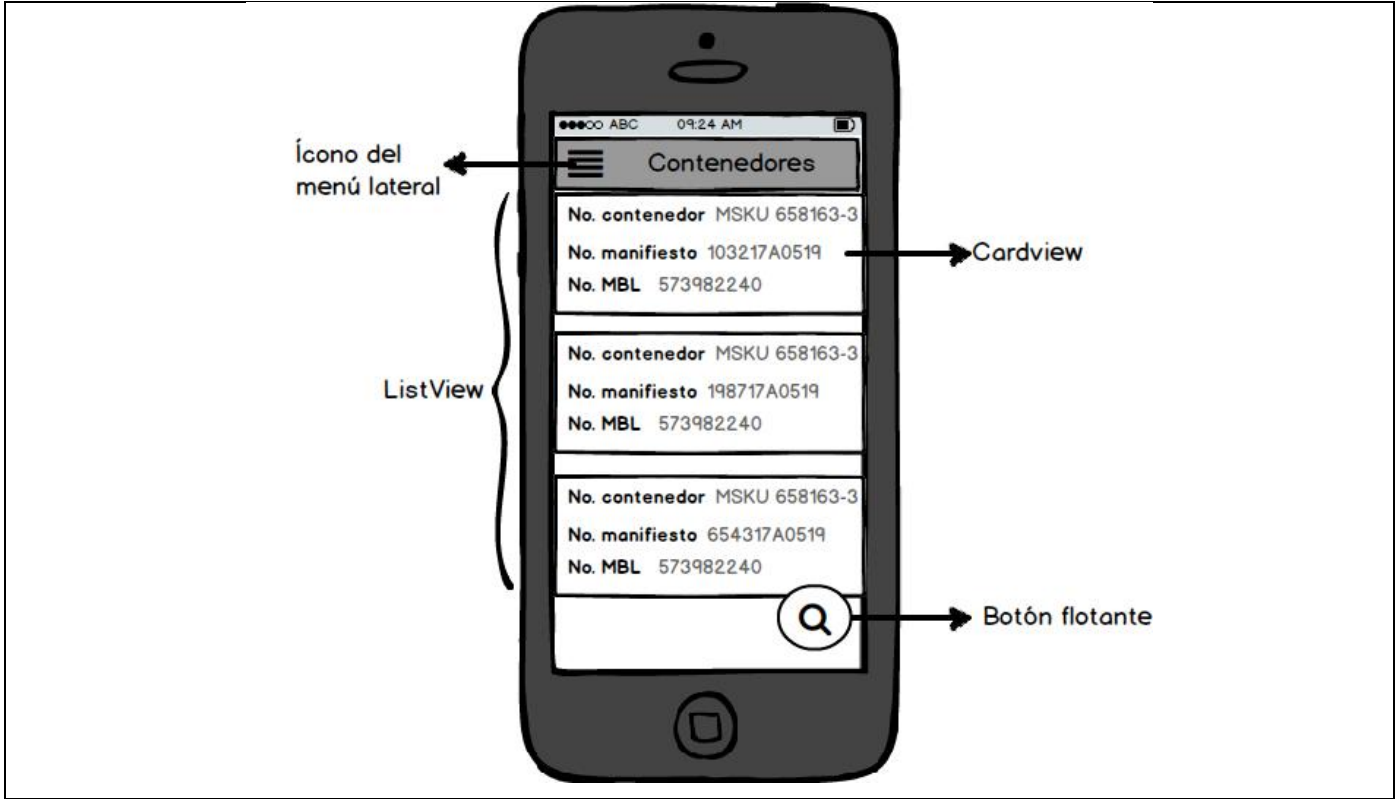
HU_ 39: Filtrar compradores

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
39	Nuevo	4	4	2	2	3
Nombre						
Filtrar compradores						
Notas						
Descripción						
El objetivo de esta actividad es filtrar el listado de los compradores existentes en el sistema. Para ello el usuario adicionará uno o varios criterios de búsqueda y presionar el botón "Buscar", se mostrará un listado con los resultados obtenidos.						
Prototipo de Interfaz: Adicionar criterio						
<p>El prototipo de interfaz muestra un dispositivo móvil con la siguiente estructura de elementos:</p> <ul style="list-style-type: none"> Encabezado: Botón "Atrás" (izquierda) y Botón "Buscar" (derecha). Sección "Agregar condición": Lista desplegable "Nombre". Formulario de búsqueda: Lista desplegable "Nombre" (mostrando "Contiene"), campo de texto "instituto", Botón "X" (eliminar) y Botón "Buscar" (ejecutar). 						
Prototipo de Interfaz: Mostrar resultados						




HU_40: Listar contenedores de logística

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
40	Nuevo	4	4	2	1	5
Nombre						
Listar contenedores de logística						
Notas						
Descripción						
El objetivo de esta actividad es listar los contenedores de logística existentes en el sistema. Para ello el usuario seleccionará la opción "Contenedores" del menú y se aparecerá un listado con los datos más descriptivos de los contenedores de logística.						
Prototipo de Interfaz						



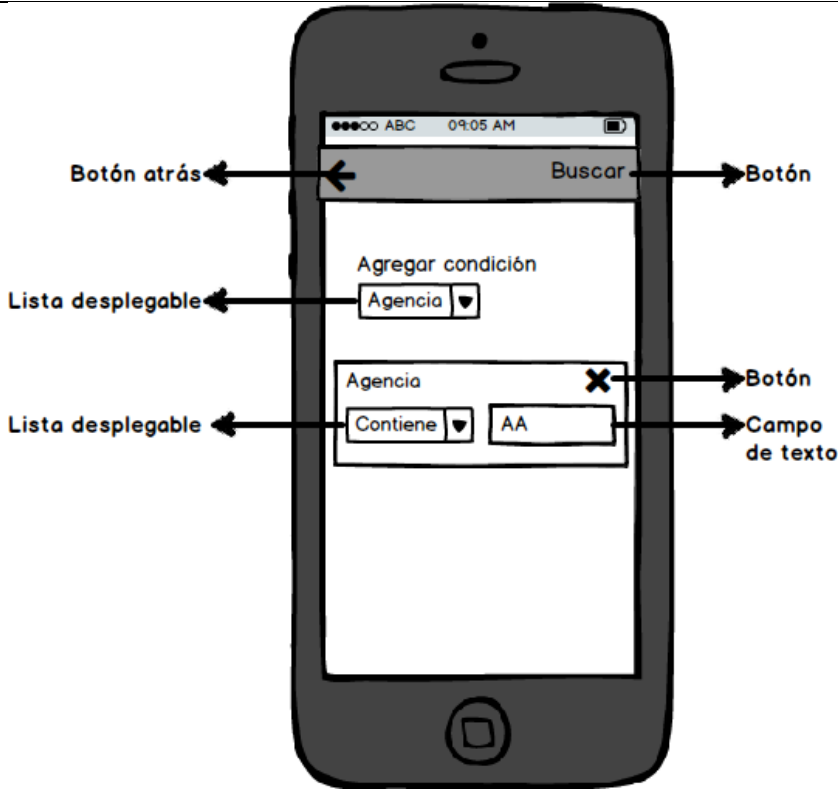
HU_ 41: Ver detalles de un contenedor de logística

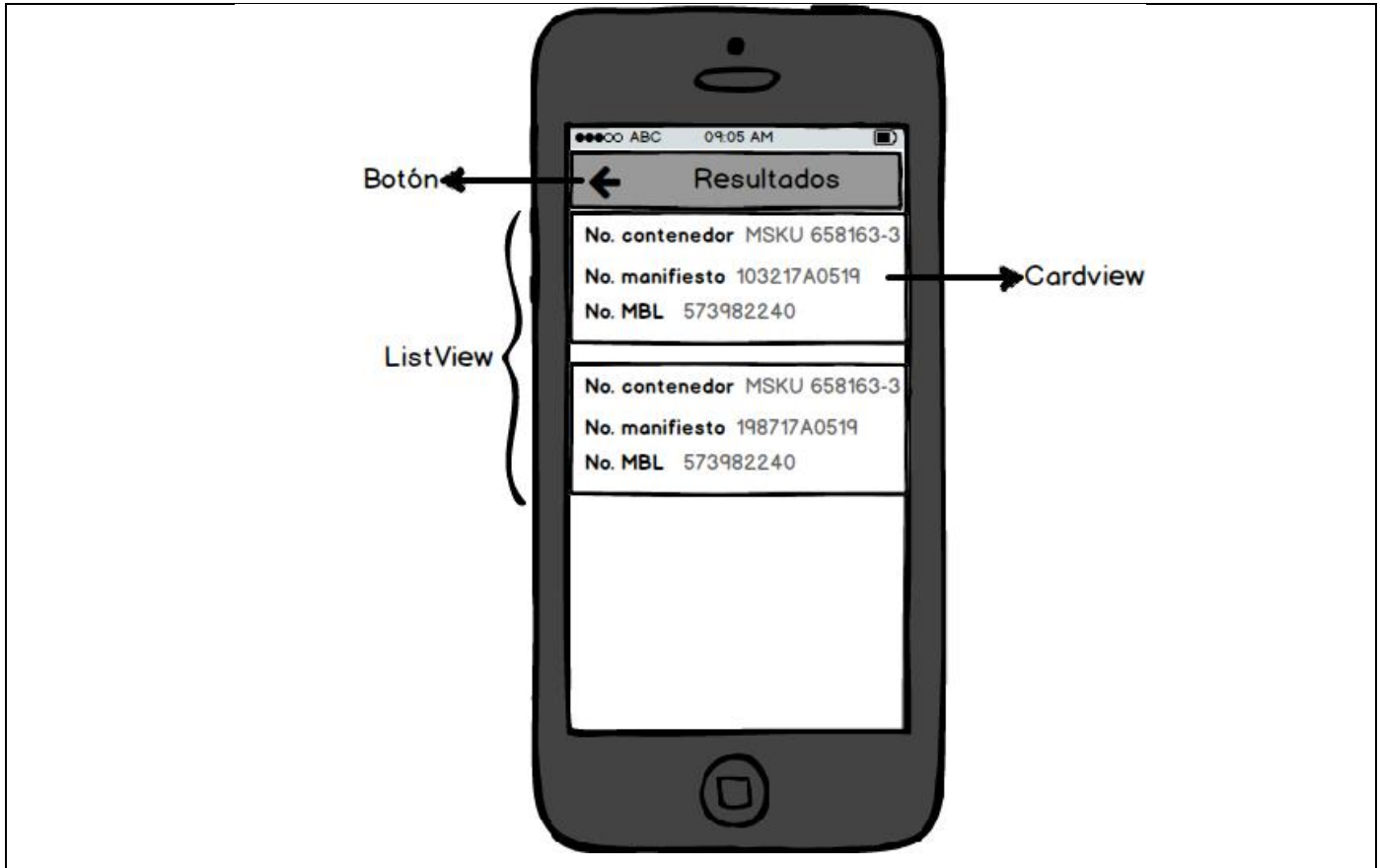
Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
41	Nuevo	4	4	2	1	5
Nombre						
Ver detalles de un contenedor de logística						
Notas						
Descripción						
<p>El objetivo de esta actividad permitir ver los detalles de un contenedor de logística existente en el sistema. Inicialmente se muestra una vista con los contenedores de logística que han sido incluidos en el sistema. Al dar clic sobre uno de esos contenedores de logística se mostrará una vista con los detalles del contenedor de logística seleccionado.</p>						
Prototipo de Interfaz						
 <p>The image shows a mobile application interface on a smartphone. The screen displays the following information: 'NO. CONTENEDOR MSKU 658163-3', 'NO. MANIFIESTO 103217A0519', 'NO. MBL 573982240', 'PESO NETO 12.0', and 'NO. BULTOS DECLARADOS 12'. At the top of the screen, there is a header bar with a back arrow icon and the text 'Datos del contenedor'. An external arrow points to this back arrow icon with the label 'Botón atrás'.</p>						

