



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA

UNIVERSIDAD POLITÉCNICA DE VALENCIA

ESCUELA TÉCNICA SUPERIOR DE  
INGENIERÍA INFORMÁTICA



# ScrAPPed cars: INTEGRACIÓN DE TECNOLOGÍAS MÓVILES EN EL NEGOCIO DE LOS DESGUACES DE VEHÍCULOS.

Proyecto final de carrera de Ingeniería Informática

Estudiante: Miguel Ángel Pérez Girbés

Director: David de Andrés Martínez

Co-director: Sergio Jiménez Sebastián



Valencia, septiembre de 2013



## Agradecimientos:

No necesito un momento especial para ser agradecido,  
con aquellos que desde que nací, me han querido.  
Papá y mamá, que me habéis educado desde el cariño,  
agradezco el esfuerzo y dedicación que de vosotros he recibido,  
y os debo el mérito de lo que en mi vida haya conseguido.

El tiempo ha pasado, y mucho hemos vivido,  
pero sólo estudiando, no habría subsistido,  
el apoyo de mi hermano siempre lo he tenido,  
su nobleza, simpatía y bondad me han abstraído,  
pero con su humor, ¡cuánto me he divertido!

Desde que empecé la carrera, mucho me ha ocurrido,  
entre otras cosas, el amor de mi vida he conocido,  
por eso unas líneas para ella no me caen en el olvido,  
porque sin su apoyo tampoco lo hubiera conseguido,

Y también doy las gracias a mis amigos,  
con los que desde pequeño he crecido,  
y aquellos que poco a poco han aparecido,  
porque cuando los he necesitado, los he tenido.  
y la lealtad es algo, que les tengo reconocido.

## Índice

Lista de ilustraciones .....	5
Glosario .....	7
1 Introducción: .....	12
2 Estudio del mercado y operativa del sector.....	15
2.1 La Normativa Europea y su aplicación en España.....	15
2.1.1 Legislación Europea.....	15
2.1.2 Aplicación de las directivas europeas en España.....	16
2.1.3 Centros colaboradores en España.....	20
2.1.4 Las obligaciones del propietario de un vehículo en el momento de su baja.....	22
2.2 Los clientes.....	24
2.2.1 El cliente objetivo de nuestra plataforma software.....	24
2.2.2 Funciones de los Centros Autorizados de Tratamiento de Vehículos Fuera de Uso..	28
2.3 Alternativas en el mercado.....	31
2.3.1 Proveedores de software de gestión de CATs.....	31
2.3.2 Programas similares .....	37
2.3.3 Ventajas para los desguaces.....	40
2.4 Establecimiento de los requisitos en base a prioridades MoSCoW.....	41
3. Monetización.....	44
4. Especificación .....	45
4.1 Arquitectura global de la APP .....	45
4.1.1 El entorno .....	45
4.1.2 Arquitectura de la plataforma.....	53
4.2 Lógica de negocio.....	54
4.3 GUI (Interfaz Gráfica de Usuario) .....	69
CRV Almacén .....	69
CRV Campa .....	76
4.4 Modelo de datos.....	82
5 Implementación y desarrollo .....	84
5.1 Herramientas utilizadas.....	85
5.1.2 El entorno de desarrollo.....	85
5.1.3 Documentación y herramientas internas y de terceros.....	93
5.2 Plataforma objetivo donde se explotará.....	95

5.3 Detalle.....	99
5.3.1 Metodología de desarrollo.....	99
5.3.2 Estructura del proyecto.....	101
5.3.3 Ejemplos de recursos del proyecto.....	104
5.3.4 Problemas surgidos durante la implementación.....	115
6 Despliegue, explotación y evaluación.....	119
7 Conclusión.....	121
8 Referencias.....	123

## Tabla de ilustraciones

<i>Ilustración 1: Porcentajes de recuperación del peso total de los vehículos desechados en los países de la UE. Fuente: SIGRAUTO.</i>	17
<i>Ilustración 2: Evolución de los niveles de recuperación de VFU en España. Fuente: SIGRAUTO</i>	18
<i>Ilustración 3: Evolución de la cantidad de CATs acreditados en España. Fuente: SIGRAUTO</i>	18
<i>Ilustración 4: Evolución del número de Fragmentadoras concertadas en España. Fuente: SIGRAUTO</i>	19
<i>Ilustración 5: Reparto geográfico en España de CATs, fragmentadoras y plantas de postfragmentación. Fuente: SIGRAUTO</i>	19
<i>Ilustración 6: Reparto por provincias del número de bajas de vehículos en España. Fuente: SIGRAUTO</i>	20
<i>Ilustración 7: Ubicación de las empresas asociadas a ANFAC. Fuente: SIGRAUTO</i>	21
<i>Ilustración 8: Reparto geográfico de las empresas fragmentadoras autorizadas en España. Fuente: SIGRAUTO</i>	22
<i>Ilustración 9: Localización de los CATs en España. Fuente: SIGRAUTO</i>	23
<i>Ilustración 10: Centros Autorizados de Tratamiento de VFUs en la provincia de Valencia (1)</i>	26
<i>Ilustración 11: Centros Autorizados de Tratamiento de VFUs en la provincia de Valencia (2)</i>	27
<i>Ilustración 12: Fragmentadoras acreditadas en la provincia de Valencia</i>	27
<i>Ilustración 13: Aplicación para la tramitación de bajas e informe de baja tramitada. Fuente: DGT</i>	28
<i>Ilustración 14: Zona de Descontaminación de VFUs en un CAT</i>	29
<i>Ilustración 15: Almacenes de piezas y componentes recuperados listos para reutilizar.</i>	30
<i>Ilustración 16: Compactación de VFUs y almacenamiento en una Fragmentadora.</i>	30
<i>Ilustración 17: Gestor CAT-VFU 2008 - Módulo de Gestión de vehículos</i>	32
<i>Ilustración 18: Gestor CAT-VFU 2008 - Módulo de Gestión de Artículos</i>	33
<i>Ilustración 19: Captura de pantalla de la aplicación on-line de Pinnacle Actual Systems</i>	35
<i>Ilustración 20: Imagen de GVisualPDA de Gira</i>	37
<i>Ilustración 21: Captura de pantalla de la herramienta móvil de Pinnacle Actual Systems</i>	38
<i>Ilustración 22: Esquema de la arquitectura de SIDE0</i>	45
<i>Ilustración 23: Captura de pantalla de la aplicación de escritorio CRVNet</i>	47
<i>Ilustración 24: Orden de despiece previa a la implantación de nuestro software</i>	48
<i>Ilustración 25: Capturas de pantalla del menú de selección de piezas para la generación de órdenes</i>	49
<i>Ilustración 26: Diagrama del esquema de los almacenes virtuales</i>	52
<i>Ilustración 27: Situación de la APP dentro de la arquitectura de la plataforma CRVNet</i>	53
<i>Ilustración 28: Diagrama de casos de uso</i>	54
<i>Ilustración 29: Captura de pantalla de la APP CRV Almacén - Listado de piezas</i>	69
<i>Ilustración 30: Captura de pantalla de la APP CRV Almacén - Descargar piezas desde orden</i>	70
<i>Ilustración 31: Captura de pantalla de la APP CRV Almacén - Selección</i>	70
<i>Ilustración 32: Captura de pantalla de la APP CRV Almacén - Pulsación prolongada</i>	70
<i>Ilustración 33: Captura de pantalla de la APP CRV Almacén - Menú Información de la pieza</i>	71
<i>Ilustración 34: Captura de pantalla de la APP CRV Almacén - Copiar/Pegar</i>	71
<i>Ilustración 35: Captura de pantalla de la APP CRV Almacén – Capturas de la galería de Imágenes</i>	72
<i>Ilustración 36: Diagramas de interacción de la interfaz de CRV Almacén</i>	74
<i>Ilustración 37: Captura de pantalla de la APP CRV Campa - Pantalla principal</i>	76
<i>Ilustración 38: Captura de pantalla de la APP CRV Campa - Descarga de vehículos por fecha</i>	76
<i>Ilustración 39: Captura de pantalla de la APP CRV Campa - Pantalla principal (Horizontal)</i>	76
<i>Ilustración 40: Captura de pantalla de la APP CRV Campa - Vista resumen (Operaciones finalizadas)</i>	77
<i>Ilustración 41: Captura de pantalla de la APP CRV Campa – Vista resumen (previo)</i>	77
<i>Ilustración 42: Captura de pantalla de la APP CRV Campa - Vista de identificación</i>	77
<i>Ilustración 43: Captura de pantalla de la APP CRV Campa - Vista de identificación (scrolled)</i>	77

<i>Ilustración 44: Captura de pantalla de la APP CRV Campa - Captura de imágenes</i>	78
<i>Ilustración 45: Captura de pantalla de la APP CRV Campa - Validación de piezas</i>	78
<i>Ilustración 46: Diagramas de interacción de la interfaz de CRV Campa</i>	80
<i>Ilustración 47: Diagrama de entidades y relaciones</i>	82
<i>Ilustración 48: Gráfico comparativo del reparto de mercado de sistemas operativos móviles. Fuente: IDC</i>	84
<i>Ilustración 49: Captura de pantalla de Eclipse</i>	86
<i>Ilustración 50: Viñeta del conjunto de aplicaciones de Xamarin</i>	87
<i>Ilustración 51: Captura de pantalla de Xamarin Studio</i>	88
<i>Ilustración 52: Captura de Android Studio - Previsualización de recursos</i>	89
<i>Ilustración 53: Captura de Android Studio – Generación de recursos de idioma</i>	90
<i>Ilustración 54: Captura de Android Studio - Previsualización de Layouts</i>	91
<i>Ilustración 55: Gráfica de porciones reparto de versiones Android</i>	95
<i>Ilustración 56: Tabla de reparto de versiones de Android en septiembre de 2013</i>	96
<i>Ilustración 57: Cuotas de mercado según versión de Android</i>	96
<i>Ilustración 58: Captura de la especificación de opciones del proyecto en Xamarin Studio</i>	98
<i>Ilustración 59: Flujo de trabajos RAD</i>	99
<i>Ilustración 60: Comparación de RAD frente a las metodologías tradicionales. Fuente: rootservices.com</i>	99
<i>Ilustración 61: Explorador de archivos de Xamarin - Visión general</i>	101
<i>Ilustración 62: Explorador de archivos de Xamarin - Recursos del proyecto</i>	102
<i>Ilustración 63: Explorador de archivos de Xamarin - Assets</i>	103
<i>Ilustración 64: Explorador de archivos de Xamarin - Adaptadores</i>	103
<i>Ilustración 65: Explorador de archivos de Xamarin -Ficheros fuente</i>	103
<i>Ilustración 66: Explorador de archivos de Xamarin - Definición de una vista</i>	104
<i>Ilustración 67: Capturas de un prototipo descartado de aplicación de CRV Campa</i>	115
<i>Ilustración 68: Captura de una notificación de API no documentada en Xamarin</i>	116
<i>Ilustración 69: Plan de precios y características incluidas de Xamarin Studio</i>	117

## Glosario

TÉRMINO	DEFINICIÓN
<b>AEDRA:</b>	Asociación Española de Desguazadores y Reciclaje del Automóvil.
<b>ANFAC:</b>	Asociación Española Nacional de Fabricantes de Automóviles y Camiones.
<b>ANIACAM:</b>	Asociación Nacional de Importadores de Automóviles Camiones Autobuses y Motocicletas.
<b>Animaciones Tween:</b>	Animaciones basadas en la interpolación de efectos básicos como rotaciones, translaciones, escalados y efectos de transparencias.
<b>Ant:</b>	Herramienta de construcción de proyectos desarrollada por Apache Software Foundation.
<b>API:</b>	Interfaz de Programación de Aplicaciones. Es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
<b>APK:</b>	Application Package File. Colección de recursos instalable que contiene la definición de una aplicación en Android.
<b>APP:</b>	Aplicación de software para móvil.
<b>AWT:</b>	Abstract Windows Toolkit. Es un Kit de herramientas de gráficos, interfaz de usuario, y sistema de ventanas independiente de la plataforma original de Java.
<b>AZELER CAT:</b>	Sistema de Gestión Integral de Vehículos para Centros Autorizados de Tratamiento.
<b>BlackBerry:</b>	Es una línea de teléfonos inteligentes desarrollada por la compañía canadiense BlackBerry (antes Research In Motion o RIM).
<b>B2B:</b>	Business-to-Business. Es un concepto de negocios en el que sólo intervienen empresas que prestan servicios a otras empresas.
<b>C#:</b>	Lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET
<b>CAMPA:</b>	Término utilizado por los profesionales de los desguaces para referirse al lugar donde almacenan los vehículos antes de ser despiezados.
<b>CASE:</b>	(Computer Aided Software Engineering), son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.
<b>CAT:</b>	Centro Autorizado de Tratamiento de vehículos fuera de uso
<b>Codificación ISO 8859 Latin 1:</b>	Es una norma de la ISO que define la codificación del alfabeto latino, incluyendo los diacríticos (como letras acentuadas, ñ, ç), y letras especiales.
<b>Codificación UTF8:</b>	Es un formato de codificación de caracteres Unicode e ISO 10646 utilizando símbolos de longitud variable.
<b>Código Data Matrix:</b>	Es un sistema industrial de codificación bidimensional que permite la generación de un gran volumen de información en un formato muy reducido, con una alta fiabilidad de lectura gracias a sus sistemas de información redundante y corrección de errores.
<b>Código QR:</b>	Es un módulo útil para almacenar información en una matriz de puntos o un código de barras bidimensional.



<b>Código OPC:</b>	Es un estándar de comunicación en el campo del control y supervisión de procesos industriales, basado en una tecnología Microsoft.
<b>CRVnet:</b>	Plataforma software de Gestión de Desguaces.
<b>CVS:</b>	(Concurrent Versions System) Es una aplicación informática que implementa un sistema de control de versiones: mantiene el registro de todo el trabajo y los cambios en los ficheros que forman un proyecto y permite que distintos desarrolladores colaboren.
<b>DBMS:</b>	Sistemas de Gestión de Bases de Datos.
<b>DGT:</b>	Dirección General de Tráfico.
<b>DSL:</b>	Lenguaje Específico de Dominio.
<b>Envelope:</b>	Envoltorio de XML que permite especificar la versión del protocolo y asegurar sus restricciones. Funciona como elemento raíz en un mensaje SOAP.
<b>ÉTER:</b>	Programa de tramitación telemática de documentos en el ámbito de la gestión de residuos.
<b>FEEDBACK:</b>	Retroalimentación. Capacidad de un emisor para recoger reacciones de los receptores y modificar su mensaje, de acuerdo con la información recibida.
<b>FER:</b>	Federación Española de la Recuperación.
<b>GCM:</b>	Google Cloud Messaging . Servicio de mensajería para comunicar aplicaciones cliente con servidor.
<b>Gestionep Europeas:</b>	Programa diseñado específicamente para desguaces de vehículos y empresas de repuestos de todo tipo.
<b>Gestor CAT-VFU:</b>	Programa diseñado específicamente para Gestión Integral de Centros Autorizados de Tratamientos y Vehículos Fuera de Uso.
<b>Groovy:</b>	Es un lenguaje dinámico, compacto, con DSL, soporte para Scripts, orientado a objetos y totalmente integrado con Java, lo cual permite el uso de código de este lenguaje en cualquier momento facilitando así la labor a los desarrolladores.
<b>GIRA:</b>	Guía Informática Recambios del Automóvil. Es una empresa que desarrolla un producto informático adaptado a la tienda de recambios.
<b>Google Code:</b>	Es un sitio de Google para desarrolladores interesados en el desarrollo Google-related/open-source. El sitio contiene códigos fuente libres, una lista de sus servicios de apoyo público y API.
<b>Grails:</b>	Es un framework basado en Groovy para desarrollo de webs. Este framework está basado en el patrón MVC (Modelo Vista Controlador) lo cual facilita bastante la realización de un proyecto web.
<b>GTK#:</b>	Es un conjunto de bibliotecas multiplataforma para desarrollar interfaces gráficas de usuario (GUI), principalmente para los entornos gráficos GNOME, XFCE Y ROX aunque también se puede usar en el escritorio de Windows, Mac Os y otros.
<b>GUI:</b>	Interfaz Gráfica de Usuario. Es un programa informático que actúa proporcionando un entorno visual sencillo para permitir la comunicación con el sistema operativo del ordenador.
<b>Hibernate:</b>	Es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación.
<b>IBM:</b>	Empresa que fabrica y comercializa hardware, software y servicios relacionados con la informática. (International Business Machines).

<b>IDC:</b>	(International Data Corporation). Es una empresa estadounidense dedicada a la investigación de mercado, análisis y consultoría especializada en tecnologías de la información, telecomunicaciones y tecnología de consumo.
<b>IDE:</b>	Entorno de Desarrollado Integrado. Es un programa informático compuesto por un conjunto de herramientas de programación.
<b>IOS:</b>	Es un sistema operativo desarrollado por Apple para su teléfono inteligente iPhone.
<b>JAVA:</b>	Es un lenguaje de programación interpretado y orientado a objetos.
<b>JSON:</b>	JavaScript Object Notation es un formato ligero para el intercambio de datos.
<b>Junit:</b>	Es un conjunto de bibliotecas que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.
<b>LaTeX:</b>	Lenguaje para procesado de texto.
<b>Licencia Indie:</b>	Licencia para un solo desarrollador.
<b>LinQ:</b>	Es un componente de la plataforma Microsoft .NET que agrega capacidades de consulta a datos de manera nativa a los lenguajes .NET
<b>Led:</b>	Componente electrónico que emite luz.
<b>LOPD:</b>	Ley Orgánica de Protección de Datos.
<b>Maven:</b>	Es una herramienta de software para la gestión y construcción de proyectos Java.
<b>.NET Framework:</b>	El Microsoft .NET Framework, es un componente de software que puede ser o es incluido en los sistemas operativos Microsoft Windows, que permite un rápido desarrollo de aplicaciones.
<b>OCR:</b>	Reconocimiento Óptico de Caracteres.
<b>Open Source:</b>	Es la expresión con la que se conoce al software distribuido y desarrollado libremente.
<b>PDF:</b>	Es un formato de almacenamiento de documentos digitales independiente de plataformas de software o hardware.
<b>PCB:</b>	Bifenilos Policlorados. Son una serie de compuestos organoclorados, que constituyen una familia de 209 congéneres, los cuales se forman mediante la cloración de diferentes posiciones del bifenilo.
<b>PCT:</b>	Son compuestos químicos del bromo llamados Polibromobifenilos o Bifenilos polibromados. Se usan en la industria electrónica como retardadores de inflamabilidad en plásticos.
<b>Plug-in</b>	Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.
<b>Proyecto Mono:</b>	Mono es el nombre de un proyecto de código abierto para crear un grupo de herramientas libres, basadas en GNU/LINUX y compatibles con .NET
<b>RAD:</b>	Desarrollo Rápido de Aplicaciones.
<b>REST:</b>	Representational State Transfer. Es una técnica de arquitectura software para sistemas hipertexto distribuidos como la World Wide Web.
<b>SDK de Android:</b>	Es el kit de herramientas con el que se fabrican todas las aplicaciones Android, incluso el propio sistema operativo.
<b>SharedPreferences:</b>	Herramienta disponible en Android que permite almacenar en una instancia de la aplicación configuraciones basadas en tipos simples.

<b>SIDEO:</b>	Es una plataforma que proporciona a los clientes canales de negocio con los que sacar más partido y difusión de sus almacenes de pieza, a la vez que maximiza la posibilidad de llegar a un público más amplio.
<b>SIGRAUTO:</b>	Asociación Española para el Tratamiento Medioambiental de los Vehículos Fuera de Uso.
<b>Scroll:</b>	Movimiento en 2D de los contenidos que conforman el escenario de una ventana que se muestra en una aplicación informática.
<b>SOAP:</b>	Protocolo estándar (Simple Object Access Protocol).
<b>Spring:</b>	Es un framework ligero y no intrusivo, de código abierto para Java, basado en la inyección de dependencias y la programación orientada a aspectos.
<b>Suite:</b>	Es un conjunto de aplicaciones y herramientas de software incluidas en un sólo paquete y que, por lo general, comparten un aspecto similar y se integran entre sí.
<b>Swing:</b>	Es una biblioteca gráfica para Java.
<b>Telnet:</b>	TELEcommunication NETwork. Es el nombre de un protocolo de red que sirve para acceder mediante una red a otra máquina para manejarla remotamente .
<b>UML:</b>	Lenguaje Unificado de Modelado.
<b>URI:</b>	Identificador Uniforme de Recursos. Sirve para identificar recursos en internet.
<b>URL:</b>	Uniform Resource Locator. Es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación.
<b>VFU:</b>	Vehículos Fuera de Uso.
<b>VisualAge:</b>	Son entornos de desarrollo integrados de IBM, que incluyen soporte para múltiples lenguajes de programación.
<b>Web:</b>	Red de tamaño mundial (World Wide Web).
<b>Web services:</b>	Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
<b>Widgets:</b>	Son pequeños programas a modo de herramientas que permiten dar fácil acceso a funciones frecuentemente usadas y proveer de información visual al usuario.
<b>WI-FI:</b>	Es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica.
<b>Windows Phone:</b>	Es un sistema operativo móvil desarrollado por Microsoft, como sucesor de la plataforma Windows Mobile.
<b>WSDL:</b>	Web Services Description Language. Es un formato XML que se utiliza para describir servicios Web.
<b>Xamarin</b>	Es una implementación libre de la plataforma de desarrollo.NET para dispositivos Android, iOS y GNU/Linux.
<b>XML:</b>	Documento de texto plano (sin formato). Lenguaje de Marcas Extensible.
<b>Zxing:</b>	Librería de código abierto de lectura de códigos multi-formato 1D/2D.



## 1 Introducción:

El panorama económico actual español, marcado por el paro y la pérdida de capacidad adquisitiva del ciudadano, está descubriendo nuevos nichos de negocio y reinventando las costumbres que en época de plenitud quedaban rezagadas a las actitudes más austeras.

Paralelamente, nos encontramos en un contexto en el que las nuevas tecnologías avanzan de manera desmesurada y en que los dispositivos móviles forman parte de nuestra vida en mayor medida de lo que jamás hubiésemos esperado; nos acompañan allá donde vamos, nos comunican con nuestro seres queridos, nos proporcionan entretenimiento, y en mayor o menor medida, se convierten en una herramienta que consume gran parte de nuestra atención del día a día. Tal es el nivel de dependencia que nos han creado, que han cambiado la manera que tenemos de comunicarnos y hasta de trabajar. Todo ello gracias a la gran cantidad de posibilidades tecnológicas que ofrecen; sensores, micrófonos, cámaras, antenas, memorias... todo ello integrado en un único dispositivo, capaz de hacer cualquier labor más fácil y cómoda.

Bien conocedora de estos hechos fehacientes, la industria del software ha hecho uso de estas tecnologías (todavía en continua expansión), orientando sus productos a proporcionar servicios dirigidos a alcanzar una gran cuota de mercado. Muchos de éstos, han sustituido aplicaciones de éxito ya existentes sobre plataformas Web, o sobre software de escritorio para migrar a plataformas móviles idóneas para acompañar al usuario en su quehacer diario y proporcionarle en cualquier momento y lugar, aquellas herramientas que anteriormente únicamente se podía acceder a través de un ordenador. Para ello, dicho software, en aras de acercarse al mayor número de segmentos del mercado, se ha construido sobre las plataformas líderes en el mercado Android, IOS, Windows Phone y RIM.

La estrategia comercial que ha dirigido la industria del software para dispositivos móviles en época de bienes, ha llevado a apostar por el tipo de aplicación enfocada al público general, es decir, una aplicación que ha satisfecho las necesidades de un gran número de usuarios. Esta maniobra ha inducido el desarrollo de aplicaciones a la creación de infinidad de variantes de las mismas soluciones de un reducido número de problemas, aquellas aplicaciones que resuelven los problemas más frecuentes. En consecuencia de lo anterior, han quedado de lado otras aplicaciones enfocadas a sectores que requieren de un alto conocimiento del negocio, y aquellas que resuelven problemas más específicos, incapaces de ser reutilizadas para otros menesteres, teniendo un bajo retorno de inversión, pero no por ello prescindibles en los sectores afectados.

Uno de los sectores que durante los pasados años ha estado lejos del punto de mira de los fabricantes de software para plataformas móviles, es el perteneciente a los desguaces o centros de reaprovechamiento de piezas de vehículos (que han salido fortalecidos por la gran disminución de ventas de vehículos nuevos y el consecuente envejecimiento del parque automovilístico español), siendo a día de hoy, una fuente de recambios nada despreciable para talleres y particulares, donde pueden encontrar una alternativa económica a la hora de sustituir las piezas defectuosas de sus vehículos.

Una vez más, la utilización de las tecnologías de la información, impulsarán la gestión, localización y tratamiento de almacenes facilitando la disposición del producto al cliente final. Y es aquí donde se centrará este proyecto, la propuesta de este PFC resuelve un caso real de una aplicación corporativa para la gestión, catalogación y comprobación de las entradas de piezas procedentes de los vehículos disponibles en los desguaces de nuestros clientes. Dicho desarrollo se realizará sobre la plataforma móvil Android (dado que es la que cuenta con la mayor tasa de crecimiento en la actualidad), con vistas a ser portado a otros escenarios en un futuro.

La mencionada aplicación se integrará con el actual software de escritorio ya implantado en los clientes y gestionará recursos comunes en una plataforma Web a través de la arquitectura SOAP para la actualización e introducción de nueva información.

La propuesta realizada sustituirá la metodología de trabajo actual basada en la generación de notas de papel impresas para la posterior digitalización de los resultados, por la utilización de una aplicación que extraerá los partes de trabajo directamente de “Web services”, presentando éstos en forma de formularios intuitivos y guiados motivando la estandarización del protocolo de recogida de datos del material extraído en los desguaces, facilitando el trabajo de los mecánicos y minimizando el acometimiento de errores, lo que simplificará y agilizará el trabajo de toda la cadena de aprovisionamiento de información en la entrada de vehículos y proporcionará una ventaja competitiva a nuestra plataforma respecto a otras alternativas del mercado.

El citado desarrollo deberá poder ser ejecutado en dispositivos “Low Cost” y se adaptará a las distintas resoluciones de pantalla estándares de mercado (tanto Tablet como Smartphone). Siendo como requisitos necesarios para la correcta explotación de la aplicación: la capacidad de conexión Wifi y la disposición de cámara de fotos.

La aplicación terminada será entregada como herramienta de trabajo a los distintos clientes a coste cero, aumentando el valor conjunto del software proporcionado por nuestra empresa a sus clientes, y al mismo tiempo fomentando una mayor calidad en la información recibida desde los desguaces.

Previo a la implementación, se realizará un estudio que enfrentará dos plataformas de desarrollo nativo para Android: Eclipse mediante desarrollo en Java y Xamarin de Mono, mediante el desarrollo en C#. Posteriormente se realizará la implementación completa en el entorno de desarrollo más adecuado para el fin pretendido.

La herramienta propuesta deberá satisfacer un conjunto de necesidades comunes a todos los clientes de la plataforma de desguaces, por lo que se establecerá un método de desarrollo que facilite la generación de prototipos suministrables a los clientes para el continuo reajuste de la herramienta final.

Tras la implementación, y paso a producción, se evaluará la satisfacción de los clientes mediante encuestas anónimas donde podrán transmitirnos el feedback de la nueva herramienta.