HU_ 42: Realizar búsqueda rápida de contenedores de logística

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
42	Nuevo	3	3	0.5	0.5	2
Nombre						
Realizar búsqueda rápida de contenedores de logística						
Notas						
Descripción						
El objetivo de esta actividad permitir al usuario realizar una búsqueda rápida sobre el listado de los contenedores de logística. Para ello se introducirá un texto y se mostrará un listado con los contenedores de logística que coincidan con el criterio.						
Prototipo de Interfaz						

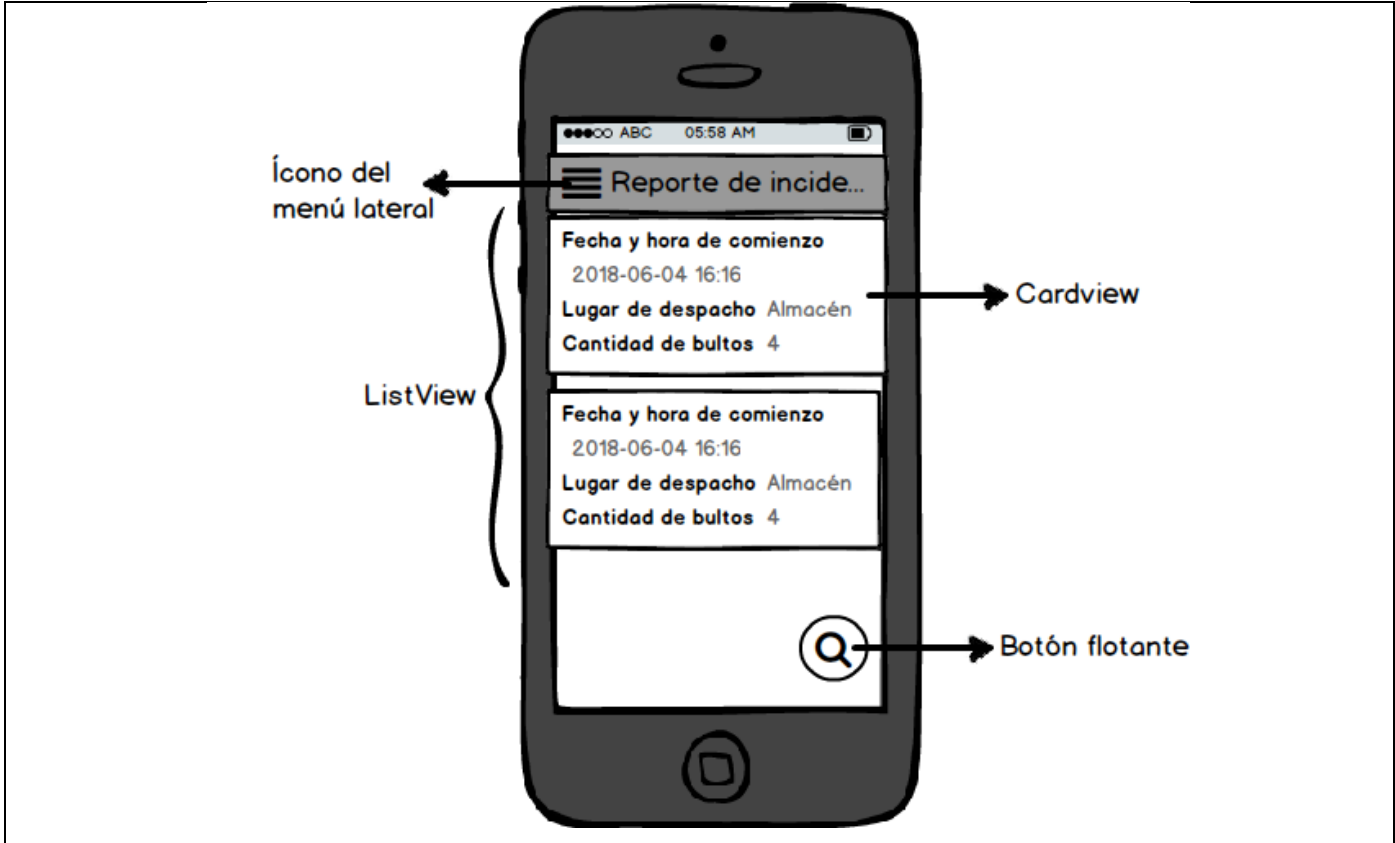
HU_ 43: Filtrar contenedores de logística

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
43	Nuevo	4	4	2	2	3
Nombre						
Filtrar contenedores de logística						
Notas						
Descripción						
El objetivo de esta actividad es filtrar el listado de los contenedores de logística existentes en el sistema. Para ello el usuario adicionará uno o varios criterios de búsqueda y presionar el botón “Buscar”, se mostrará un listado con los resultados obtenidos.						
Prototipo de Interfaz: Adicionar criterio						
						
Prototipo de Interfaz: Mostrar resultados						

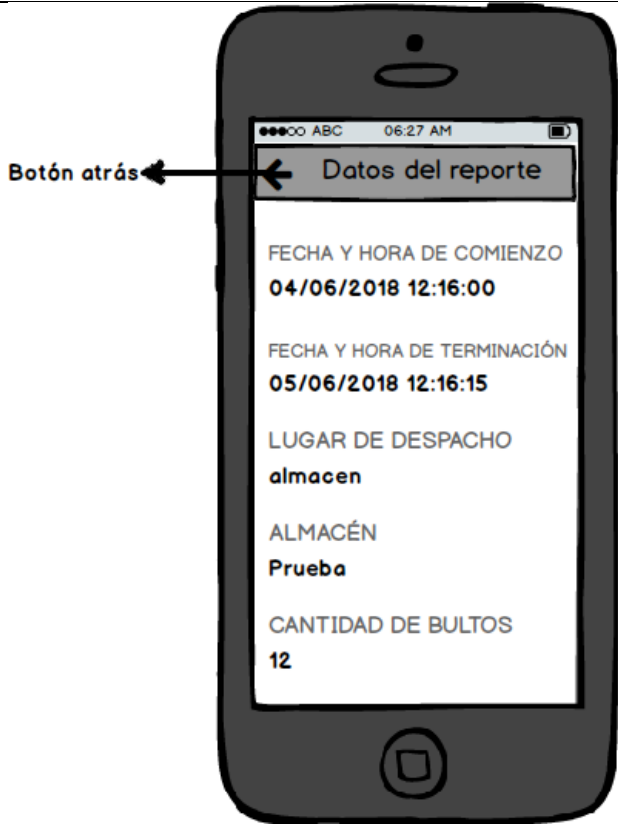


HU_44: Listar reportes de incidencias

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
44	Nuevo	4	4	2	1	5
Nombre						
Listar reportes de incidencias						
Notas						
Descripción						
El objetivo de esta actividad es listar los reportes de incidencias existentes en el sistema. Para ello el usuario seleccionará la opción “Reporte de incidencias” del menú y se aparecerá un listado con los datos más descriptivos de los reportes de incidencias.						
Prototipo de Interfaz						



HU_ 45: Ver detalles de un reporte de incidencias

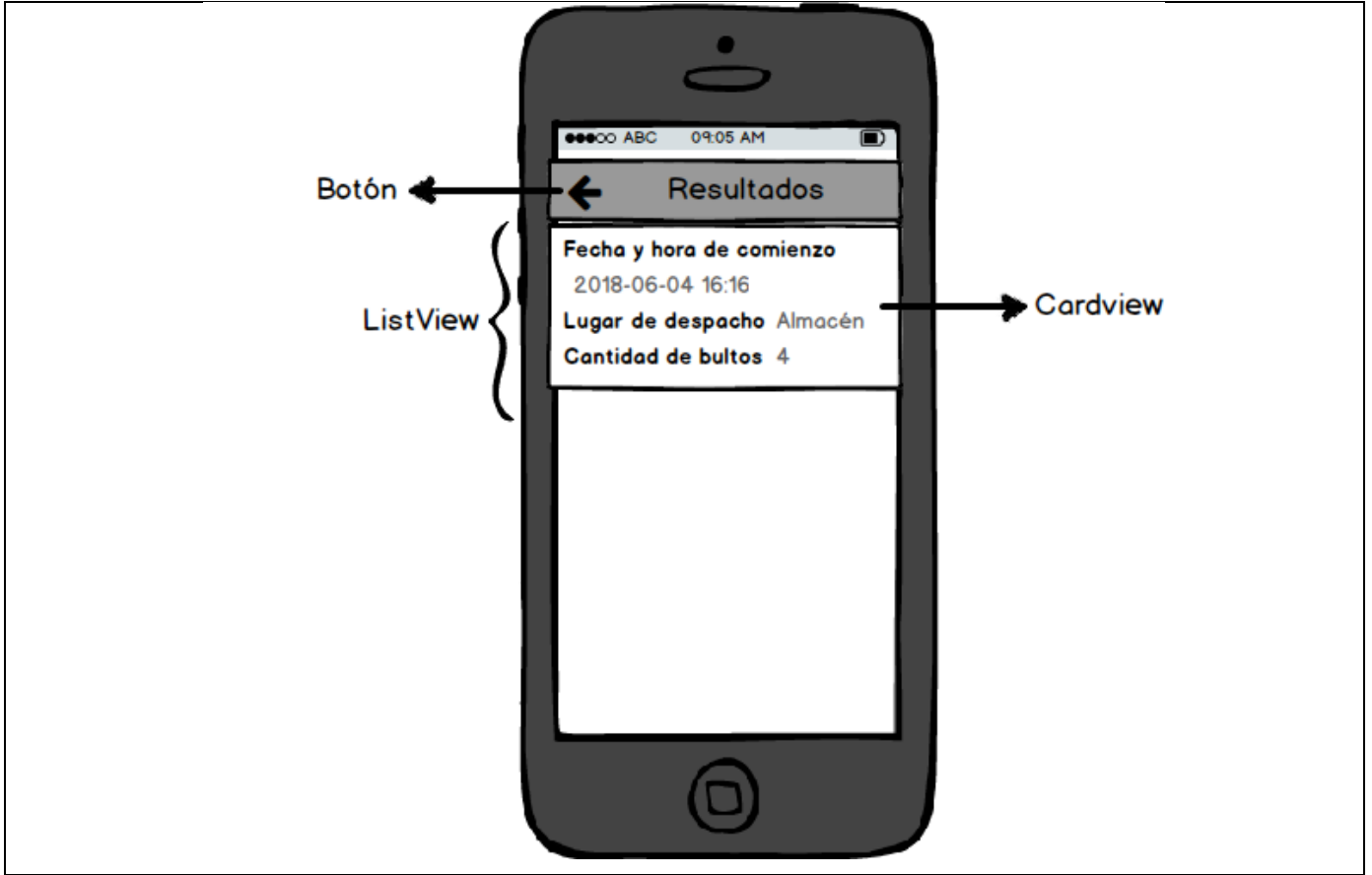
Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
45	Nuevo	4	4	2	1	5
Nombre						
Ver detalles de un reporte de incidencias						
Notas						
Descripción						
El objetivo de esta actividad permitir ver los detalles de un reporte de incidencias existente en el sistema. Inicialmente se muestra una vista con los reportes de incidencias que han sido incluidos en el sistema. Al dar clic sobre uno de esos reportes de incidencias se mostrará una vista con los detalles del reporte de incidencias seleccionado.						
Prototipo de Interfaz						
						