El resultado de esta iniciativa, así como el nivel de aceptación entre los usuarios finales, promoverá o no, la migración de otras características de la plataforma de desguaces a plataformas móviles.

Para cubrir los objetivos planteados, este documento se estructura en varias secciones: Empezaremos realizando un estudio de mercado y analizando la operativa del sector de los desguaces, así como las normativas que los regulan. A continuación evaluaremos una alternativa de monetización del producto, seguiremos realizando una especificación de la aplicación en base a su arquitectura, su lógica de negocio y el diseño de su interfaz gráfica. Posteriormente detallaremos aspectos de su implementación, así como las herramientas y metodologías de desarrollo utilizadas. Y finalmente detallaremos los resultados de su explotación y realizaremos las conclusiones finales.

### **Consideraciones previas.**

Por explícito cumplimiento de políticas comerciales de la empresa, de ahora en adelante, el nombre del producto propuesto “ScrAPPed Cars” en el ámbito académico, pasará a llamarse CRVNetCheckAPP, distinguiendo entre sus actualmente dos extensiones; CRV Almacén (para el tratamiento de piezas) y CRV Campa (para vehículos).

## 2 Estudio del mercado y operativa del sector.

### 2.1 La Normativa Europea y su aplicación en España

#### 2.1.1 Legislación Europea

Todos los años, los remanentes de vehículos desechados en desguace generan entre 8 y 9 millones de toneladas de residuos en la Comunidad Europea, que deben gestionarse correctamente. En 1997, la Comisión Europea adoptó una propuesta de Directiva con el objeto de realizar el desmontaje y reciclado de vehículos más ecológicos, estableciendo objetivos claros y cuantificados de reutilización, reciclado y valorización de los vehículos y de sus componentes, empujando a los productores a la fabricación de vehículos nuevos con vistas a su reciclabilidad.

Esta legislación fue adoptada oficialmente por el Parlamento Europeo y el Consejo en septiembre de 2000 y se publicó en el Diario Oficial L 269 del 21 de octubre (Directiva 2000/53/CE - la "Directiva de VFU« ). Pudiendo ser consultada al completo on-line en la siguiente referencia: [\(PARLAMENTO EUROPEO Y DEL CONSEJO, 2005\)](#)

De esta directiva aplicable a los vehículos de motor (en la mayoría de sus variantes), se extrae que:

- Los Estados miembros establecerán sistemas de recogida de los residuos procedentes de vehículos. Además, velarán para que los vehículos al final de su vida útil se transfieran a instalaciones de tratamiento autorizadas.
- El propietario o poseedor del vehículo al final de su vida útil deberán recibir un **certificado de destrucción** cuando el vehículo es transferido a una instalación de tratamiento autorizada. Este certificado es expedido por la institución. Permite al propietario o titular a dar de baja el vehículo con las autoridades públicas. Los costos o parte de los gastos incurridos en la transferencia de un vehículo a un centro de tratamiento serán sufragados por los fabricantes de vehículos.
- Los Estados miembros organizarán el almacenamiento y tratamiento de vehículos al final de su vida útil, de acuerdo con los requisitos de la *Directiva marco sobre residuos* (EUROPE DIRECT, 2009) y las del Anexo I de la Directiva. Instalaciones de tratamiento autorizadas despojarán vehículos al final de su vida útil antes del tratamiento y recuperar todos los componentes peligrosos para el ambiente.



- Debe darse prioridad a la reutilización y la recuperación (reciclaje, regeneración, etc.) de los componentes del vehículo. El objetivo de esta Directiva es aumentar la tasa de reutilización y recuperación.
  
- Además quedan fijados unos objetivos en cuanto a la tasa de reutilización y **recuperación** (de peso medio por vehículo y año) debe llegar a:
  - o 85% antes del 1 de enero de 2006.
  - o 95% antes del 1 de enero de 2015
  
- La tasa de reutilización y **reciclado** (en peso medio por vehículo y año) debe llegar a:
  - o 80% antes del 1 de enero de 2006.
  - o 85% antes del 1 de enero de 2015

### 2.1.2 Aplicación de las directivas europeas en España.

La presente directiva obliga a los Estados miembros a presentar un informe a la Comisión cada tres años sobre la base de esta información. Estos cuestionarios deberán permitir a la Comisión observar posibles cambios en la estructura del tráfico de vehículos de motor y de la recogida, desmontaje, las industrias de trituración, valorización y reciclado. Dentro de los nueve meses siguientes a la recepción de los informes de los Estados miembros, la Comisión publica un informe sobre la aplicación de la presente Directiva.

Según los plazos marcados, los últimos datos que los Estados Miembro han reportado a la Comisión Europea son los relativos al año 2009. A continuación se muestra una gráfica con los datos relativos al año 2009 publicados por la Agencia Eurostat de la Comisión Europea.

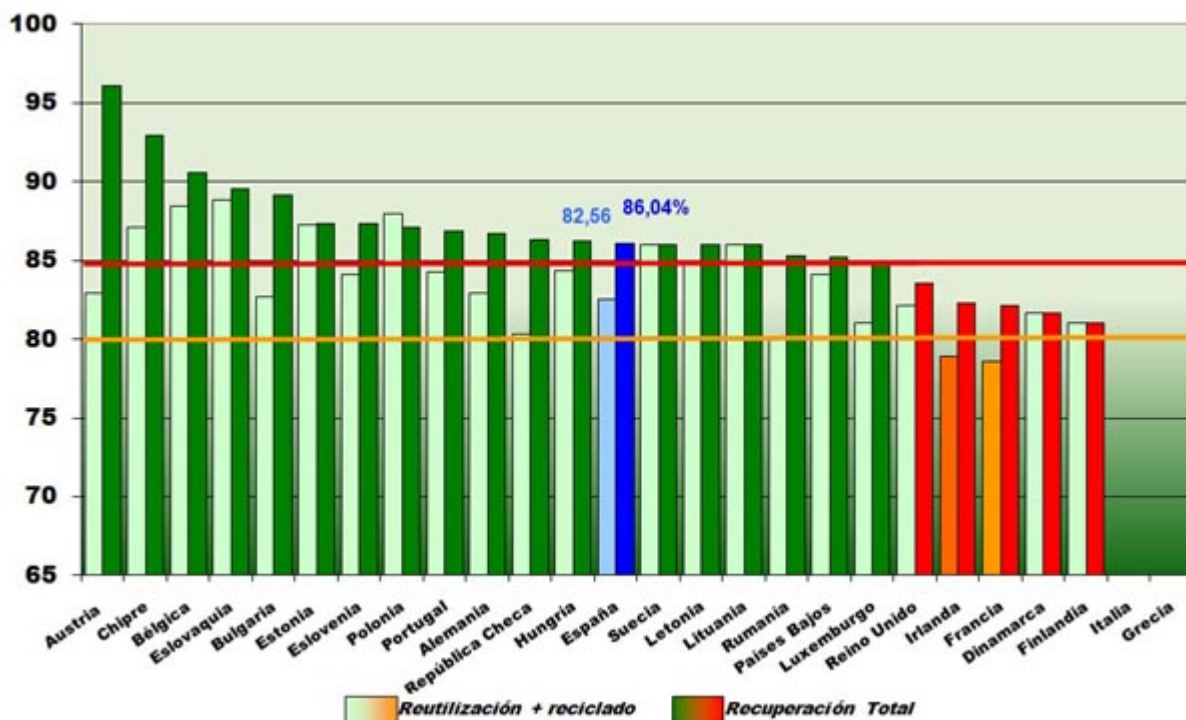


Ilustración 1: Porcentajes de recuperación del peso total de los vehículos desechados en los países de la UE.  
Fuente: SIGRAUTO.

Como se puede comprobar en los datos publicados por la Comisión Europea (EUROPA, 2013), los datos de España son 82,56% de reutilización y reciclado y 86,04% de recuperación total, habiendo alcanzado, por tanto, los objetivos del 80 y 85% fijados para el periodo 2006-2015. En estos momentos hay 5 países que han reportado no cumplir alguno de los dos objetivos, dos que no han reportado y 19 que sí cumplen ambos (entre los que está España).

Los datos oficiales reportados por el Ministerio de Medio Ambiente español muestran una evolución constante. Los avances que se han producido en los dos últimos años y los que seguro se reflejarán en los datos a reportar por España referentes a 2010 y 2011, son debidos fundamentalmente a algunos avances en la línea de la valorización energética de determinadas fracciones obtenidas tras la fragmentación de los vehículos descontaminados y desmontados.

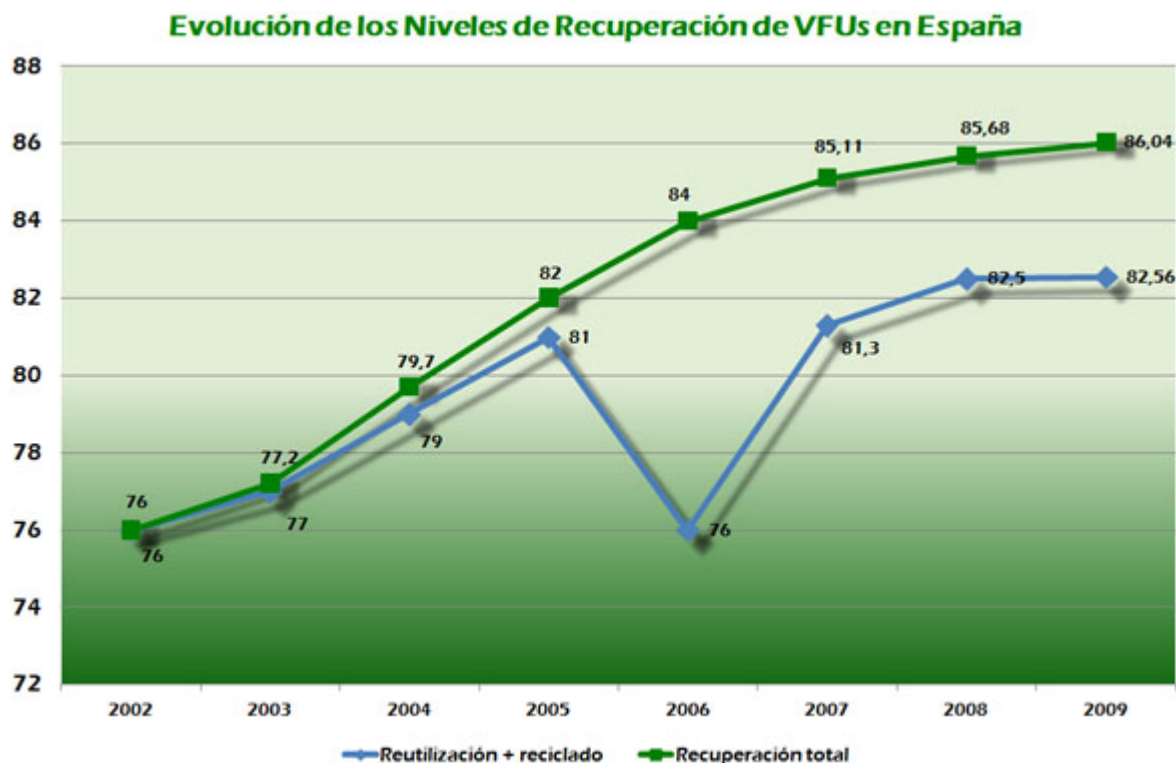


Ilustración 2: Evolución de los niveles de recuperación de VFU en España. Fuente: SIGRAUTO

Durante el año 2012, la red de CATs (Centros Autorizado de Tratamiento de vehículos fuera de uso) y Fragmentadores concertada por los fabricantes e importadores de vehículos en España a través de SIGRAUTO ha continuado aumentando en lo que a CATs se refiere. La cifra de Fragmentadoras se mantiene más o menos estable ya que se trata de un número reducido de instalaciones en nuestro país y todas ellas pertenecen a la red. En las gráficas siguientes puede verse la evolución que las respectivas cifras han sufrido desde la creación de la red en el año 2003.

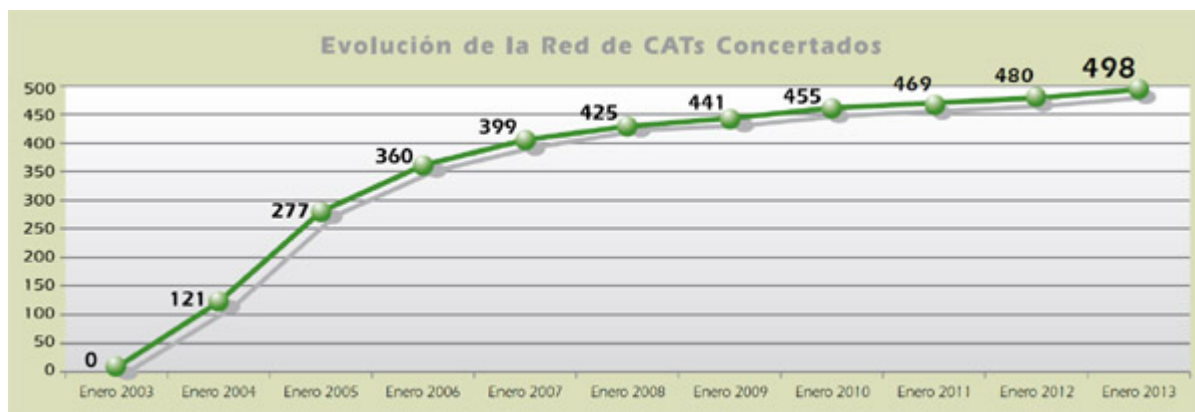


Ilustración 3: Evolución de la cantidad de CATs acreditados en España. Fuente: SIGRAUTO

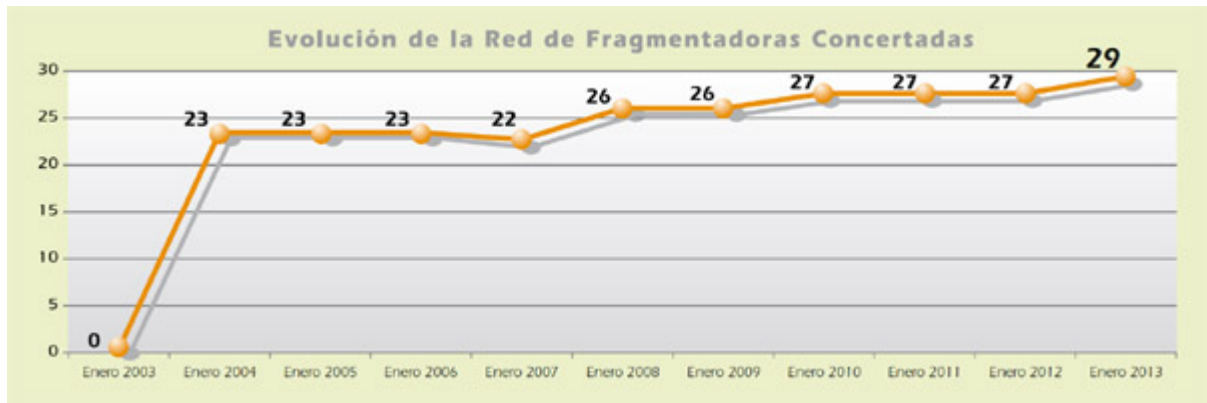


Ilustración 4: Evolución del número de Fragmentadoras concertadas en España. Fuente: SIGRAUTO

De ambas gráficas se puede concluir que, con los 498 Centros Autorizados de Tratamiento (CATs) y las 29 plantas fragmentadoras (situación a Enero 2013), la red concertada por los fabricantes e importadores de vehículos tiene capacidad suficiente para ofrecer una cobertura total del territorio español y que cualquier usuario puede entregar su vehículo en una de estas instalaciones sin problemas.