HU_ 46: Realizar búsqueda rápida de reportes de incidencias

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
42	Nuevo	3	3	0.5	0.5	2
Nombre						
Realizar búsqueda rápida de reportes de incidencias						
Notas						
Descripción						
El objetivo de esta actividad permitir al usuario realizar una búsqueda rápida sobre el listado de los reportes de incidencias. Para ello se introducirá un texto y se mostrará un listado con los reportes de incidencias que coincidan con el criterio.						
Prototipo de Interfaz						

HU_ 47: Filtrar contenedores de logística

Historia de usuario						
Número	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
47	Nuevo	4	4	2	2	3
Nombre						
Filtrar reportes de incidencias						
Notas						
Descripción						
El objetivo de esta actividad es filtrar el listado de los reportes de incidencias existentes en el sistema. Para ello el usuario adicionará uno o varios criterios de búsqueda y presionar el botón "Buscar", se mostrará un listado con los resultados obtenidos.						
Prototipo de Interfaz: Adicionar condición						
<p>Botón atrás ←</p> <p>→ Botón</p> <p>Lista desplegable ←</p> <p>→ Botón</p> <p>Lista desplegable ←</p> <p>→ Campo de texto</p>						
Prototipo de Interfaz: Mostrar resultados						



Anexo 2: Pruebas de aceptación

CPA_ 1: Listar conductores

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_1	HISTORIA DE USUARIO: HU_1
DESCRIPCIÓN: prueba la funcionalidad de listar conductores.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un conductor. • El usuario debe poseer los permisos de transporte. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Una vez autenticado el usuario se selecciona la opción “Conductor”.	
RESULTADO ESPERADO:	
Aparece un listado con todos los conductores en el sistema. En el listado se deben mostrar los siguientes datos:	
<ul style="list-style-type: none"> • Licencia • Nombre y apellidos • CI 	
De no existir ningún conductor en el sistema se mostrará un mensaje.	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 2: Ver un conductor

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_2	HISTORIA DE USUARIO: HU_2
DESCRIPCIÓN: prueba la funcionalidad de ver un conductor.	

<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un conductor. • El usuario debe poseer los permisos de transporte.
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Inicialmente se mostrará un listado con los conductores. El usuario seleccionará uno de los elementos de la lista.</p>
<p>RESULTADO ESPERADO:</p> <p>Se mostrará una vista con los siguientes datos del conductor:</p> <ul style="list-style-type: none"> • Licencia • Nombre y apellidos • CI • Carta operativa • Número de accidentes • Tasa de accidentes • Dirección • Estado civil • Fecha de control médico • Resultado del control médico • Fecha de reclasificación
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_3: Realizar búsqueda rápida de conductores

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_3	HISTORIA DE USUARIO: HU_3
DESCRIPCIÓN: prueba la funcionalidad de realizar una búsqueda rápida de conductores.	
<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. 	

<ul style="list-style-type: none"> • Debe existir en el sistema al menos un conductor. • El usuario debe poseer los permisos de transporte.
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>El usuario introducirá un texto sobre el cual realizar una búsqueda rápida.</p>
<p>RESULTADO ESPERADO:</p> <p>Se mostrará los resultados para la búsqueda rápida basándose en los siguientes criterios:</p> <ul style="list-style-type: none"> • Licencia • Nombre y apellidos • CI <p>Se devuelve un listado con aquellos conductores que cumplan con el criterio de búsqueda. En caso de no existir ninguna coincidencia se mostrará un mensaje al usuario.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 4.1: Adicionar criterio de búsqueda para filtrar conductores

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_4.1	HISTORIA DE USUARIO: HU_4
DESCRIPCIÓN: prueba la funcionalidad de realizar criterio de búsqueda para filtrar conductores.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un conductor. • El usuario debe poseer los permisos de transporte. 	
ENTRADA/PASOS DE EJECUCIÓN:	
<p>Inicialmente se muestra un listado con los conductores. Al seleccionar la opción “Buscar” se muestra un formulario. El usuario selecciona la opción “Agregar condición”. Los posibles criterios serán:</p> <ul style="list-style-type: none"> • Licencia • Nombre y Apellidos 	

- CI
- Carta operativa
- Número de accidentes
- Tasa de accidentes
- Dirección
- Estado civil
- Fecha de control médico
- Fecha de reclasificación

RESULTADO ESPERADO:

Se adiciona un criterio de búsqueda. Cada criterio constará de un criterio de comparación y un campo donde introducir el valor.

EVALUACIÓN DE LA PRUEBA: SATISFACTORIA

CPA_ 4.2: Realizar búsqueda filtrada de conductores

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_4.2	HISTORIA DE USUARIO: HU_4
DESCRIPCIÓN: prueba la funcionalidad de realizar búsqueda filtrada de conductores.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un conductor. • El usuario debe poseer los permisos de transporte. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Inicialmente se muestra la vista donde el usuario ya ha definido los criterios de búsqueda. El usuario entonces seleccionará la opción buscar.	
RESULTADO ESPERADO:	

Se mostrará una lista con los que cumplan las condiciones definidas en los criterios de búsqueda. De no existir coincidencias se mostrará un mensaje al usuario. Si el usuario no adicionó ningún criterio se mostrará un mensaje.

EVALUACIÓN DE LA PRUEBA: SATISFACTORIA

CPA_ 5: Listar vehículos

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_5	HISTORIA DE USUARIO: HU_5
DESCRIPCIÓN: prueba la funcionalidad de listar vehículos.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un vehículo. • El usuario debe poseer los permisos de transporte. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Una vez autenticado el usuario se selecciona la opción "Vehículo".	
RESULTADO ESPERADO:	
Aparece un listado con todos los vehículos en el sistema. En el listado se deben mostrar los siguientes datos:	
<ul style="list-style-type: none"> • Modelo • Chapa de matrícula • Nombre del conductor • Tipo de vehículo 	
De no existir ningún vehículo en el sistema se mostrará un mensaje.	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 6: Ver un vehículo

CASO DE PRUEBA DE ACEPTACIÓN

CÓDIGO: CPA_6	HISTORIA DE USUARIO: HU_6
DESCRIPCIÓN: prueba la funcionalidad de ver un vehículo.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un vehículo. • El usuario debe poseer los permisos de transporte. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Inicialmente se mostrará un listado con los vehículos. El usuario seleccionará uno de los elementos de la lista.	
RESULTADO ESPERADO:	
Se mostrará una vista con los siguientes datos del vehículo:	
<ul style="list-style-type: none"> • Modelo • Chapa de matrícula • Nombre del conductor • Tipo de vehículo • Tipo de batería • Estado • Tipo de combustible • Fecha de fabricación • Fecha de FICAO • Fecha de LOT 	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 7: Realizar búsqueda rápida de vehículos

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_7	HISTORIA DE USUARIO: HU_7
DESCRIPCIÓN: prueba la funcionalidad de realizar una búsqueda rápida de vehículos.	

<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un vehículo. • El usuario debe poseer los permisos de transporte.
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>El usuario introducirá un texto sobre el cual realizar una búsqueda rápida.</p>
<p>RESULTADO ESPERADO:</p> <p>Se mostrará los resultados para la búsqueda rápida basándose en los siguientes criterios:</p> <ul style="list-style-type: none"> • Modelo • Chapa de matrícula • Nombre del conductor • Tipo de vehículo <p>Se devuelve un listado con aquellos vehículos que cumplan con el criterio de búsqueda. En caso de no existir ninguna coincidencia se mostrará un mensaje al usuario.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 8.1: Adicionar criterio de búsqueda para filtrar vehículos

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_8.1	HISTORIA DE USUARIO: HU_8
DESCRIPCIÓN: prueba la funcionalidad de realizar criterio de búsqueda para filtrar vehículos.	
<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un vehículo. • El usuario debe poseer los permisos de transporte. 	
<p>ENTRADA/PASOS DE EJECUCIÓN:</p>	

Inicialmente se muestra un listado con los vehículos. Al seleccionar la opción “Buscar” se muestra un formulario. El usuario selecciona la opción “Agregar condición”. Los posibles criterios serán:

- Modelo
- Nombre del conductor
- Chapa de matrícula
- Tipo de vehículo
- Tipo de batería
- Estado
- Tipo de combustible
- Fecha de fabricación
- Fecha de FICAO
- Fecha de LOT

RESULTADO ESPERADO:

Se adiciona un criterio de búsqueda. Cada criterio constará de un criterio de comparación y un campo donde introducir el valor.

EVALUACIÓN DE LA PRUEBA: SATISFACTORIA

CPA_ 8.2: Realizar búsqueda filtrada de vehículos

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_8.2	HISTORIA DE USUARIO: HU_8
DESCRIPCIÓN: prueba la funcionalidad de realizar búsqueda filtrada de vehículos.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un vehículo. • El usuario debe poseer los permisos de transporte. 	
ENTRADA/PASOS DE EJECUCIÓN:	

Inicialmente se muestra la vista donde el usuario ya ha definido los criterios de búsqueda. El usuario entonces seleccionará la opción buscar.
RESULTADO ESPERADO: Se mostrará una lista con los que cumplan las condiciones definidas en los criterios de búsqueda. De no existir coincidencias se mostrará un mensaje al usuario. Si el usuario no adicionó ningún criterio se mostrará un mensaje.
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA

CPA_ 9: Listar multas

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_9	HISTORIA DE USUARIO: HU_9
DESCRIPCIÓN: prueba la funcionalidad de listar multas.	
CONDICIONES DE EJECUCIÓN: <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una multa. • El usuario debe poseer los permisos de transporte. 	
ENTRADA/PASOS DE EJECUCIÓN: Una vez autenticado el usuario se selecciona la opción “Multa”.	
RESULTADO ESPERADO: Aparece un listado con todas las multas en el sistema. En el listado se deben mostrar los siguientes datos: <ul style="list-style-type: none"> • Licencia • Conductor • Puntos de la licencia • Monto en CUP De no existir ninguna multa en el sistema se mostrará un mensaje.	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 10: Ver una multa

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_10	HISTORIA DE USUARIO: HU_10
DESCRIPCIÓN: prueba la funcionalidad de ver una multa.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una multa. • El usuario debe poseer los permisos de transporte. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Inicialmente se mostrará un listado con las multas. El usuario seleccionará uno de los elementos de la lista.	
RESULTADO ESPERADO:	
Se mostrará una vista con los siguientes datos de la multa:	
<ul style="list-style-type: none"> • Conductor • Licencia • Fecha • Puntos de la licencia • Monto en CUP • Causa 	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 11: Realizar búsqueda rápida de multas

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_11	HISTORIA DE USUARIO: HU_11
DESCRIPCIÓN: prueba la funcionalidad de realizar una búsqueda rápida de multas.	