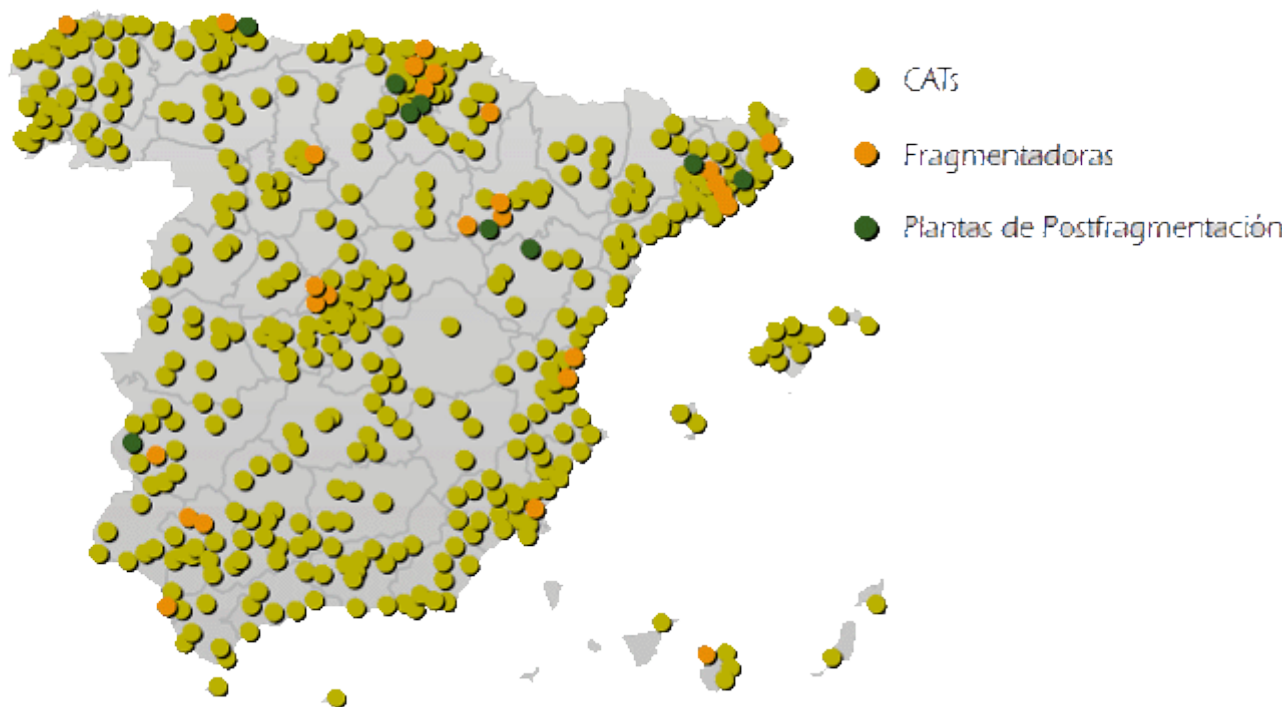


Ilustración 5: Reparto geográfico en España de CATs, fragmentadoras y plantas de postfragmentación. Fuente: SIGRAUTO

Los centros autorizados de tratamiento de la Red concertada son los que mayor volumen de vehículos tratan anualmente y prueba de ello es que, a pesar de suponer algo más del 50% en número frente al total de centros autorizados en España, los 498 centros de la Red concertada

por los fabricantes e importadores de vehículos trataron aproximadamente el 85% de los vehículos que se dieron de baja en 2012 y tienen capacidad sobrada para tratar el 100%.

### 2.1.3 Centros colaboradores en España.

La Asociación Española para el tratamiento medioambiental de los vehículos fuera de uso (SIGRAUTO) nace por acuerdo de las asociaciones que representan a los principales sectores involucrados en la cadena de tratamiento de los vehículos al final de su vida útil. SIGRAUTO es un foro permanente, en el que fabricantes e importadores, desguaces y fragmentadores de automóviles, analizan los problemas que afectan al tratamiento de los vehículos al final de su vida útil, buscando las soluciones más adecuadas y tratando de proporcionar a sus asociados los instrumentos necesarios para que puedan cumplir sus nuevas obligaciones medioambientales relacionadas con dicho tratamiento. Entre sus socios podemos encontrar:

- **La Asociación Española de Desguazadores y Reciclaje del Automóvil (AEDRA)**

AEDRA tiene aproximadamente 530 desguaces afiliados en todo el territorio español de los cuales más de 300 están autorizados como Centro de Tratamiento de Vehículos Fuera de Uso de acuerdo con las exigencias del Anexo I de la Directiva 2000/53/CE sobre vehículos al final de su vida útil. En la figura se muestra la distribución geográfica de los desguaces asociados a AEDRA:

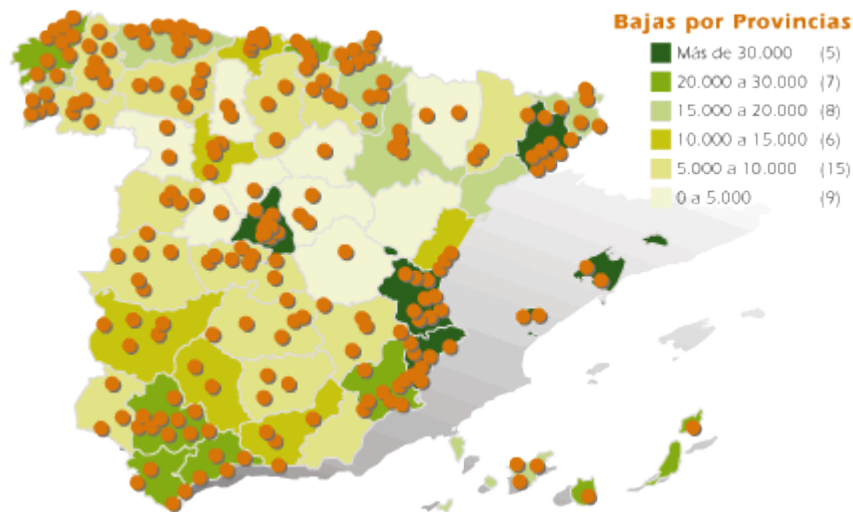


Ilustración 6: Reparto por provincias del número de bajas de vehículos en España. Fuente: SIGRAUTO

- **La Asociación Española Nacional de Fabricantes de Automóviles y Camiones (ANFAC)**

ANFAC es una asociación profesional sin ánimo de lucro que tiene por objeto fomentar el adecuado desarrollo del Sector de Automoción en España, y colaborar en la defensa de sus intereses, promoviendo la colaboración entre los fabricantes asociados en los aspectos técnicos, económicos, financieros y en cuantos otros se consideren

necesarios para el progreso armónico del sector. Para ello asume la representación y gestión colectiva de sus miembros asociados ante la Administración y ante toda clase de entidades e instituciones públicas y privadas.

ANFAC cuenta con 12 empresas asociadas cuyas plantas de producción se muestran en la siguiente figura:



Ilustración 7: Ubicación de las empresas asociadas a ANFAC. Fuente: SIGRAUTO

- **La Asociación Nacional de Importadores de Automóviles Camiones Autobuses y Motocicletas (ANIACAM)**

ANIACAM es una Asociación sin ánimo de lucro cuyo objetivo primordial es la defensa de los intereses profesionales de sus asociados y su perfeccionamiento en todos los aspectos. Representa a todos los importadores y su colectivo está compuesto por 41 marcas. Entre sus objetivos también se incluyen los de asesorar y participar en todos los niveles y Organismos que sea conveniente su presencia.

- **La Federación Española de la Recuperación (FER)**

La Federación Española de la Recuperación está formada por las principales empresas españolas de reciclado y recuperación de materiales metálicos. Además, esta asociación agrupa a las 26 plantas fragmentadoras españolas. En la figura se muestra la localización de las plantas fragmentadoras pertenecientes a la Federación y se puede apreciar que éstas se encuentran distribuidas de forma muy similar a la distribución de bajas:

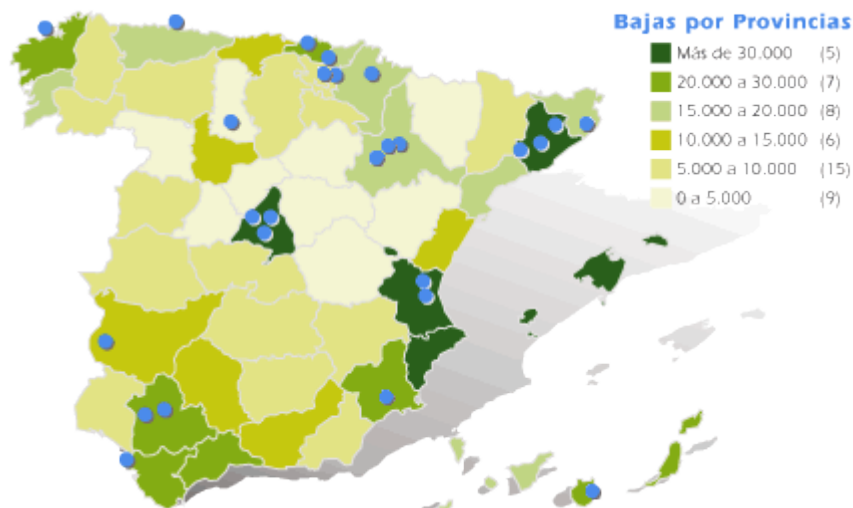


Ilustración 8: Reparto geográfico de las empresas fragmentadoras autorizadas en España. Fuente: SIGRAUTO

#### 2.1.4 Las obligaciones del propietario de un vehículo en el momento de su baja.

Cuando el propietario de un vehículo decide deshacerse de él, de acuerdo con el Real Decreto 1383/2002, tiene la obligación de entregarlo en un Centro Autorizado de Tratamiento (o en una instalación de recepción) para su baja y destrucción convirtiéndolo en un vehículo fuera de uso. Estas instalaciones deben contar con toda una serie de medios técnicos fijados en la normativa vigente y provienen en su mayoría de la adaptación/reconversión de los antiguos desguaces a estos requisitos. La verificación de que las instalaciones cumplen con los requisitos mínimos establecidos en la normativa se lleva a cabo por los órganos ambientales competentes de las Comunidades Autónomas que son los encargados de conceder las autorizaciones como gestores de residuos peligrosos en su territorio.

España cuenta con aproximadamente 900 CATs que cubren perfectamente todo el territorio tal y como se muestra en la siguiente figura. Los fabricantes e importadores de vehículos en España, cumpliendo con las obligaciones impuestas por la Directiva 2000/53/CE y por el Real Decreto 1383/2002 han concertado una red de centros autorizados de tratamiento común a todos ellos a través de SIGRAUTO en la que se garantiza la entrega gratuita a todos los usuarios. A día de hoy, la red concertada por fabricantes e importadores a través de SIGRAUTO cuenta con cerca de 440 centros y recoge en sus instalaciones el 80% de los vehículos que se dan de baja anualmente en España.





Ilustración 9: Localización de los CATs en España. Fuente: SIGRAUTO



## 2.2 Los clientes.

### 2.2.1 El cliente objetivo de nuestra plataforma software.

Los explícitos requisitos que deben cumplir los desguaces para recibir la consideración de *Centros Autorizados de Tratamiento de vehículos al final de su vida útil* (CAT), por parte de las autoridades estatales, acota notablemente el cómputo de las empresas capacitadas para poder gestionar estos servicios dada la inversión necesaria para el cumplimiento de las directrices europeas. A continuación detallamos los requerimientos técnicos:

Las instalaciones de recepción de vehículos y las de gestión de los vehículos al final de su vida útil tienen que cumplir los siguientes requisitos técnicos:

A) Los lugares de recepción y almacenamiento, incluso temporal, previo a la descontaminación de los vehículos al final de su vida útil, deberán disponer de:

1. Zonas adecuadas al número de vehículos a almacenar y dotadas de pavimento impermeable, con instalaciones para la recogida de derrames, de decantación y separación de grasas.
2. Equipos para el tratamiento de aguas, incluidas las pluviales, que han de ser tratadas conforme a la reglamentación sanitaria y medioambiental antes de deshacerse de las mismas.