<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una multa. • El usuario debe poseer los permisos de transporte.
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>El usuario introducirá un texto sobre el cual realizar una búsqueda rápida.</p>
<p>RESULTADO ESPERADO:</p> <p>Se mostrará los resultados para la búsqueda rápida basándose en los siguientes criterios:</p> <ul style="list-style-type: none"> • Licencia • Conductor • Puntos de la licencia • Monto en CUP <p>Se devuelve un listado con aquellas multas que cumplan con el criterio de búsqueda. En caso de no existir ninguna coincidencia se mostrará un mensaje al usuario.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 12.1: Adicionar criterio de búsqueda para filtrar multas

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_12.1	HISTORIA DE USUARIO: HU_12
DESCRIPCIÓN: prueba la funcionalidad de realizar criterio de búsqueda para filtrar vehículos.	
<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una multa. • El usuario debe poseer los permisos de transporte. 	
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Inicialmente se muestra un listado con las multas. Al seleccionar la opción “Buscar” se muestra un formulario. El usuario selecciona la opción “Agregar condición”. Los posibles criterios serán:</p>	

<ul style="list-style-type: none"> • Licencia • Conductor • Puntos de la licencia • Monto en CUP • Fecha • Causa
<p>RESULTADO ESPERADO:</p> <p>Se adiciona un criterio de búsqueda. Cada criterio constará de un criterio de comparación y un campo donde introducir el valor.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_12.2: Realizar búsqueda filtrada de multas

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_12.2	HISTORIA DE USUARIO: HU_12
DESCRIPCIÓN: prueba la funcionalidad de realizar búsqueda filtrada de multas.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una multa. • El usuario debe poseer los permisos de transporte. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Inicialmente se muestra la vista donde el usuario ya ha definido los criterios de búsqueda. El usuario entonces seleccionará la opción buscar.	
RESULTADO ESPERADO:	
Se mostrará una lista con los que cumplan las condiciones definidas en los criterios de búsqueda. De no existir coincidencias se mostrará un mensaje al usuario. Si el usuario no adicionó ningún criterio se mostrará un mensaje.	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 13: Listar contenedores disponibles

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_13	HISTORIA DE USUARIO: HU_13
DESCRIPCIÓN: prueba la funcionalidad de listar contenedores disponibles.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor disponible. • El usuario debe poseer los permisos de arrendamiento de contenedores. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Una vez autenticado el usuario se selecciona la opción “Contenedores disponibles”.	
RESULTADO ESPERADO:	
Aparece un listado con todos los contenedores disponibles en el sistema. En el listado se deben mostrar los siguientes datos:	
<ul style="list-style-type: none"> • No. • Longitud • Tipo • Ubicación 	
De no existir ningún contenedor disponible en el sistema se mostrará un mensaje.	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 14: Ver contenedor disponible

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_14	HISTORIA DE USUARIO: HU_14
DESCRIPCIÓN: prueba la funcionalidad de ver un contenedor disponible.	

<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor disponible. • El usuario debe poseer los permisos de arrendamiento de contenedores.
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Inicialmente se mostrará un listado con las contenedores disponibles. El usuario seleccionará uno de los elementos de la lista.</p>
<p>RESULTADO ESPERADO:</p> <p>Se mostrará una vista con los siguientes datos del contenedor disponible:</p> <ul style="list-style-type: none"> • No. de contenedor • Longitud • Tipo de contenedor • Proveedor • Estado • Ubicación • Observación
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 15: Realizar búsqueda rápida de contenedores disponibles

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_15	HISTORIA DE USUARIO: HU_15
DESCRIPCIÓN: prueba la funcionalidad de realizar una búsqueda rápida de contenedores disponible.	
<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor disponible. • El usuario debe poseer los permisos de arrendamiento de contenedores. 	
ENTRADA/PASOS DE EJECUCIÓN:	

El usuario introducirá un texto sobre el cual realizar una búsqueda rápida.
<p>RESULTADO ESPERADO:</p> <p>Se mostrará los resultados para la búsqueda rápida basándose en los siguientes criterios:</p> <ul style="list-style-type: none"> • No. • Longitud • Tipo • Ubicación <p>Se devuelve un listado con aquellos contenedores disponibles que cumplan con el criterio de búsqueda. En caso de no existir ninguna coincidencia se mostrará un mensaje al usuario.</p>
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA

CPA_ 16.1: Adicionar criterio de búsqueda para filtrar contenedores disponibles

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_16.1	HISTORIA DE USUARIO: HU_16
DESCRIPCIÓN: prueba la funcionalidad de realizar criterio de búsqueda para filtrar contenedores disponibles.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor disponible. • El usuario debe poseer los permisos de arrendamiento de contenedores. 	
ENTRADA/PASOS DE EJECUCIÓN:	
<p>Inicialmente se muestra un listado con los contenedores disponibles. Al seleccionar la opción “Buscar” se muestra un formulario. El usuario selecciona la opción “Agregar condición”. Los posibles criterios serán:</p> <ul style="list-style-type: none"> • No. de contenedor • Longitud • Tipo de contenedor • Ubicación • Proveedor • Estado 	

<ul style="list-style-type: none"> Observación
<p>RESULTADO ESPERADO:</p> <p>Se adiciona un criterio de búsqueda. Cada criterio constará de un criterio de comparación y un campo donde introducir el valor.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_16.2: Realizar búsqueda filtrada de contenedores disponibles

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_16.2	HISTORIA DE USUARIO: HU_16
<p>DESCRIPCIÓN: prueba la funcionalidad de realizar búsqueda filtrada de contenedores disponibles.</p>	
<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> Estar autenticado en el sistema. Debe existir en el sistema al menos un contenedor disponible. El usuario debe poseer los permisos de arrendamiento de contenedores. 	
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Inicialmente se muestra la vista donde el usuario ya ha definido los criterios de búsqueda. El usuario entonces seleccionará la opción buscar.</p>	
<p>RESULTADO ESPERADO:</p> <p>Se mostrará una lista con los que cumplan las condiciones definidas en los criterios de búsqueda. De no existir coincidencias se mostrará un mensaje al usuario. Si el usuario no adicionó ningún criterio se mostrará un mensaje.</p>	
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>	

CPA_17: Listar contenedores arrendados

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_17	HISTORIA DE USUARIO: HU_17

DESCRIPCIÓN: prueba la funcionalidad de listar contenedores arrendados.
CONDICIONES DE EJECUCIÓN: <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor arrendado. • El usuario debe poseer los permisos de arrendamiento de contenedores.
ENTRADA/PASOS DE EJECUCIÓN: <p>Una vez autenticado el usuario se selecciona la opción “Contenedores arrendados”.</p>
RESULTADO ESPERADO: <p>Aparece un listado con todos los contenedores arrendados en el sistema. En el listado se deben mostrar los siguientes datos:</p> <ul style="list-style-type: none"> • No. • Longitud • Tipo • Ubicación <p>De no existir ningún contenedor arrendado en el sistema se mostrará un mensaje.</p>
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA

CPA_ 18: Ver contenedor arrendado

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_18	HISTORIA DE USUARIO: HU_18
DESCRIPCIÓN: prueba la funcionalidad de ver un contenedor arrendado.	
CONDICIONES DE EJECUCIÓN: <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor arrendado. • El usuario debe poseer los permisos de arrendamiento de contenedores. 	

<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Inicialmente se mostrará un listado con los contenedores arrendados. El usuario seleccionará uno de los elementos de la lista.</p>
<p>RESULTADO ESPERADO:</p> <p>Se mostrará una vista con los siguientes datos del contenedor arrendado:</p> <ul style="list-style-type: none"> • No. de contenedor • Longitud • Tipo de contenedor • Proveedor • Cliente • Estado • Ubicación • Depósito • Observación • Fecha de arrendamiento • Fecha de retorno
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 19: Realizar búsqueda rápida de contenedores arrendados

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_19	HISTORIA DE USUARIO: HU_19
DESCRIPCIÓN: prueba la funcionalidad de realizar una búsqueda rápida de contenedores arrendados.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor arrendado. • El usuario debe poseer los permisos de arrendamiento de contenedores. 	
ENTRADA/PASOS DE EJECUCIÓN:	
El usuario introducirá un texto sobre el cual realizar una búsqueda rápida.	

<p>RESULTADO ESPERADO:</p> <p>Se mostrará los resultados para la búsqueda rápida basándose en los siguientes criterios:</p> <ul style="list-style-type: none"> • No. • Longitud • Tipo • Ubicación <p>Se devuelve un listado con aquellos contenedores arrendados que cumplan con el criterio de búsqueda. En caso de no existir ninguna coincidencia se mostrará un mensaje al usuario.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 20.1: Realizar búsqueda filtrada de contenedores arrendados

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_20.1	HISTORIA DE USUARIO: HU_20
DESCRIPCIÓN: prueba la funcionalidad de realizar criterio de búsqueda para filtrar contenedores arrendados.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor arrendado. • El usuario debe poseer los permisos de arrendamiento de contenedores. 	
ENTRADA/PASOS DE EJECUCIÓN:	
<p>Inicialmente se muestra un listado con los contenedores arrendados. Al seleccionar la opción “Buscar” se muestra un formulario. El usuario selecciona la opción “Agregar condición”. Los posibles criterios serán:</p> <ul style="list-style-type: none"> • No. de contenedor • Longitud • Tipo de contenedor • Ubicación • Proveedor • Cliente • Estado 	

<ul style="list-style-type: none"> • Depósito • Observación • Fecha de arrendamiento • Fecha de retorno
<p>RESULTADO ESPERADO:</p> <p>Se adiciona un criterio de búsqueda. Cada criterio constará de un criterio de comparación y un campo donde introducir el valor.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_20.2: Realizar búsqueda filtrada de contenedores arrendados

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_20.2	HISTORIA DE USUARIO: HU_20
DESCRIPCIÓN: prueba la funcionalidad de realizar búsqueda filtrada de contenedores arrendados.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor arrendado. • El usuario debe poseer los permisos de arrendamiento de contenedores. 	
ENTRADA/PASOS DE EJECUCIÓN:	
<p>Inicialmente se muestra la vista donde el usuario ya ha definido los criterios de búsqueda. El usuario entonces seleccionará la opción buscar.</p>	
RESULTADO ESPERADO:	
<p>Se mostrará una lista con los que cumplan las condiciones definidas en los criterios de búsqueda. De no existir coincidencias se mostrará un mensaje al usuario. Si el usuario no adicionó ningún criterio se mostrará un mensaje.</p>	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_21: Listar contenedores fuera de servicio

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_21	HISTORIA DE USUARIO: HU_21
DESCRIPCIÓN: prueba la funcionalidad de listar contenedores fuera de servicio.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor fuera de servicio. • El usuario debe poseer los permisos de arrendamiento de contenedores. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Una vez autenticado el usuario se selecciona la opción “Contenedores fuera de servicios”.	
RESULTADO ESPERADO:	
Aparece un listado con todos los contenedores fuera de servicios en el sistema. En el listado se deben mostrar los siguientes datos:	
<ul style="list-style-type: none"> • No. • Longitud • Tipo • Ubicación 	
De no existir ningún contenedor fuera de servicio en el sistema se mostrará un mensaje.	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_22: Ver un contenedor fuera de servicio

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_22	HISTORIA DE USUARIO: HU_22
DESCRIPCIÓN: prueba la funcionalidad de ver un contenedor fuera de servicio.	

<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor fuera de servicio. • El usuario debe poseer los permisos de arrendamiento de contenedores.
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Inicialmente se mostrará un listado con los contenedores fuera de servicios. El usuario seleccionará uno de los elementos de la lista.</p>
<p>RESULTADO ESPERADO:</p> <p>Se mostrará una vista con los siguientes datos del contenedor arrendado:</p> <ul style="list-style-type: none"> • No. de contenedor • Longitud • Tipo de contenedor • Estado • Ubicación • Observación
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_23: Realizar búsqueda rápida de contenedores fuera de servicio

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_23	HISTORIA DE USUARIO: HU_23
DESCRIPCIÓN: prueba la funcionalidad de realizar una búsqueda rápida de contenedores fuera de servicio.	
<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor fuera de servicio. • El usuario debe poseer los permisos de arrendamiento de contendores. 	
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>El usuario introducirá un texto sobre el cual realizar una búsqueda rápida.</p>	

<p>RESULTADO ESPERADO:</p> <p>Se mostrará los resultados para la búsqueda rápida basándose en los siguientes criterios:</p> <ul style="list-style-type: none"> • No. • Longitud • Tipo • Ubicación <p>Se devuelve un listado con aquellos contenedores fuera de servicio que cumplan con el criterio de búsqueda. En caso de no existir ninguna coincidencia se mostrará un mensaje al usuario.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_24.1: Realizar búsqueda filtrada de contenedores fuera de servicio

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_24.1	HISTORIA DE USUARIO: HU_24
DESCRIPCIÓN: prueba la funcionalidad de realizar criterio de búsqueda para filtrar contenedores fuera de servicio.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor fuera de servicio. • El usuario debe poseer los permisos de arrendamiento de contenedores. 	
ENTRADA/PASOS DE EJECUCIÓN:	
<p>Inicialmente se muestra un listado con los contenedores fuera de servicio. Al seleccionar la opción “Buscar” se muestra un formulario. El usuario selecciona la opción “Agregar condición”. Los posibles criterios serán:</p> <ul style="list-style-type: none"> • No. de contenedor • Longitud • Tipo de contenedor • Ubicación • Estado • Observación 	

<p>RESULTADO ESPERADO:</p> <p>Se adiciona un criterio de búsqueda. Cada criterio constará de un criterio de comparación y un campo donde introducir el valor.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 24.2: Realizar búsqueda filtrada de contenedores fuera de servicio

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_24.2	HISTORIA DE USUARIO: HU_24
DESCRIPCIÓN: prueba la funcionalidad de realizar búsqueda filtrada de contenedores fuera de servicio.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor fuera de servicio. • El usuario debe poseer los permisos de arrendamiento de contenedores. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Inicialmente se muestra la vista donde el usuario ya ha definido los criterios de búsqueda. El usuario entonces seleccionará la opción buscar.	
RESULTADO ESPERADO:	
Se mostrará una lista con los que cumplan las condiciones definidas en los criterios de búsqueda. De no existir coincidencias se mostrará un mensaje al usuario. Si el usuario no adicionó ningún criterio se mostrará un mensaje.	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 25: Listar declaraciones de mercancía

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_25	HISTORIA DE USUARIO: HU_25

<p>DESCRIPCIÓN: prueba la funcionalidad de listar declaraciones de mercancía.</p>
<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una declaración de mercancía. • El usuario debe poseer los permisos de agenciamiento aduanal.
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Una vez autenticado el usuario se selecciona la opción “Declaración de mercancía”.</p>
<p>RESULTADO ESPERADO:</p> <p>Aparece un listado con todas las declaraciones de mercancía en el sistema. En el listado se deben mostrar los siguientes datos:</p> <ul style="list-style-type: none"> • Tipo de declaración • Fecha • Fecha de levante <p>De no existir ninguna declaración de mercancía en el sistema se mostrará un mensaje.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_26: Ver una declaración de mercancía

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_26	HISTORIA DE USUARIO: HU_26
<p>DESCRIPCIÓN: prueba la funcionalidad de ver una declaración de mercancía.</p>	
<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una declaración de mercancía. • El usuario debe poseer los permisos de agenciamiento aduanal. 	
<p>ENTRADA/PASOS DE EJECUCIÓN:</p>	

<p>Inicialmente se mostrará un listado con las declaraciones de mercancía. El usuario seleccionará uno de los elementos de la lista.</p>
<p>RESULTADO ESPERADO:</p> <p>Se mostrará una vista con los siguientes datos de la declaración de mercancía:</p> <ul style="list-style-type: none"> • Tipo de declaración de mercancía • Declaración de tránsito • No. de manifiesto • No. BL/Guía aérea • Cantidad de facturas • Fecha • Fecha levante
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_27: Realizar búsqueda rápida de declaración de mercancía

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_27	HISTORIA DE USUARIO: HU_27
DESCRIPCIÓN: prueba la funcionalidad de realizar una búsqueda rápida de declaraciones de mercancía.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una declaración de mercancía. • El usuario debe poseer los permisos de agenciamiento aduanal. 	
ENTRADA/PASOS DE EJECUCIÓN:	
El usuario introducirá un texto sobre el cual realizar una búsqueda rápida.	
RESULTADO ESPERADO:	
Se mostrará los resultados para la búsqueda rápida basándose en los siguientes criterios:	
<ul style="list-style-type: none"> • Tipo de declaración • Fecha 	

- Fecha Levante

Se devuelve un listado con aquellas declaraciones de mercancía que cumplan con el criterio de búsqueda. En caso de no existir ninguna coincidencia se mostrará un mensaje al usuario.

EVALUACIÓN DE LA PRUEBA: SATISFACTORIA

CPA_ 28.1: Realizar búsqueda filtrada de declaraciones de mercancía

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_28.1	HISTORIA DE USUARIO: HU_28
DESCRIPCIÓN: prueba la funcionalidad de realizar criterio de búsqueda para filtrar declaraciones de mercancía.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una declaración de mercancía. • El usuario debe poseer los permisos de agenciamiento aduanal. 	
ENTRADA/PASOS DE EJECUCIÓN:	
<p>Inicialmente se muestra un listado con las declaraciones de mercancía. Al seleccionar la opción “Buscar” se muestra un formulario. El usuario selecciona la opción “Agregar condición”. Los posibles criterios serán:</p> <ul style="list-style-type: none"> • Tipo de declaración • Declaración de tránsito • No. de manifiesto • No. BL/ Guía aérea • Cantidad de facturas • Fecha • Fecha de levante 	
RESULTADO ESPERADO:	
<p>Se adiciona un criterio de búsqueda. Cada criterio constará de un criterio de comparación y un campo donde introducir el valor.</p>	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 28.2: Realizar búsqueda filtrada de declaraciones de mercancía

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_28.2	HISTORIA DE USUARIO: HU_28
DESCRIPCIÓN: prueba la funcionalidad de realizar búsqueda filtrada de declaraciones de mercancía.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una declaración de mercancía. • El usuario debe poseer los permisos de agenciamiento aduanal. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Inicialmente se muestra la vista donde el usuario ya ha definido los criterios de búsqueda. El usuario entonces seleccionará la opción buscar.	
RESULTADO ESPERADO:	
Se mostrará una lista con los que cumplan las condiciones definidas en los criterios de búsqueda. De no existir coincidencias se mostrará un mensaje al usuario. Si el usuario no adicionó ningún criterio se mostrará un mensaje.	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 29: Listar órdenes de pago

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_29	HISTORIA DE USUARIO: HU_29
DESCRIPCIÓN: prueba la funcionalidad de listar órdenes de pago.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una orden de pago. • El usuario debe poseer los permisos de facturación. 	

<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Una vez autenticado el usuario se selecciona la opción “Declaración de mercancía”.</p>
<p>RESULTADO ESPERADO:</p> <p>Aparece un listado con todas las órdenes de pago en el sistema. En el listado se deben mostrar los siguientes datos:</p> <ul style="list-style-type: none"> • Proveedor • Acreedor • Cantidad • Fecha <p>De no existir ninguna orden de pago en el sistema se mostrará un mensaje.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_30: Ver una orden de pago

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_30	HISTORIA DE USUARIO: HU_30
DESCRIPCIÓN: prueba la funcionalidad de ver una orden de pago.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una orden de pago. • El usuario debe poseer los permisos de facturación. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Inicialmente se mostrará un listado con las órdenes de pago. El usuario seleccionará uno de los elementos de la lista.	
RESULTADO ESPERADO:	
Se mostrará una vista con los siguientes datos de la orden de pago:	
<ul style="list-style-type: none"> • Proveedor • Acreedor • Acuerdo 	

<ul style="list-style-type: none"> • Cantidad • Importe • Observaciones • Fecha • GAR
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA

CPA_31: Realizar una búsqueda rápida de órdenes de pago

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_31	HISTORIA DE USUARIO: HU_31
DESCRIPCIÓN: prueba la funcionalidad de realizar una búsqueda rápida de órdenes de pago.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una orden de pago. • El usuario debe poseer los permisos de facturación. 	
ENTRADA/PASOS DE EJECUCIÓN:	
El usuario introducirá un texto sobre el cual realizar una búsqueda rápida.	
RESULTADO ESPERADO:	
Se mostrará los resultados para la búsqueda rápida basándose en los siguientes criterios:	
<ul style="list-style-type: none"> • Proveedor • Acreedor • Cantidad • Fecha 	
Se devuelve un listado con aquellas órdenes de pago que cumplan con el criterio de búsqueda. En caso de no existir ninguna coincidencia se mostrará un mensaje al usuario.	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 32.1: Realizar búsqueda filtrada de órdenes de pago

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_32.1	HISTORIA DE USUARIO: HU_32
DESCRIPCIÓN: prueba la funcionalidad de realizar criterio de búsqueda para filtrar órdenes de pago.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una orden de pago. • El usuario debe poseer los permisos de facturación. 	
ENTRADA/PASOS DE EJECUCIÓN:	
<p>Inicialmente se muestra un listado con las órdenes de pago. Al seleccionar la opción “Buscar” se muestra un formulario. El usuario selecciona la opción “Agregar condición”. Los posibles criterios serán:</p> <ul style="list-style-type: none"> • Proveedor • Acreedor • Acuerdo • Cantidad • Importe • Observaciones • Fecha • GAR 	
RESULTADO ESPERADO:	
Se adiciona un criterio de búsqueda. Cada criterio constará de un criterio de comparación y un campo donde introducir el valor.	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 32.2: Realizar búsqueda filtrada de órdenes de pago

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_32.2	HISTORIA DE USUARIO: HU_32

DESCRIPCIÓN: prueba la funcionalidad de realizar búsqueda filtrada de órdenes de pago.
CONDICIONES DE EJECUCIÓN: <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una orden de pago. • El usuario debe poseer los permisos de facturación.
ENTRADA/PASOS DE EJECUCIÓN: Inicialmente se muestra la vista donde el usuario ya ha definido los criterios de búsqueda. El usuario entonces seleccionará la opción buscar.
RESULTADO ESPERADO: Se mostrará una lista con los que cumplan las condiciones definidas en los criterios de búsqueda. De no existir coincidencias se mostrará un mensaje al usuario. Si el usuario no adicionó ningún criterio se mostrará un mensaje.
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA

CPA_ 33: Listar facturas

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_33	HISTORIA DE USUARIO: HU_33
DESCRIPCIÓN: prueba la funcionalidad de listar facturas.	
CONDICIONES DE EJECUCIÓN: <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una factura. • El usuario debe poseer los permisos de agenciamiento aduanal. 	
ENTRADA/PASOS DE EJECUCIÓN: Una vez autenticado el usuario se selecciona la opción "Facturas".	
RESULTADO ESPERADO:	