B) Las instalaciones de los centros autorizados de tratamiento que procedan a la descontaminación dispondrán de:

1. Zonas cubiertas adecuadas al número de vehículos a descontaminar con pavimento impermeable y con instalaciones para la recogida de derrames, de decantación y de separación de grasas.
2. Zonas cubiertas y con pavimento impermeable para almacenar los componentes retirados del vehículo y que estén contaminados, en especial para aquellos que estén impregnados de aceite.
3. Contenedores adecuados para almacenar las baterías (con neutralización del electrolito allí mismo o en sitio próximo para casos de accidente), filtros y condensadores de PCB/PCT.
4. Depósitos adecuados para almacenar separadamente los fluidos de los vehículos al final de su vida útil, es decir: combustible, aceite de motor, aceite de cajas de cambio, aceite de transmisión, aceite hidráulico, líquidos de refrigeración, líquido anticongelante, líquido de frenos, ácido de baterías, fluidos del equipo del aire acondicionado y cualquier otro fluido contenido en el vehículo.
5. Equipos de recogida y tratamiento de aguas, incluidas las de lluvia en las zonas no cubiertas, las cuales han de ser tratadas previamente a su vertido, de conformidad con

la normativa ambiental y sanitaria establecidas por las distintas Administraciones públicas.

6. Zonas apropiadas para almacenar neumáticos usados, que incluyan medidas contra incendios y prevención de riesgos derivados de almacenamientos excesivos.

C) Las instalaciones de reciclado y fragmentación posterior a la descontaminación de los vehículos al final de su vida útil, en las que se pueda dar lugar a lixiviados de sustancias peligrosas por agua de lluvia, tendrán zonas cubiertas y dotadas de pavimento impermeable, así como equipos de recogida de aguas sucias y pluviales, que serán tratadas de conformidad con la normativa ambiental y sanitaria.

En ocasiones algunos de los posibles clientes, carecerán del cumplimiento de alguno de los requisitos legales o técnicos y formará parte de nuestra misión comercial; el asesoramiento y orientación de las decisiones del cliente en aras de facilitar su incorporación a nuestra cartera de clientes.

La última actualización de los datos de centros concertados (tanto CAT como Fragmentadores) a fecha de hoy 15 de agosto de 2013, se tiene del 8 de julio 2013 y permite consultar por comunidades autónomas los centros de las distintas provincias españolas a través de un Localizador Web que podemos encontrar en la siguiente referencia. ([SIGRAUTO: Asociación Española para el tratamiento medioambiental de los vehículos fuera de uso, 2013](#))

A modo de ejemplo, tras consultar los centros autorizados en la Comunidad Valenciana, hemos extraído una lista con los datos proporcionados por SIGRAUTO de los centros pertenecientes a la provincia de Valencia:

<b>PROVINCIA DE VALENCIA</b>					
<b>LOCALIDAD</b>	<b>EMPRESA</b>	<b>DIRECCIÓN</b>	<b>CP</b>	<b>TELÉFONO</b>	
Alcudia de Crespins	DESGUACES GONZÁLEZ C.B.	POL. IND. EL CANARI, PARC 3	46690	96 224 50 12	Autorizado
Almoines	AUTO DESGUACE ALMOINES, S.L.	C/ FERRERS, 9. POL. IND. ALMOINES	46723	96 280 42 70	Autorizado
Alquería de la Comtessa	DESPIECE SAFOR, S.L.	C/ SOTAIA, 20 y 22	46715	96 289 32 96	Autorizado
				www.despiecesafor.com	
Benaguasil	DESGUACE CORTES, S.L.	CTRA. TAVERNES DE VALLDIGNA S/N	46180	96 273 73 92	Autorizado
				www.desguacecortes.es	
Carcaixent	DESGUACES EL CORDOBÉS	C/ MAYOR S/N (COGULLADA)	46740	96 243 34 99	Autorizado
				www.desguaceselcordobes.com	
Cullera	DESGUACE GARCÍA C.B.	CTRA. FAVARA, 42	46400	96 172 58 33	Autorizado
				www.desguacegarcia.es	
La Pobla de Vallbona	UVI DEL AUTOMÓVIL, S.L.	C/ PARIS, 11. POL. IND. MAS DE TOUS SECT. II-1	46185	96 276 09 26	Autorizado
Liria	DESGUACE NALDO	CTRA. ADEMUZ, KM. 31,5 Salida 30	46160	96 279 07 14	Autorizado
Manises	GRÚAS Y DESGUACES MANISES, S.L.	C/ MARINA BAIXA, 2. POL. IND. LA COVA	46940	96 153 39 19	Autorizado
				www.desguacemanises.es	
Masalaves	DESGUACE DON OCASIÓN V.O., S.L.	CTRA. N-340 KM. 867	46680	902 11 75 42	Autorizado
Massamagrell	RECICLAUTO S.L.	POL. IND. MASAMAGRELL C/ Nº 4	46130	96 145 13 11	Autorizado
Museros	DESGUACE EUROPA 2000, S.L.U.	CMO. LA CALDERONA, POLIG. 13 PARC 110	46136	96 144 10 78/8	Autorizado
				www.desguaceeuropa2000.com	
Náquera	DESGUACE ALEGRE, S.L.	C/ TRAMONTANA, 5. POL. IND. LOS VIENTOS	46119	96 160 90 09	Autorizado
				www.desguacealegre.com	

Ilustración 10: Centros Autorizados de Tratamiento de VFUs en la provincia de Valencia (1)

## INTEGRACIÓN DE TECNOLOGÍAS MÓVILES EN EL NEGOCIO DE LOS DESGUACES DE VEHÍCULOS

Paterna	LAJO Y RODRÍGUEZ, S.A.	C/ CIUDAD DE CARTAGENA, 17. POL. IND. FUENTE DEL JARRO	46988 www.lyrsa.es	96 132 05 05	Autorizado
Puig	JOSÉ JAREÑO, S.A.	POL. IND. Nº7 C/ LES ROTES, 3	46540 www.josejareno.es	96 147 24 54	Autorizado
Quartell	DESGUACE HNOS. PELLICER, S.L.	POL. IND. LA FOIA. MANZANA 5 PARC 6	46510 www.desguacepellicer.com	96 11 01 868	Autorizado
Real de Gandía	RECYCLING GANDÍA, S.L.	CAMI CEMENTERI BENIPEIXCAR S/N	46727	607 44 14 67-	Autorizado
Ribarroja del Turia	DESGUACE MALVARROSA, S.L.	POL. IND. LA REVA. C/ MALLORCA, 5	46129 www.desguacemalvarrosa.com	902 21 20 40	Autorizado
San Juan de Énova	DESGUACES EL CORDOBÉS, S.L.	POL. IND. SANT JOANET C/ SENIA, 3	46669 www.desguaceelcordobes.com	96 259 13 25	Autorizado
Silla	DESGUACES SILLA, S.L.	C/ VENT DEL LLEBEIG, 29	46460	96 121 64 79	Autorizado
Valencia	AUTODESGUACE VALENCIA S.L.	CAMI DELS ROCHS, 73	46013 www.autodesguacevalencia.com	96 374 54 05	Autorizado
Xàtiva	BELLVER PLA, S.L.	POL. IND. PLA DE LA MEZQUITA, NAVE 14	46800 www.bellverpla.com	96 228 15 42	Autorizado
Masalfasar	HIERROS JOSÉ AGUILAR MARCO, S.L.	C/ MITJERA, 7. POL. IND. MEDITERRÁNEO	46560	96 140 01 66	En Trámite. No tramita bajas

Ilustración 11: Centros Autorizados de Tratamiento de VFUs en la provincia de Valencia (2)

A las que se suman las siguientes plantas fragmentadoras:

<b>VALENCIANA, COMUNIDAD</b>				
<b>VALENCIA</b>				
<b>Localidad</b>	<b>Empresa</b>	<b>Dirección</b>	<b>Cod. Postal</b>	<b>Teléfono</b>
Silla	JAP-2 RECUPERACIONES, S.L.	CTRA. ALCASER, 3	46460	96 120 03 79
Sollana	FERIMET, S.L.	PARTIDA ELS CLOTS CTRA. 331 S/N	46430 www.gcelsa.com	96 174 08 82

Ilustración 12: Fragmentadoras acreditadas en la provincia de Valencia

## 2.2.2 Funciones de los Centros Autorizados de Tratamiento de Vehículos Fuera de Uso.

### a) Recepción, verificación y baja

Para que un centro autorizado de tratamiento pueda recibir en sus instalaciones un vehículo fuera de uso e iniciar su tratamiento, el titular del mismo debe entregarlo junto con la documentación para que el centro pueda comprobar su titularidad y verificar que no existe ningún impedimento para proceder a su baja administrativa y a su tratamiento medioambiental.

Es importante destacar que la baja definitiva de los vehículos de turismo y de los comerciales ligeros de menos de 3.500 kg sólo puede tramitarse a través de los CATs y que en el resto de casos la baja puede realizarse tanto en las Jefaturas de Tráfico como en los CATs. Además, a partir de Enero de 2009 el sistema de tramitación de las bajas desde los CATs se va a realizar de forma telemática pudiendo el usuario recibir del CAT el documento acreditativo de la baja en el mismo momento de la entrega.

**Bajas de Vehículos (BATEVE)**

Solicitud de Baja > Forma solicitud > Justificante solicitud > Justificante salida

**Datos del Solicitante**

Nombre: CENTRO AUTORIZADO TRATAMIENTO VEHICULOS  
 CBF: S 12345678  
 Jefatura: Palma de Mallorca

**Datos del Vehículo**

Matrícula: 9955 ABC  
 Bastidor: 756C29 6 últimos caracteres  
 DNI/NIF/NIE Titular: 12345678Z  
 DNI/NIF/NIE Cotitular 1\*\*): 84567890A  
 DNI/NIF/NIE Cotitular 2\*\*): 72720001R  
 Tipo de Baja:  Voluntaria  Oficio por atención  
 Provincia: Arita Apuntamiento: Arita  
 Fecha de la baja: 06/11/2007

\*\* Los cotitulares informados deberán coincidir con el número exacto de cotitulares del vehículo.

---

**BAJA DE VEHÍCULO**

Motivo: 002000 Fecha Generación: 15/02/2008 14:30

**DATOS GENERALES**

Fecha Matriculación: 26/5/2008 Jefatura: MADRID  
 Placa: CANCELAS LEON/SANESA  
 Marca: Modelo: 807  
 Tipo: VEHICULO TURISTICO Bastidor: 1230420000025

**DATOS BAJA DEFINITIVA**

Nombre CATV: CARLOS Identificador Made Anterior: 912345678912345678  
 CATV CIF: 787654321  
 Fecha de Baja: 15/02/2008 Provincia: 28  
 Tipo de Baja: VOLUNTARIA Ayuntamiento: 28013

**DOMICILIO DEL VEHICULO**

Municipio: BUDU Puesto: BELVISO BUENO  
 Calle y N°: LG SAN BERNABE 14 Puesto: PONTEVEDRA Cod Postal: 36

Este informe y formulario pertenece al Centro de Tratamiento Autorizado. CATEG 01, en cumplimiento de lo dispuesto en la Orden INT/4311, de 20/10/2007 (BOE 14/11), ha procedido a la baja definitiva de un vehículo que se indica anteriormente en el Registro Telemático de la Dirección General de Tráfico. Hasta la fecha y con respecto a dicho vehículo, el Registro General de Tráfico podrá proporcionar la información de la baja en el Registro de la Dirección General de Tráfico. La actividad de este documento puede considerarse obsoleta al ser un informe de verificación de documentos. Disponible en la web de la Dirección General de Tráfico: <http://www.dgt.es>

Ilustración 13: Aplicación para la tramitación de bajas e informe de baja tramitada. Fuente: DGT

### a) Descontaminación y fomento del reciclado

Una vez que el vehículo es recibido por el CAT y se han realizado los trámites administrativos necesarios, éste pasa a ser un residuo peligroso y debido a los distintos materiales y fluidos que lo componen, debe procederse en primer lugar a su descontaminación, que consiste en la retirada de todos los líquidos y elementos que tienen la consideración de residuo peligroso. Estos son los aceites hidráulicos, aceites del motor, del diferencial y de la caja de cambios (salvo que se reutilice el bloque completo, en cuyo caso se puede mantener lubricado)

combustibles, líquidos de frenos, anticongelantes, filtros, baterías, etc. Asimismo, al objeto de facilitar el reciclado, se retiran también algunos residuos no peligrosos como catalizadores, neumáticos, vidrios, etc.



Ilustración 14: Zona de Descontaminación de VFUs en un CAT

Todos los elementos retirados deben almacenarse en depósitos adecuados para su posterior entrega a los gestores autorizados de tratamiento de los mismos que se encargarán de su recuperación en plantas especializadas.

### c) Reutilización

El siguiente paso es la retirada de todos aquellos elementos que todavía están en condiciones de ser utilizados para reparar otros vehículos. En esta fase, se evalúan y retiran todos aquellos componentes susceptibles de ser reutilizados y son claramente identificados y almacenados para su posterior comercialización.

Debe destacarse que la reutilización es la vía más respetuosa con el Medio Ambiente para recuperar cualquier producto o material y como tal es la primera opción en la jerarquía de recuperación establecida por la Comisión Europea. España cuenta con un mercado muy desarrollado y muy profesional que es también uno de los factores de los altos niveles de recuperación que se alcanzan.





Ilustración 15: Almacenes de piezas y componentes recuperados listos para reutilizar.

#### d) Compactación y envío a la planta de fragmentación

El último de los procesos que se lleva a cabo en los centros autorizados de tratamiento es la compactación de los vehículos una vez que han sido descontaminados y desmontados. El fin de este proceso es minimizar el volumen de los vehículos para optimizar el transporte de los mismos hasta las instalaciones de fragmentación.



Ilustración 16: Compactación de VFUs y almacenamiento en una Fragmentadora.

## 2.3 Alternativas en el mercado.

El sector de los desguaces de vehículos cuenta con varias alternativas de aplicaciones verticales que resuelven de manera concreta las tareas más frecuentes en dichos centros. A parte de la plataforma de gestión de desguaces CRVNet, encontramos en el mercado español, algunas alternativas de la mano de pymes autóctonas y otras varias soluciones globales extranjeras.

### 2.3.1 Proveedores de software de gestión de CATs.

A continuación conoceremos a algunos de los competidores directos de CRVNet, así como los productos que están implantando y las tecnologías de las que disponen. Dicha experiencia nos servirá para posteriormente extraer características candidatas a ser implementadas en nuestro software.

#### 2.3.1.1 Gira: *gVisualRec*, *gVisualTal*.

Es una empresa que nació en Madrid en 1990 al amparo de Repuestos Macry con el objetivo de desarrollar un producto informático adaptado a la tienda de recambios. Este programa debía tener unos costos adecuados a sus posibilidades y que diera una solución apropiada a la informatización de este tipo de comercio.

Actualmente GIRA cuenta con una Tienda Virtual integrada con el software de gestión *gVisualRec* (Gestión Integral del Recambio. Versión Windows) y cuyas características son una herramienta para la gestión diaria de negocio, permitiendo tener información acerca de la rentabilidad de la empresa

Dispone de un programa de Gestión Integral del Taller – Versión Windows (*gVisualTal*) que se encarga de controlar un taller mecánico con todas las reparaciones que realiza diariamente, y va desde el seguimiento de las reparaciones de los vehículos de sus clientes hasta el tiempo invertido por los operarios en cada una de las reparaciones.

Y, por último, dispone también de un programa de Contabilidad, adaptado a las necesidades actuales. Versión Windows (*gVisualCon*).

Podemos encontrar más información en: <http://www.gira.net>



### 2.3.1.2 Gestor CAT-VFU 2008.

Gestor CAT-VFU 2008 es una Aplicación de Gestión Integral de CATVFU.

Partiendo de las generalidades de una aplicación para gestión empresarial contiene Gestión Multi-Empresas (se pueden generar distintas empresas), Gestión Multi-Usuario, Módulo de Clientes (Albaranes, Facturas, Presupuestos, Pedidos, Tickets), Módulo de Proveedores (Albaranes, Facturas, Presupuestos, Pedidos, Tickets), Módulo de Almacenes (permite gestionar múltiples almacenes), Traspasos de Artículos entre almacenes. Los módulos específicos para la Gestión de CAT-VFU son:

- Módulo de Modelos de Vehículos:

Gestiona los datos genéricos de los modelos:

- o Versiones.
- o Motorizaciones.
- o Grupos de artículos.
- o Artículos.

La Aplicación contiene una base de datos inicial de 400.000 referencias de artículos, pudiéndose ampliar por el propio usuario.

- Módulo de Gestión de Vehículos Fuera de Uso (VFU):

Gestiona todos los datos necesarios para la tramitación de los VFU:

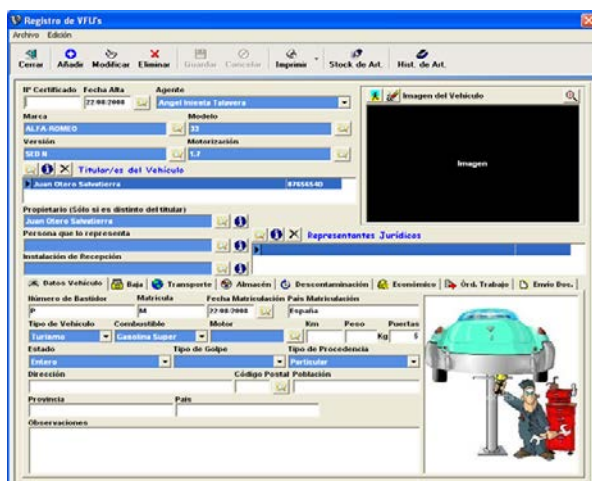
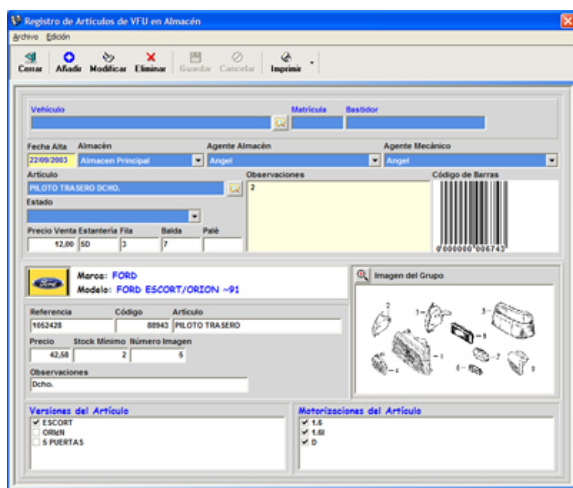


Ilustración 17: Gestor CAT-VFU 2008 - Módulo de Gestión de vehículos

- Solicitud de Baja.
- Certificado de Destrucción.
- Parte de Residuos.
- Informes para Jefatura de Tráfico.
- Informes para Medio Ambiente.
- Memoria Anual de Residuos.
- Orden de despiece según las estadísticas de ventas y stock en almacén del modelo versión y motorización del vehículo tramitado.

- Módulo de Gestión de Artículos de VFU:

Gestiona los Artículos de VFU para venta:



- Generación de Etiquetas de Código de Barras.
- Colocación en Almacén.
- Precio del Artículo.

Esta aplicación pertenece a la segunda generación de Aplicaciones de Gestión de Desguaces desarrollada por Calzada Soft, por lo que está evolucionada sobre una versión anterior con 4 años de uso en Centros CAT.

Ilustración 18: Gestor CAT-VFU 2008 - Módulo de Gestión de Artículos

Podemos encontrar más información en: <http://www.calzadasoft.com/vfu.htm>

### 2.3.1.3 AzelerCat.

AzelerCat es una empresa tecnológica especializada en soluciones de negocio para el Sector Automoción. Ha desarrollado el Portal Europeo de Recambios de Automóvil, [www.azelerecambios.com](http://www.azelerecambios.com) y un sistema de Gestión Integral de Vehículos para Centros Autorizados de Tratamiento (CATs), además de soluciones para el área de ventas, postventa, taller, peritación y tasación.

Por otro lado Azeler, en colaboración con AEDRA (Asociación Española de Desguazadores y Recicladores del Automóvil) y el apoyo del Ministerio de industria y Turismo, ha desarrollado un conjunto de soluciones a medida para el sector del Desguace (Centros Autorizados de Tratamiento-CAT). El sistema de Gestión Integral (SGI) permite al CAT gestionar todos sus procesos de forma automática mediante los siguientes módulos:

- Gestión de entrada de vehículos.
- Gestión de despiece de vehículos.
- Gestión de almacenes (multialmacén, multiubicación).
- Tratamiento y gestión de residuos.
- Gestión de contenedores.
- Gestión de clientes y proveedores.
- Gestión de facturación.
- Gestión de cobros.

- Gestión de pedidos.
- Gestión de repartos.
- Gestión de venta por mostrador.
- Gestión de devoluciones.
- Análisis de rentabilidad.

El almacén de recambio recuperado, denominado AzelerCAT, recibe automáticamente la información de inventario del CAT. Para unificar los criterios de entrada y búsqueda de recambios, Azeler ha desarrollado un Catálogo Electrónico a medida del colectivo. Este Catálogo Electrónico está permanentemente actualizado vía Internet y permite, entre otras cosas, tener al día las distintas codificaciones de Marcas, modelos, versión y códigos de motor.

Este servicio permite al usuario final, normalmente un profesional del sector o cliente particular, tener acceso al almacén virtual de recambio recuperado que cuenta con el respaldo de la asociación del sector (AEDRA) y del Ministerio de Industria

Podemos encontrar más información en: <http://www.azeler.com/cat.aspx>

### **2.3.1.4 RENET.**

RENET AUTOMÓVIL, S.L. es una empresa especializada en maquinaria y herramientas para el sector de la recuperación que ofrece variedad de maquinaria, herramientas e informática para solucionar cualquier problema o tema en relación a la organización, descontaminación, desmontaje, reciclaje y almacenaje de VFU para los Centros de Reciclaje, Chatarrerías, Desguaces, etc. que buscan soluciones para un tratamiento profesional de VFU.

Están ubicados en San Sebastián (País Vasco) y trabajan por toda España y Portugal, representando a fabricantes Alemanes, y ofreciendo la descontaminación y el desmontaje de VFU.

Sus productos:

Para satisfacer las necesidades software de sus clientes, cuentan con un programa informático para desguaces llamado AOL-Kfz.

Esta herramienta se adapta a las necesidades de los desguaces del Automóvil y refleja todo el proceso de los desguaces desde la entrada de vehículos, registro automático de piezas, tanto desmontadas como no desmontadas, búsqueda para la venta, hasta elaborar facturas, recibos, informes, etc. El programa permite controlar la organización interna de los desguaces, facilitando los trabajos legales-administrativos y gestionar las piezas nuevas y usadas para la venta.

Podemos encontrar más información en <http://renet-automovil.es/productos/inform%C3%A1tica/>

### 2.3.1.5 Pinnacle Actual Systems.

Pinnacle Actual Systems se creó en 1985 en el Reino Unido para dar soporte a la industria de la fotografía mediante el desarrollo de un sistema informático para comunicarse con las grandes y costosas impresoras a color utilizadas ampliamente en esta industria. Proporcionado tanto el hardware como software de este producto.

No fue hasta el año 1995 cuando Pinnacle se sumergió en el software de gestión de desguaces, hasta el día de hoy, teniendo en la actualidad más de 300 clientes (en Europa, Australia y EE.UU) y siendo ésta la alternativa más fuerte que hemos encontrado.

Un programa de utilidad directa donde el usuario puede personalizar su pantalla basada en la función de su trabajo que desempeña. Los vendedores pueden mostrar todas sus citas abiertas, órdenes de trabajo, facturas, órdenes de compra y mucho más para gestionar el flujo de su carga de trabajo. El personal de inventario puede visualizar todos sus vehículos que no han llegado y vehículos que deben ser inventariados. El centro de control se actualiza al instante. Opciones: rellamadas, facturas, estadísticas dinámicas, facturas de depósito pendientes de pago, órdenes de compra, órdenes de compra de crédito, cotizaciones, órdenes de trabajo sin confirmar, viejas órdenes de trabajo, personalizable de órdenes de venta, personalizable Solicitar Factura, Vehículos No Llegó, los vehículos no inventariados, resolución por bastidores, saldos de clientes, y los pagos recibidos.

La característica más destacable se basa en la oferta de un software de gestión de acceso competo a través del web, permitiendo el acceso a las piezas y vehículos directamente a través de Internet, o realizar copias de seguridad remotas.

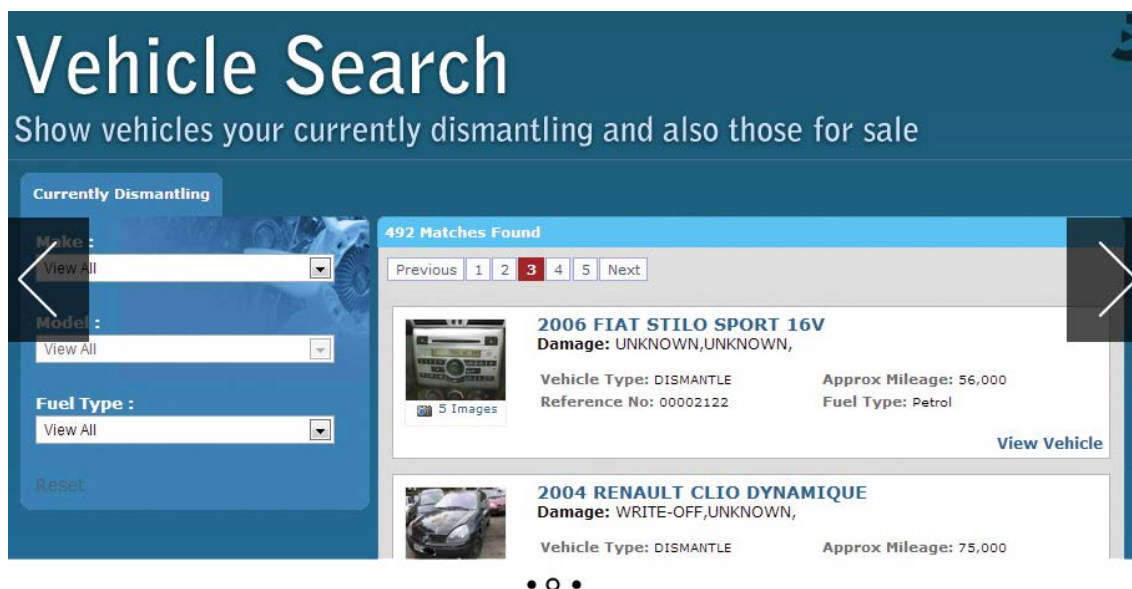


Ilustración 19: Captura de pantalla de la aplicación on-line de Pinnacle Actual Systems

Podemos encontrar más información en <http://www.actual-systems.com/solutions>.

### 2.3.1.6 *GestioneP Europeizas.*

GESTIONE P Europeizas, es un programa de gestión para empresas de recambios (especial para CATS) creado en León (España) en el año 2000, que está diseñado para todo tipo de empresas que venden o gestionan piezas, recambios, repuestos, tanto de automóviles, como de motos e incluso para maquinaria. Desde un ordenador, se puede realizar la gestión interna de todo el almacén local y la posterior venta de las piezas.