<p>Aparece un listado con todas las facturas en el sistema. En el listado se deben mostrar los siguientes datos:</p> <ul style="list-style-type: none"> • No. de factura • DM <p>De no existir ninguna factura en el sistema se mostrará un mensaje.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 34: Ver factura

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_34	HISTORIA DE USUARIO: HU_34
DESCRIPCIÓN: prueba la funcionalidad de ver una factura.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una factura. • El usuario debe poseer los permisos de agenciamiento aduanal. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Inicialmente se mostrará un listado con las facturas. El usuario seleccionará uno de los elementos de la lista.	
RESULTADO ESPERADO:	
Se mostrará una vista con los siguientes datos de la factura:	
<ul style="list-style-type: none"> • No. de factura • DM • VF S/DE • Descuento • Otros add • Flete • Otros gastos • Seguro 	
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>	

CPA_ 35: Realizar búsqueda rápida de facturas

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_35	HISTORIA DE USUARIO: HU_35
DESCRIPCIÓN: prueba la funcionalidad de realizar una búsqueda rápida de facturas.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una factura. • El usuario debe poseer los permisos de agenciamiento aduanal. 	
ENTRADA/PASOS DE EJECUCIÓN:	
El usuario introducirá un texto sobre el cual realizar una búsqueda rápida.	
RESULTADO ESPERADO:	
Se mostrará los resultados para la búsqueda rápida basándose en los siguientes criterios:	
<ul style="list-style-type: none"> • No. de factura • DM <p>Se devuelve un listado con aquellas facturas que cumplan con el criterio de búsqueda. En caso de no existir ninguna coincidencia se mostrará un mensaje al usuario.</p>	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 36.1: Realizar búsqueda filtrada de facturas

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_36.1	HISTORIA DE USUARIO: HU_36
DESCRIPCIÓN: prueba la funcionalidad de realizar criterio de búsqueda para filtrar facturas.	

<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una factura. • El usuario debe poseer los permisos de agenciamiento aduanal.
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Inicialmente se muestra un listado con las facturas. Al seleccionar la opción “Buscar” se muestra un formulario. El usuario selecciona la opción “Agregar condición”. Los posibles criterios serán:</p> <ul style="list-style-type: none"> • No. de factura • DM • VF S/DE • Descuento • Otros add • Flete • Otros gastos • Seguro
<p>RESULTADO ESPERADO:</p> <p>Se adiciona un criterio de búsqueda. Cada criterio constará de un criterio de comparación y un campo donde introducir el valor.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 36.2: Realizar búsqueda filtrada de facturas

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_36.2	HISTORIA DE USUARIO: HU_36
DESCRIPCIÓN: prueba la funcionalidad de realizar búsqueda filtrada de facturas.	
<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos una factura. • El usuario debe poseer los permisos de agenciamiento aduanal. 	

<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Inicialmente se muestra la vista donde el usuario ya ha definido los criterios de búsqueda. El usuario entonces seleccionará la opción buscar.</p>
<p>RESULTADO ESPERADO:</p> <p>Se mostrará una lista con los que cumplan las condiciones definidas en los criterios de búsqueda. De no existir coincidencias se mostrará un mensaje al usuario. Si el usuario no adicionó ningún criterio se mostrará un mensaje.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 37: Listar compradores

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_37	HISTORIA DE USUARIO: HU_37
DESCRIPCIÓN: prueba la funcionalidad de listar compradores.	
<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un comprador. • El usuario debe poseer los permisos de agenciamiento aduanal. 	
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Una vez autenticado el usuario se selecciona la opción “Compradores”.</p>	
<p>RESULTADO ESPERADO:</p> <p>Aparece un listado con todos los compradores en el sistema. En el listado se deben mostrar los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre • Código • Teléfono • Dirección <p>De no existir ningún comprador en el sistema se mostrará un mensaje.</p>	
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>	

CPA_ 38: Realizar búsqueda rápida de compradores

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_38	HISTORIA DE USUARIO: HU_38
DESCRIPCIÓN: prueba la funcionalidad de realizar una búsqueda rápida de compradores.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un comprador. • El usuario debe poseer los permisos de agenciamiento aduanal. 	
ENTRADA/PASOS DE EJECUCIÓN:	
El usuario introducirá un texto sobre el cual realizar una búsqueda rápida.	
RESULTADO ESPERADO:	
Se mostrará los resultados para la búsqueda rápida basándose en los siguientes criterios:	
<ul style="list-style-type: none"> • Nombre • Código • Teléfono • Dirección 	
Se devuelve un listado con aquellos compradores que cumplan con el criterio de búsqueda. En caso de no existir ninguna coincidencia se mostrará un mensaje al usuario.	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 39.1: Realizar búsqueda filtrada de compradores

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_39.1	HISTORIA DE USUARIO: HU_39
DESCRIPCIÓN: prueba la funcionalidad de realizar criterio de búsqueda para filtrar compradores.	

<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un comprador. • El usuario debe poseer los permisos de agenciamiento aduanal.
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Inicialmente se muestra un listado con los compradores. Al seleccionar la opción “Buscar” se muestra un formulario. El usuario selecciona la opción “Agregar condición”. Los posibles criterios serán:</p> <ul style="list-style-type: none"> • Nombre • Código • Teléfono • Dirección
<p>RESULTADO ESPERADO:</p> <p>Se adiciona un criterio de búsqueda. Cada criterio constará de un criterio de comparación y un campo donde introducir el valor.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 39.2: Realizar búsqueda filtrada de facturas

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_39.2	HISTORIA DE USUARIO: HU_39
DESCRIPCIÓN: prueba la funcionalidad de realizar búsqueda filtrada de compradores.	
<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un comprador. • El usuario debe poseer los permisos de agenciamiento aduanal. 	
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Inicialmente se muestra la vista donde el usuario ya ha definido los criterios de búsqueda. El usuario entonces seleccionará la opción buscar.</p>	

<p>RESULTADO ESPERADO:</p> <p>Se mostrará una lista con los que cumplan las condiciones definidas en los criterios de búsqueda. De no existir coincidencias se mostrará un mensaje al usuario. Si el usuario no adicionó ningún criterio se mostrará un mensaje.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 40: Listar contenedores de logística

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_40	HISTORIA DE USUARIO: HU_40
DESCRIPCIÓN: prueba la funcionalidad de listar los contenedores de logística.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor. • El usuario debe poseer los permisos de logística. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Una vez autenticado el usuario se selecciona la opción “Contenedores”.	
RESULTADO ESPERADO:	
Aparece un listado con todos los contenedores en el sistema. En el listado se deben mostrar los siguientes datos:	
<ul style="list-style-type: none"> • No. contenedor • No. manifiesto • No MBL 	
De no existir ningún contenedor en el sistema se mostrará un mensaje.	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 371: Ver contenedor de logística

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_41	HISTORIA DE USUARIO: HU_41
DESCRIPCIÓN: prueba la funcionalidad de ver un contenedor.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor. • El usuario debe poseer los permisos de logística. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Inicialmente se mostrará un listado con los contenedores. El usuario seleccionará uno de los elementos de la lista.	
RESULTADO ESPERADO:	
Se mostrará una vista con los siguientes datos del contenedor:	
<ul style="list-style-type: none"> • No. contenedor • No. manifiesto • No. MBL • Peso neto • No. bultos declarados • Propietario • Agencia • País de origen • Fecha de arribo 	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 42: Realizar búsqueda rápida de contenedores de logística

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_42	HISTORIA DE USUARIO: HU_42

<p>DESCRIPCIÓN: prueba la funcionalidad de realizar una búsqueda rápida de contenedores.</p>
<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor. • El usuario debe poseer los permisos de logística.
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>El usuario introducirá un texto sobre el cual realizar una búsqueda rápida.</p>
<p>RESULTADO ESPERADO:</p> <p>Se mostrará los resultados para la búsqueda rápida basándose en los siguientes criterios:</p> <ul style="list-style-type: none"> • No. contenedor • No. manifiesto • No. MBL <p>Se devuelve un listado con aquellos contenedores que cumplan con el criterio de búsqueda. En caso de no existir ninguna coincidencia se mostrará un mensaje al usuario.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 43.1: Realizar búsqueda filtrada de contenedores de logística

CASO DE PRUEBA DE ACEPTACIÓN	
<p>CÓDIGO: CPA_43.1</p>	<p>HISTORIA DE USUARIO: HU_43</p>
<p>DESCRIPCIÓN: prueba la funcionalidad de realizar criterio de búsqueda para filtrar contenedores.</p>	
<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor. • El usuario debe poseer los permisos de logística. 	
<p>ENTRADA/PASOS DE EJECUCIÓN:</p>	

<p>Inicialmente se muestra un listado con los contenedores. Al seleccionar la opción “Buscar” se muestra un formulario. El usuario selecciona la opción “Agregar condición”. Los posibles criterios serán:</p> <ul style="list-style-type: none"> • No. contenedor • No. manifiesto • No. MBL • Peso neto • No. bultos declarados • Propietario • Agencia • País de origen
<p>RESULTADO ESPERADO:</p> <p>Se adiciona un criterio de búsqueda. Cada criterio constará de un criterio de comparación y un campo donde introducir el valor.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 43.2: Realizar búsqueda filtrada de contenedores de logística

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_43.2	HISTORIA DE USUARIO: HU_43
DESCRIPCIÓN: prueba la funcionalidad de realizar búsqueda filtrada de contenedores.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un contenedor. • El usuario debe poseer los permisos de logística. 	
ENTRADA/PASOS DE EJECUCIÓN:	
<p>Inicialmente se muestra la vista donde el usuario ya ha definido los criterios de búsqueda. El usuario entonces seleccionará la opción buscar.</p>	
RESULTADO ESPERADO:	

Se mostrará una lista con los que cumplan las condiciones definidas en los criterios de búsqueda. De no existir coincidencias se mostrará un mensaje al usuario. Si el usuario no adicionó ningún criterio se mostrará un mensaje.

EVALUACIÓN DE LA PRUEBA: SATISFACTORIA

CPA_ 44: Listar reportes de incidencias

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_44	HISTORIA DE USUARIO: HU_44
DESCRIPCIÓN: prueba la funcionalidad de listar los reportes de incidencias.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un reporte de incidencias. • El usuario debe poseer los permisos de logística. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Una vez autenticado el usuario se selecciona la opción “Reporte de incidencias”.	
RESULTADO ESPERADO:	
Aparece un listado con todos los reportes de incidencias en el sistema. En el listado se deben mostrar los siguientes datos:	
<ul style="list-style-type: none"> • Fecha y hora de comienzo • Fecha y hora de terminación • Lugar de despacho • Cantidad de bultos 	
De no existir ningún reporte de incidencias en el sistema se mostrará un mensaje.	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 385: Ver reporte de incidencias

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_45	HISTORIA DE USUARIO: HU_45
DESCRIPCIÓN: prueba la funcionalidad de ver un reporte de incidencias.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un reporte de incidencias. • El usuario debe poseer los permisos de logística. 	
ENTRADA/PASOS DE EJECUCIÓN:	
Inicialmente se mostrará un listado con los reportes de incidencia. El usuario seleccionará uno de los elementos de la lista.	
RESULTADO ESPERADO:	
Se mostrará una vista con los siguientes datos del reporte de incidencias:	
<ul style="list-style-type: none"> • Fecha y hora de comienzo • Fecha y hora de terminación • Lugar de despacho • Almacén • Cantidad de bultos • Cantidad HBL • Cantidad de bultos en el almacén • Cantidad de bultos faltantes • Cantidad de bultos sobrantes • Cantidad de bultos no manifestados • Cantidad de bultos averiados 	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 46: Realizar búsqueda rápida de reportes de incidencias

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_46	HISTORIA DE USUARIO: HU_46
DESCRIPCIÓN: prueba la funcionalidad de realizar una búsqueda rápida de reportes de incidencias.	
CONDICIONES DE EJECUCIÓN:	
<ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un reporte de incidencias. • El usuario debe poseer los permisos de logística. 	
ENTRADA/PASOS DE EJECUCIÓN:	
El usuario introducirá un texto sobre el cual realizar una búsqueda rápida.	
RESULTADO ESPERADO:	
Se mostrará los resultados para la búsqueda rápida basándose en los siguientes criterios:	
<ul style="list-style-type: none"> • Fecha y hora de comienzo • Fecha y hora de terminación • Lugar de despacho • Cantidad de bultos 	
Se devuelve un listado con aquellos reportes de incidencias que cumplan con el criterio de búsqueda. En caso de no existir ninguna coincidencia se mostrará un mensaje al usuario.	
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA	

CPA_ 47.1: Realizar búsqueda filtrada de reportes de incidencias

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_47.1	HISTORIA DE USUARIO: HU_47
DESCRIPCIÓN: prueba la funcionalidad de realizar criterio de búsqueda para filtrar reportes de incidencias.	