Permite realizar las siguientes funciones:

- Despiezar el vehículo de manera rápida al disponer de todos los catálogos de las marcas de vehículos, y las piezas más comunes del vehículo; pudiendo añadir las que se deseen.
- Impresión de las etiquetas de piezas y vehículos además de las estanterías con código de barras.
- Configurar estanterías virtuales, así como la ubicación de las piezas dentro del programa, conforme a las estanterías reales del almacén del cliente.
- Realizar órdenes de trabajo, para llevar el control del despiece del vehículo.
- Exportar el listado de todo el Stock a un archivo de Excel o XML ordenado como mejor venga, bien por modelo del vehículo o por nombre de la pieza.
- Llevar un registro de Proveedores y Clientes, así como de gastos.
- Imprimir etiquetas de piezas y clientes y utilizar un lector de código de barras.
- Realizar Facturas, Tickets, Albaranes, Notas de Entrega, Facturas Rectificativas, Tickets Rectificativos, Facturas Proforma, Facturas de Compra con varias series, obteniendo diversos balances de contabilidad.
- Hacer contratos de Compra/Venta y si es un gestor autorizado, los certificados de destrucción y de baja del vehículo; Listados para Medio Ambiente, Tráfico y Guardia Civil.
- Llevar un control y gestión de los contaminantes y de los contenedores de los que se disponga.
- Contiene una libreta de pedidos que permite realizar un seguimiento de cada solicitud de piezas; descargar peticiones del buzón de Europeizas.com, realizar búsquedas on-line y realizar peticiones de piezas.
- Posibilidad de buscar on-line la pieza que se necesite accediendo a la base central de datos o realizar una petición insertándola en el buzón de demandas.
- Realizar una valoración teórica y otra real de los contaminantes generados en un periodo además de múltiples informes.
- Está preparado para trabajar en Red, si se dispone de varios ordenadores.
- Permite la sincronización de los almacenes locales del cliente con el portal Europeizas.com de manera automática.

Podemos encontrar más información en:

<http://www.europeizas.com/software/softwarecats.aspx>

## 2.3.2 Programas similares

Dadas las características tan concretas del mercado al que se destina el Software de Gestión de Centros de Tratamiento de Vehículos al final de su vida útil, con la circunstancia de que nuestro desarrollo consiste en una aplicación vertical, definida para una plataforma, donde las opciones del cliente ya son de por sí limitadas, provoca que el desarrollo de una aplicación para dispositivos móviles en este ámbito, tenga una competencia escasa o nula, no encontrando ningún software similar dentro de la plataforma Android.

Si analizamos otras plataformas, podemos encontrar dos iniciativas de software para plataformas móviles por parte de las empresas GIRA y Pinnacle Actual Systems, la que apenas hemos encontrado información dado que parece que tiempo atrás quedó en desuso.

Pero sí hemos encontrado una solución bastante desarrollada en Windows Mobile 5.0: GVisualPDA, de la empresa Gira.



Ilustración 20: Imagen de GVisualPDA de Gira

Dicha alternativa software, ocupa el mismo lugar que nuestro proyecto dentro de un software de gestión de escritorio. A continuación detallaremos sus características y debatiremos sus ventajas e inconvenientes respecto a ser una alternativa a nuestra APP:

GVisualPDA se ofrece como una aplicación para Windows Mobile 5 y 6 que amplía las posibilidades de integración del software de gestión GVisualRec (el software de escritorio) permitiendo estar conectados en tiempo real a los operarios de almacén y a los agentes comerciales

Permite realizar búsquedas por referencias, por índices y la lectura de códigos de barras de referencias desde el lector.

El Módulo de Almacén ofrece las funcionalidades para la realización de:

- Inventario
- Recepción de Pedidos
- Entrada de Material
- Inventario Rápido
- Preparación de Albaranes
- Inventario por Fabricante

**Ventajas de GVisualPDA:**

- Integración con otras APPs de la empresa (Módulo de Agente).
- Posibilidad de comunicación entre dispositivos (mensajería interna).
- Preparación de albaranes desde el mismo dispositivo.
- Aplicación con un nivel de madurez alto.
- Plataforma estabilizada.

**Inconvenientes de GVisualPDA:**

- Imposibilidad de adjuntar fotos a las piezas
- Plataforma obsoleta.
- Dispositivos industriales con elevado precio y bajas prestaciones.
- Interfaz desfasada y complicada al uso.
- Imposibilidad de mostrar toda la información en una sola pantalla.
- Poca precisión en los menús al interactuar sin un lápiz táctil para realizar las pulsaciones en pantalla.
- Requiere un entrenamiento avanzado para su uso.

La alternativa de Pinnacle:

Conforme hemos comentado, ha sido difícil encontrar alguna información sobre la propuesta móvil de Pinnacle Actual Systems , salvo alguna captura de pantalla que no devela ninguna característica relevante de su potencial. Sabemos que dichas aplicaciones funcionan sobre Windows Mobile 5.

Ante tal desconcierto, dado el potencial de esta empresa, más que demostrado a través de la globalización de sus licencias vendidas, decidimos ponernos en contacto con su departamento técnico, donde recibimos noticias de que ya han empezado el desarrollo de aplicaciones para la plataforma Android.

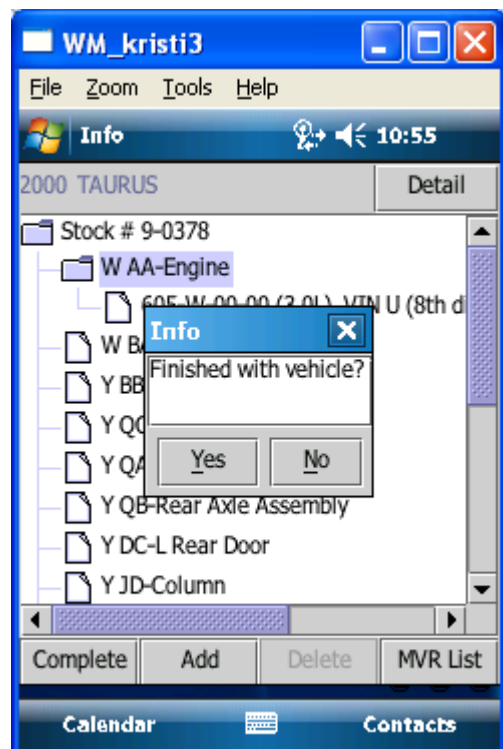


Ilustración 21: Captura de pantalla de la herramienta móvil de Pinnacle Actual Systems

Teniendo en cuenta las alternativas de las principales y únicas empresas competidoras en el desarrollo de aplicaciones móviles para este sector, podemos asumir que nuestro desarrollo será pionero en ofrecer dicha característica a través de la tecnología actual. Dado que con el hardware contemporáneo a Windows Mobile 5, no podía presumir de aportar unas especificaciones mínimas para realizar tareas básicas como tomar fotografías de calidad, disponer de autofocus, un “led” para proporcionar un flash, potencia de proceso, etc.



### 2.3.3 Ventajas para los desguaces.

El uso de las tecnologías móviles puede convertirse en una ventaja tecnológica para cualquier cliente, dado que cualquier teléfono inteligente o Smartphone actual, dispone de características físicas fácilmente explotables desde el punto de vista del software para este sector. A continuación listaremos una serie de ventajas que podemos encontrar:

- Agilización del proceso de recopilación de datos en general.
- Digitalización automática de las órdenes de trabajo (ya no se requiere disponer de las hojas impresas en papel para los mecánicos), Anotaciones, observaciones varias...
- Posibilidad de convivencia del sistema de recopilación de datos antiguo con el nuevo.
- Flexibilidad en la obtención de datos (Se pueden introducir piezas que no están asociadas a ningún vehículo, simplemente anotando sus referencias, sus atributos y tomando fotografías).
- Acceso en tiempo real a las características de una pieza con sólo leer el código de barras de la misma o consultando a través de la referencia de fabricante, junto con la posibilidad de actualizar datos de las piezas ya introducidas en cualquier momento.
- Asociación de fotografías a las piezas en el mismo momento que se recopilan los datos (minimización de errores en las asociaciones).
- Disminución del número de procesos de negocio para la recopilación de datos. (Tanto la introducción de piezas nuevas como las procedentes de órdenes de despiece se pueden registrar en el sistema aplicando el mismo proceso.)
- Posibilidad de registrar un vehículo de manera offline desde el dispositivo móvil, sin disponer de una referencia del mismo, para con posterioridad enviar los datos a la herramienta de escritorio y tratar desde allí el caso.
- Posibilidad de acceder al estado de la pieza a través de su código de barras. (Vendida, Pendiente de revisión, Almacenada).
- Posibilidad de reubicar las piezas desde el dispositivo móvil, a través de la lectura automática de los códigos de barras de piezas y su asociación con los códigos de barras de estantería o almacén.

## 2.4 Establecimiento de los requisitos en base a prioridades MoSCoW.

En base a las características deseables encontradas en sistemas similares de este ámbito y un estudio de las necesidades de nuestros clientes, seleccionaremos, agrupando por prioridades, las distintas características que posteriormente implementaremos en nuestro software. A continuación describimos los requisitos agrupándolos en los niveles que especifica el método MoSCoW.

### **Must have**

Incluirá los requisitos que son de obligado cumplimiento por su importancia de cara al éxito global del proyecto. De cara a garantizar un producto de alta calidad, cumpliendo con las expectativas del cliente, se han seleccionado los siguientes requerimientos:

Dadas las características del entorno en el que nos encontramos, un entorno industrial, una de las prioridades de nuestra aplicación será maximizar la productividad por lo que se exigirá una excelente capacidad de respuesta en la interfaz de usuario.

Tolerancia 0 a errores de usuario. Cualquier información introducida deberá ser validada desde la parte del servidor impidiendo la introducción de información al sistema gestor de bases de datos comprometiendo la calidad de los datos.

Capacidad para la realización de trabajo de manera autónoma desde el dispositivo y la actualización asíncrona del servidor respecto al momento de la toma de datos.

Control de acceso de usuarios. Cualquier acción ejecutada en los dispositivos móviles y posteriormente actualizada en el servidor, incluirá información adicional que posibilitará la identificar la autoría de los datos.

### **Should have**

Incluye los requisitos que se consideran “de alta prioridad”, que en la medida de lo posible deberían ser incluidos en la solución final, pero que llegado el momento y si fuera necesario, podrían ser prescindibles si hubiera alguna causa que lo justificara, o en su defecto podrían no conseguirse en su totalidad.

Simplicidad. El software desarrollado debe enmascarar la complejidad del modelo de datos para poder ser utilizado sin dificultad por un público poco familiarizado con la informática, o en su defecto, con una formación mínima. La interfaz de usuario deberá guiar al usuario en el proceso de la toma de datos.

Inmunidad a la fragmentación de versiones de Android. En su defecto, alcanzar un porcentaje superior al 95% de los dispositivos disponibles actualmente en el mercado

Las aplicaciones proporcionadas deberán ofrecer vistas adaptadas a los dispositivos en modo horizontal y vertical y se deberán mostrar correctamente en distintas resoluciones de pantalla (tanto móviles como tabletas).

El software se podrá suministrar en una versión limitada para su evaluación por parte del cliente.

### **Could have**

Requisito deseables pero no necesarios, se implementarían si hubiera posibilidades presupuestarias y temporales, por lo que serán considerados en todo momento pero quedaría relegados a la consecución previa de requisitos de mayor importancia.

Si las circunstancias lo permiten, el software desarrollado podría incorporar características como animaciones (ligeras, sin que puedan afectar al tiempo de respuesta) para mejorar el feedback del usuario y asegurar las acciones realizadas.

Multi-idioma. Aunque el software de la plataforma CRVnet se distribuye principalmente en territorio nacional y la temática del mismo enfoca un sector lo suficientemente técnico como para que los términos utilizados sean conocidos, se propondrá la traducción al portugués y al catalán (para facilitarlos a los clientes minoritarios) y al inglés para facilitar la internacionalización del producto.

### **Won't have**

La lista de requisitos se completará con aquellos que no se considerarán en esta iteración de desarrollo, aunque posteriormente podrán ser tenidos en cuenta para su implementación.

El software proporcionado no mostrará errores del lado de servidor al usuario.

La interfaz de las aplicaciones no mostrará controles de usuario (como botones) que sean de uso minoritario.

El software desarrollado no debe posibilitar más de una manera de realizar una misma acción.

El software no podrá interactuar con la plataforma de escritorio y web sin la autorización y puesta a punto de nuestro departamento técnico.

### 3. Monetización

La plataforma CRVNet es una herramienta ampliamente distribuida por el territorio español y Portugal, y dicha herramienta ya dispone de un modelo de negocio capaz de rentabilizar los desarrollos actuales a través del cobro de cuotas mensuales en función de la selección del cliente de un conjunto de servicios (como mantenimientos, actualizaciones, soporte telefónico...).

Con la incorporación de las aplicaciones móviles, se genera un valor añadido en la forma de característica diferenciadora ante otras soluciones software para el mercado vertical de la gestión de Centros Autorizados de Tratamiento de vehículos al final de su vida útil (CATs).

La compra de licencias de herramientas de desarrollo, el aprovisionamiento del hardware necesario para pruebas (como tabletas, teléfonos...), el desarrollo en sí, y el futuro mantenimiento del mismo, así como la puesta a punto en las instalaciones del cliente o la inclusión de nuevas características, comportan un nuevo gasto en la empresa proveedora, que irremediablemente deberá repercutir en los clientes en aras de mantener el equilibrio a corto plazo.

El modelo de negocio que garantizará la rentabilidad de las aplicaciones móviles, se basará en conformar con éstas un pack seleccionable por el cliente dentro del catálogo de servicios ofertados, incrementando así la mensualidad del cliente, a cambio de las nuevas características recibidas.