<p>CONDICIONES DE EJECUCIÓN:</p> <ul style="list-style-type: none"> • Estar autenticado en el sistema. • Debe existir en el sistema al menos un reporte de incidencias. • El usuario debe poseer los permisos de logística.
<p>ENTRADA/PASOS DE EJECUCIÓN:</p> <p>Inicialmente se muestra un listado con los reportes de incidencias. Al seleccionar la opción “Buscar” se muestra un formulario. El usuario selecciona la opción “Agregar condición”. Los posibles criterios serán:</p> <ul style="list-style-type: none"> • Fecha y hora de comienzo • Fecha y hora de terminación • Lugar de despacho • Almacén • Cantidad de bultos • Cantidad HBL • Cantidad de bultos en el almacén • Cantidad de bultos faltantes • Cantidad de bultos sobrantes • Cantidad de bultos no manifestados • Cantidad de bultos averiados
<p>RESULTADO ESPERADO:</p> <p>Se adiciona un criterio de búsqueda. Cada criterio constará de un criterio de comparación y un campo donde introducir el valor.</p>
<p>EVALUACIÓN DE LA PRUEBA: SATISFACTORIA</p>

CPA_ 47.2: Realizar búsqueda filtrada de reportes de incidencias

CASO DE PRUEBA DE ACEPTACIÓN	
CÓDIGO: CPA_47.2	HISTORIA DE USUARIO: HU_47
DESCRIPCIÓN: prueba la funcionalidad de realizar búsqueda filtrada de reportes de incidencias.	
CONDICIONES DE EJECUCIÓN:	

- Estar autenticado en el sistema.
- Debe existir en el sistema al menos un reporte de incidencias.
- El usuario debe poseer los permisos de logística.

ENTRADA/PASOS DE EJECUCIÓN:

Inicialmente se muestra la vista donde el usuario ya ha definido los criterios de búsqueda. El usuario entonces seleccionará la opción buscar.

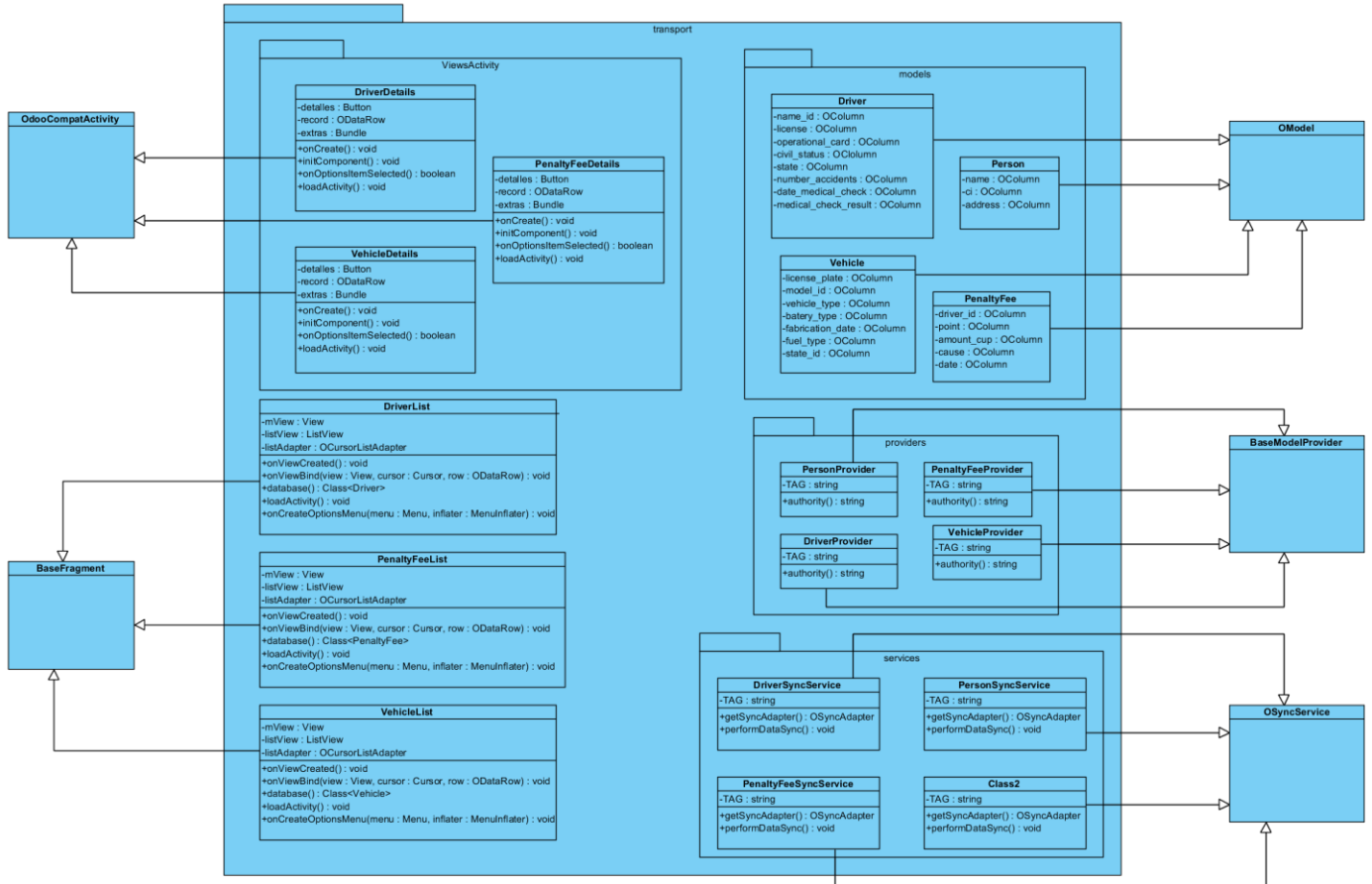
RESULTADO ESPERADO:

Se mostrará una lista con los que cumplan las condiciones definidas en los criterios de búsqueda. De no existir coincidencias se mostrará un mensaje al usuario. Si el usuario no adicionó ningún criterio se mostrará un mensaje.

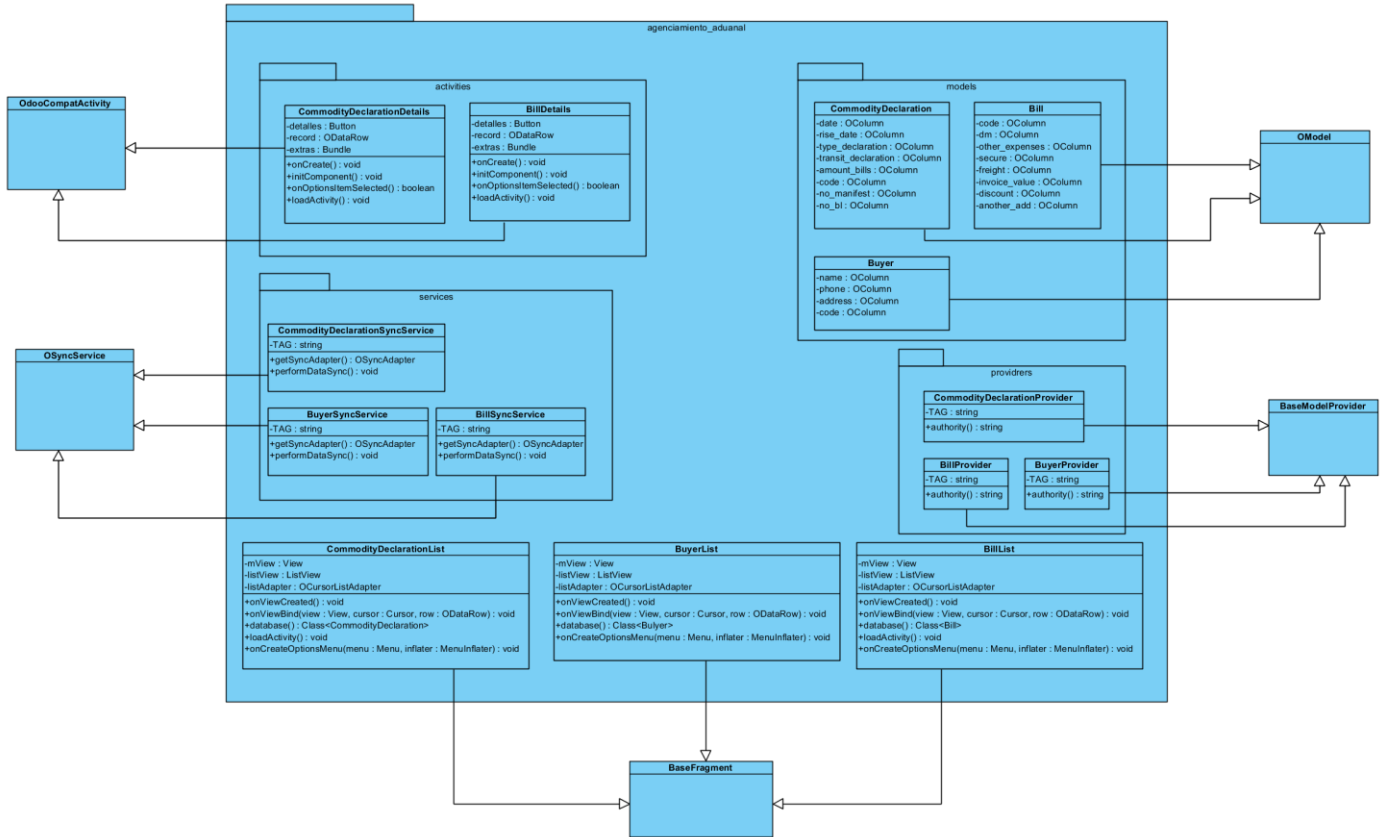
EVALUACIÓN DE LA PRUEBA: SATISFACTORIA

Anexo 3: Diagramas de clases

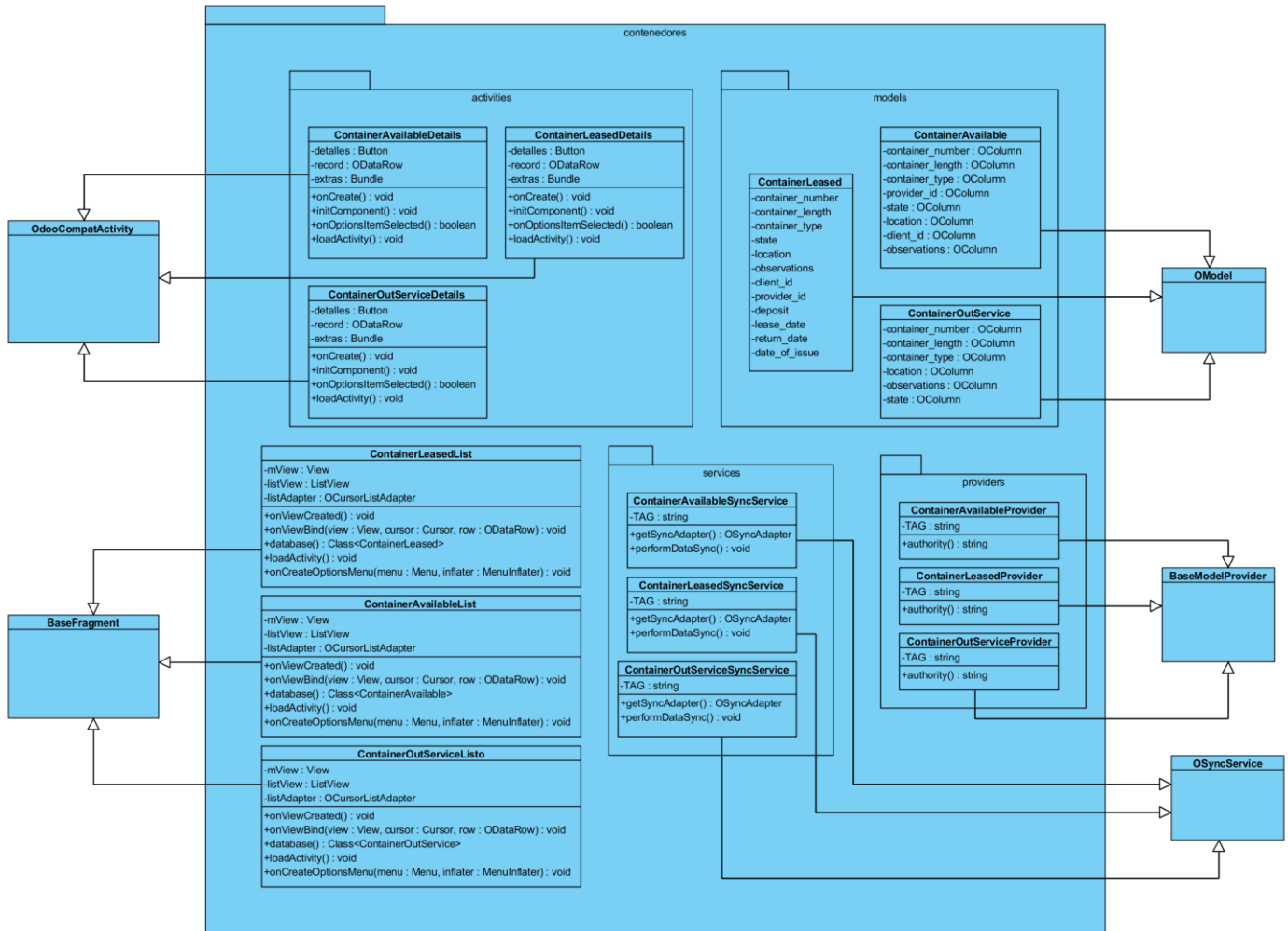
DC_1: Módulo de transporte



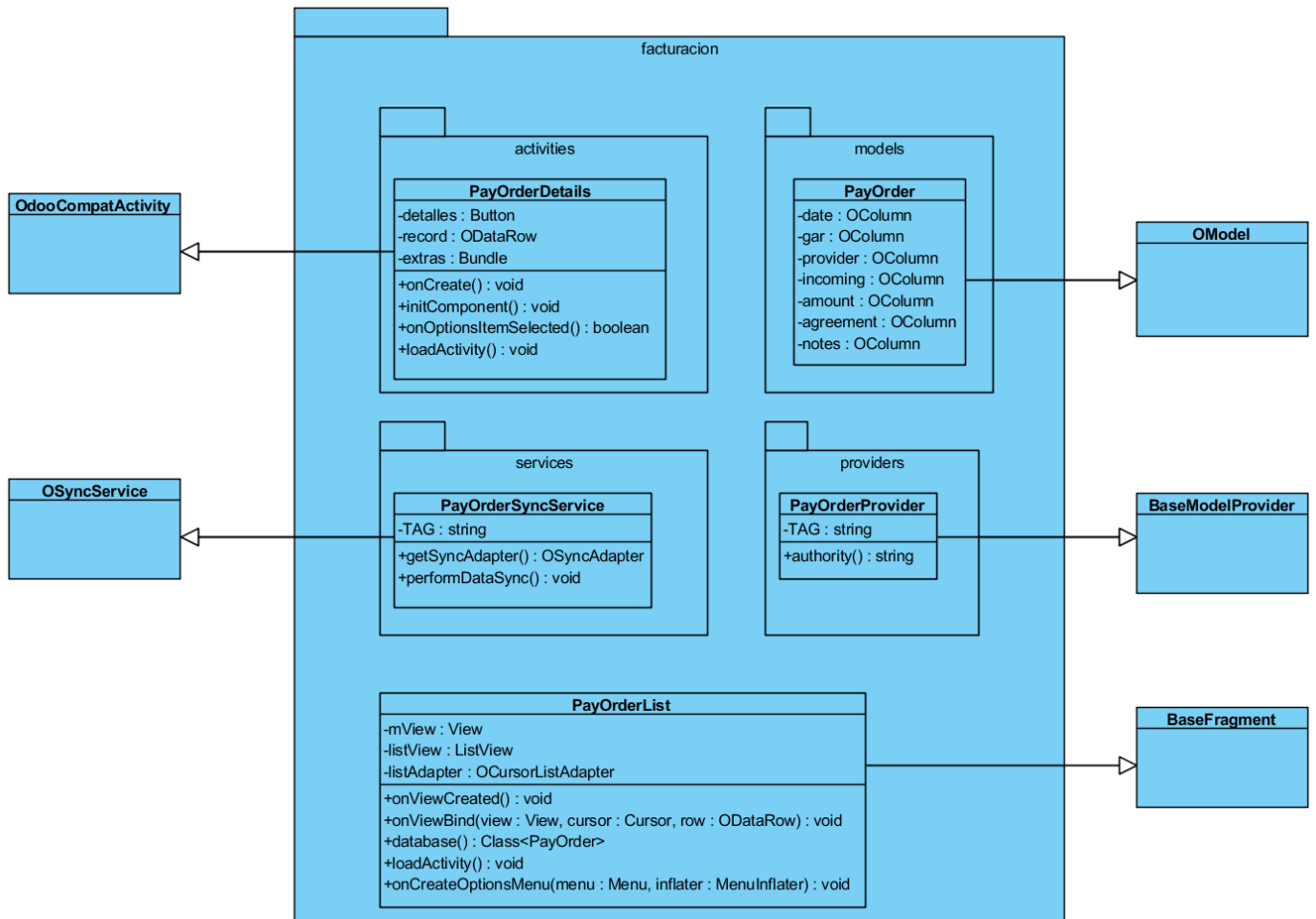
DC_2: Módulo de agenciamiento aduanal



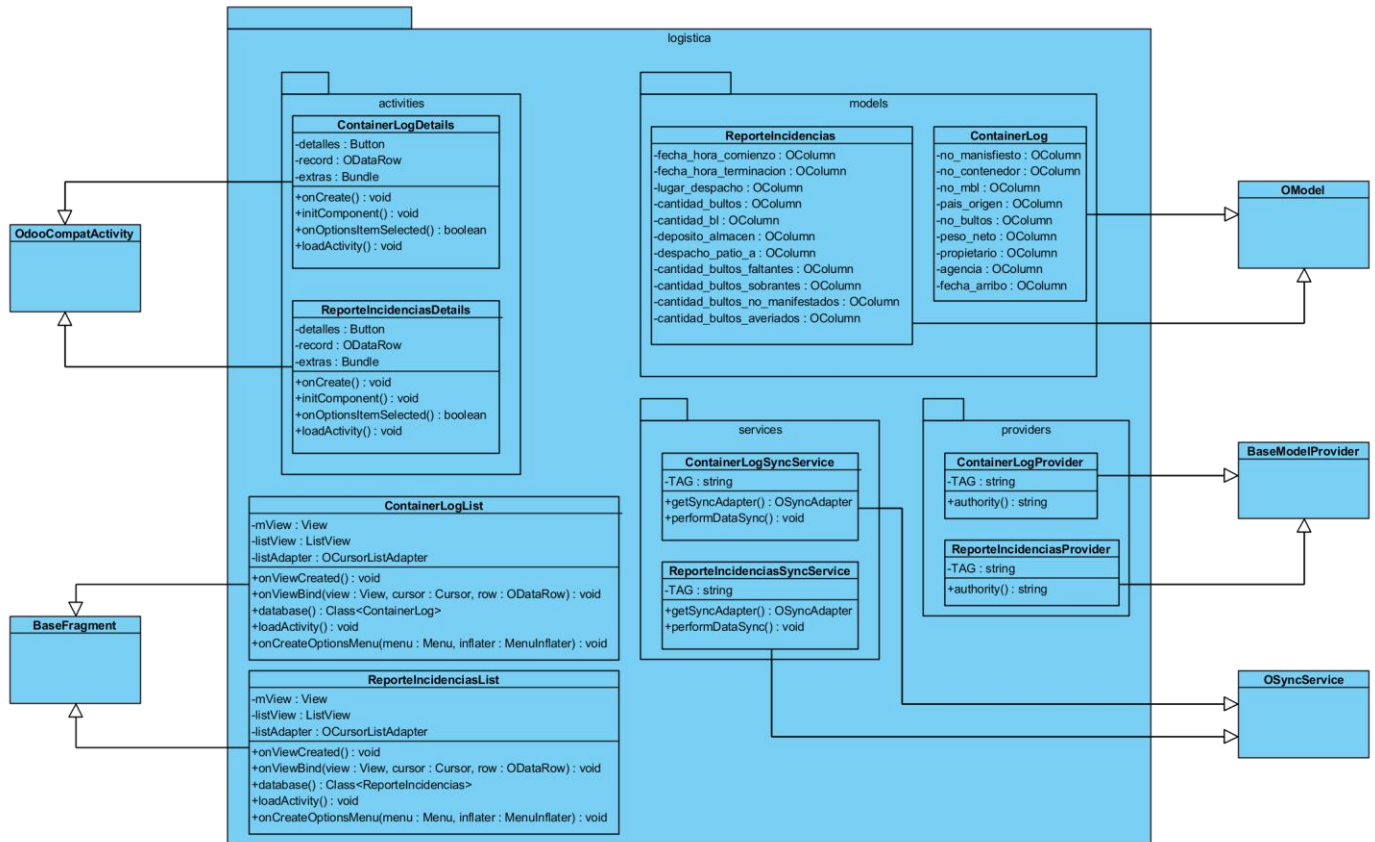
DC_3: Módulo de arrendamiento de contenedores



DC_ 4: Módulo de Facturación



DC_ 5: Módulo de logística



DC_ 6: Módulo de Orden de pago