Por otro lado, el cliente obtiene una herramienta que le agilizará el trabajo y le permitirá reducir la partida de recursos humanos dedicados a las acciones comprendidas en la aplicación móvil, obteniendo con ello, no sólo un balance económico enormemente favorable, sino una ventaja tecnológica que le diferencie ante sus competidores.

En la fase inicial de la puesta en producción de las aplicaciones móviles, dichas aplicaciones serán proporcionadas al cliente a un coste cero para promover la familiarización del mismo con la herramienta y la detección y corrección de errores por el equipo técnico. Con ello, se pretende acelerar el periodo de maduración de la herramienta y la detección de nuevos casos de uso implementables antes de la comercialización del producto.

## 4. Especificación

### 4.1 Arquitectura global de la APP

No podemos eludir que la aplicación móvil propuesta se deberá incorporar a un ecosistema ya activo, manteniendo la estabilidad del mismo y simplificando algunas de las tareas que ya se pueden realizar con anterioridad a la implantación.

Antes de realizar una propuesta de arquitectura de la aplicación que vamos a desarrollar, convendrá determinar con claridad la envergadura de la plataforma sobre la que se va a implantar. Por ello, a continuación presentaremos las distintas funcionalidades que podemos encontrar en el software de escritorio, para poder ubicar con posterioridad aquellas pertenecientes al nuevo desarrollo.

En dicha plataforma se aúnan una serie de herramientas construidas con la finalidad de simplificar la gestión de almacenes, facilitar la venta de stock del cliente, promover la venta B2B y promocionar el negocio del cliente. A continuación se muestra un esquema global de SIDEO.

#### 4.1.1 El entorno



Ilustración 22: Esquema de la arquitectura de SIDEO

### ¿Qué es SIDEО? (Sistema de DEsguaces Online)

- **SIDEО** es una plataforma que proporciona a los clientes nuevos canales de negocio con los que sacar más partido y difusión de sus almacenes de pieza, a la vez que maximiza la posibilidad de llegar a un público más amplio.
- Con esta plataforma el cliente tiene la posibilidad de hacer que sus piezas se vean en distintos canales de venta y en distintos almacenes virtuales con los precios que él mismo decide, además de permitirle con un sólo clic acceder a un gran número de piezas recuperadas y nuevas con las que completar sus pedidos.
- Los canales de venta pueden ser portales como **<http://www.recambioverde.es>**, aplicaciones de consulta y venta para agentes comerciales y repartidores, sucursales y tiendas presenciales propias y sus portales y tiendas Web.

#### **Ventajas para el cliente:**

- La plataforma SIDEО es un sistema que le asiste a la hora de vender sus piezas en diversos canales de negocio para así aumentar sus beneficios.
- Almacenes virtuales, posibilidad de compartir información y ventas entre almacenes remotos.
- Control de filtros de piezas para su subida al portal RecambioVerde
- Definición libre y personalizada de diversos canales de venta para el trabajo con comerciales, webs, catálogos, y todo tipo de profesionales del sector.
- Pagos y transacciones online.
- Facturación electrónica.

Elementos del software de escritorio CRVNet:

CRVNet es el nombre que recibe la aplicación de escritorio que permite a los clientes la gestión de sus desguaces. Esta aplicación será complementada con el nuevo software para dispositivos móviles. A continuación definiremos algunas de las funcionalidades que permite:

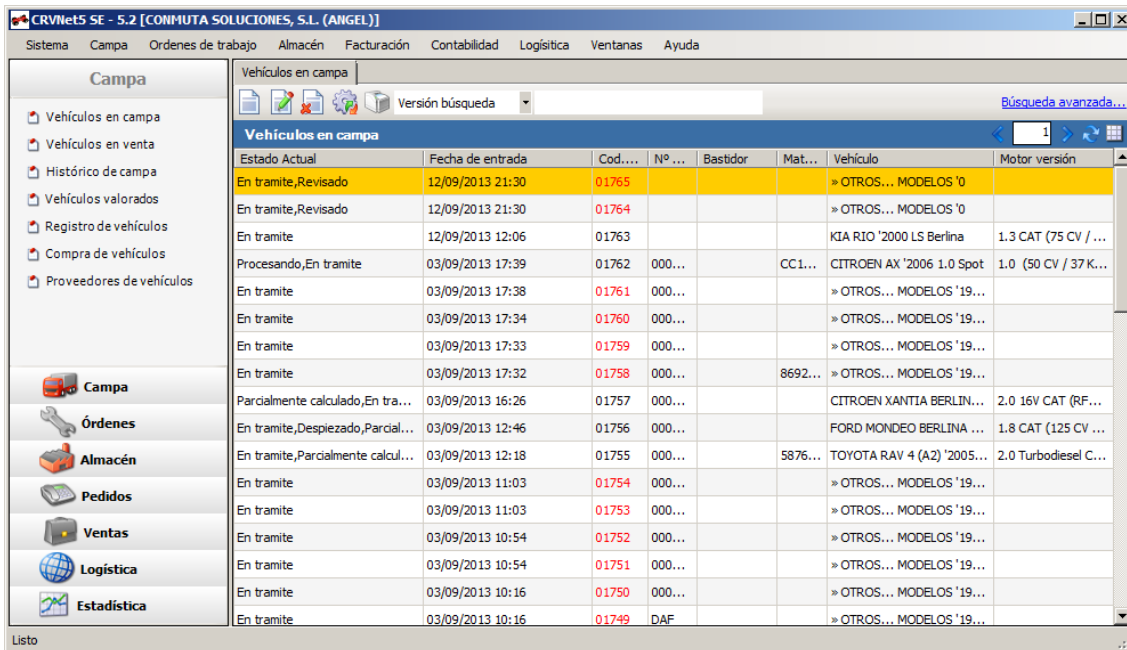


Ilustración 23: Captura de pantalla de la aplicación de escritorio CRVNet

Campa (Gestión de vehículos)

- Árbol de versiones de vehículos especializado para el sector actualizado online y personalizable por el usuario, incluye vehículos industriales y motocicletas. Identificación inmediata del vehículo por criterios como código de motor, potencia, bastidor, descripción...
- Sistema de gestión documental asociado a cada vehículo, documentos, imágenes, escáner y OCR. Preparado para la LOPD.
- Sistema de gestión de contaminantes: plantillas de contaminantes por vehículo y estadístico, informes, estadísticas, registro de justificantes.
- Gestión de aprovechamiento del vehículo.



- Compra de vehículos para despiece a través de SIDE0.
- Gestión de grúas: vehículos valorados, sistema de recogida y entrega en campa, registro de grúas, trabajadores y costes.
- Personalización y emisión de todos los impresos necesarios para la actividad del desguace: Documentos de venta, códigos de barras, documentación oficial y administrativa, documentos de compra de vehículos, declaraciones de conformidad, autorizaciones y certificados, justificantes, etc....
- Completa operatividad telemática: Bajas DGT, Certificados de Destrucción. Aplicación compatible con el estándar del programa ÉTER.

*NOTA: Sobre ésta área operará la aplicación móvil CRV Campa, permitiendo la consulta, creación y actualización de los vehículos del desguace.*

### Órdenes de trabajo (Gestión de personal)

01653 - Orden Despiece n° 2013-000203 Fecha: 03/09/2013

Vehículo	FORD FOCUS BERLINA (CAK) 1998 Trend		
Matrícula	Kilometraje	0.00	
Combustible	GASOIL	Tipo motor	ENC50
Color	Basidor	WFGAXXWPD42J57858	
Ubicación			
Trabajador	CONSULTA		

**DESCONTAMINACION**

ACEITE  ANTICONGELANTE  BATERIA USADA  
 FILTRO ACEITE  LIQUIDO DE FRENS  NEUMATICOS

**DESPIECE**

**MOTOR / ADMISION / ESCAPE**

AFORADOR  ABS  ESPEJO INTERIOR  
 ARBOL DE LEVAS  AMORTIGUADOR DELANTERO E  GUARNECIDO PUERTA DELANTI  
 BLOQUE  AMORTIGUADOR DELANTERO C  GUARNECIDO PUERTA DELANTI  
 BOMBA ACEITE  BALLESTA TRASERA  GUARNECIDO PUERTA TRASER  
 BOMBA AGUA  BARRA ESTABILIZADORA DELA  GUARNECIDO PUERTA TRASER  
 CARTER  BARRA ESTABILIZADORA TRAS  GUARNECIDOS PALANCA CAMB  
 CATALIZADOR  BOMBA FRENO  MANETA ELEVANAS TRASER  
 CIGUEÑAL  BRAZO SUSPENSION INFERIOR  MANETA INTERIOR DELANTERA  
 COLECTOR ADMISION  BRAZO SUSPENSION INFERIOR  MANETA INTERIOR DELANTERA  
 COLECTOR ESCAPE  BRAZO SUSPENSION INFERIOR  PARASOL DERECHO  
 CORREA DISTRIBUCION  BRAZO SUSPENSION INFERIOR  PARASOL IZQUIERDO  
 CULATA  DEPÓSITO SERVO  RELOJ CUENTA KILOMETROS  
 DEPÓSITO COMBUSTIBLE  DEPRESOR FRENO / BOMBA VA  SALPICADERO  
 DEPÓSITO EXPANSION  DISCO FRENO DELANTERO  TECHO INTERIOR  
 ENFRIDADOR ACEITE MOTOR  DISCO FRENO TRASERO  VOLANTE  
 FILTRO ACEITE  MUELLE AMORTIGUACION **ALUMBRADO**  
 FILTRO AIRE  PALANCA FRENO DE MANO  FARO ANTINEBLA DERECHO  
 FILTRO GASOIL  PINZA FRENO DELANTERA DER  FARO ANTINEBLA IZQUIERDO  
 INTERCOOLER  PINZA FRENO DELANTERA IZQ  FARO ANTINEBLA TRASERO IZ  
 INYECTOR  PINZA FRENO TRASERA DERE  FARO DERECHO  
 MOTOR COMPLETO  PINZA FRENO TRASERA IZQUIE  FARO IZQUIERDO  
 POLEA ARBOL LEVAS  SERVOFRENO  LUZ INTERIOR  
 POLEA GIGUEÑAL  TIRANTE DELANTERO DERECH  PILOTO LATERAL DERECHO  
 RADIADOR AGUA  TIRANTE DELANTERO IZQUIER  PILOTO LATERAL IZQUIERDO  
 RANPA INYECTORA  INTERIOR  PILOTO MATRÍCULA  
 TAPON COMBUSTIBLE  AIRBAG DELANTERO DERECHO  PILOTO TRASERO DERECHO  
 TENSOR CORREA AUXILIAR  AIRBAG DELANTERO IZQUIERDO  PILOTO TRASERO IZQUIERDO  
 TERMOSTATO  AIRBAG LATERAL DERECHO **CLIMATIZACION**  
 TUBO ESCAPE TRASERO  AIRBAG LATERAL IZQUIERDO  CALSFACCION ENTERA NORMA  
 VARILLA ACEITE  BANDEJA TRASERA  MANDO CALEFACCION / AIRE A  
 VOLANTE MOTOR  CENICERO  RESISTENCIA CALEFACCION  
 SUSPENSION / FRENS  CINTURON SEGURIDAD DELANT **ACCESORIOS**  
  CINTURON SEGURIDAD DELANT  ANTENA

**Ilustración 24: Orden de despiece previa a la implantación de nuestro software**

- Registro de trabajadores. Definición de permisos de acceso y vistas. Pertenencia a grupos y privilegios específicos.
- Definición de tareas, asignación de tareas a trabajadores, control de tareas....
- Gestión de partes de trabajo.

*NOTA: Sobre esta área operará la aplicación móvil CRV Campa, la cual permitirá la creación de órdenes, digitalizado y simplificando algunos de los procesos de negocio que hasta el momento han requerido de la interacción de varios usuarios y del intercambio de documentación física (formularios impresos en papel).*

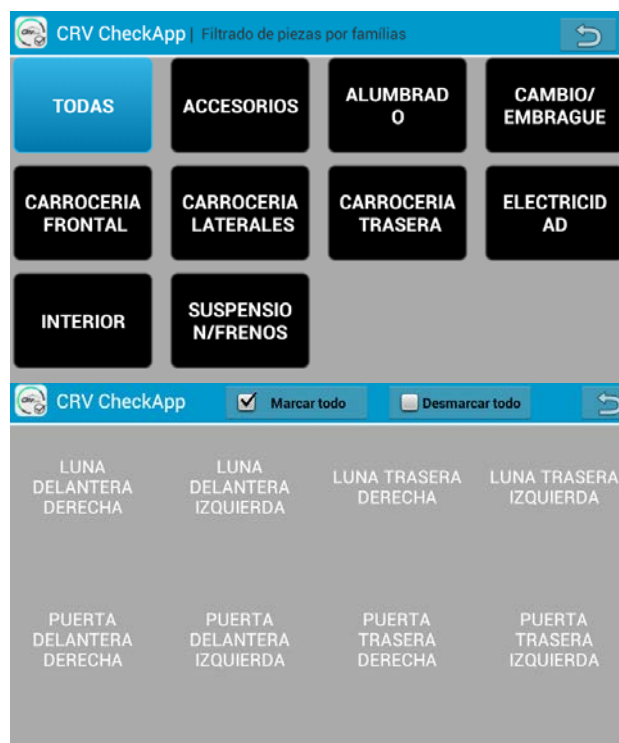


Ilustración 25: Capturas de pantalla del menú de selección de piezas para la generación de órdenes

## Almacén y portal

- Definición de plantillas de aprovechamiento y grupos de extracción sobre vehículos.
- Sistema de decisión de extracción de piezas.
- Reconocimiento asistido de las piezas extraídas a través de la conexión a las principales bases de datos de referencias del mercado.
- Control de filtros de piezas para su subida al portal RecambioVerde.
- Control de movimiento de piezas mediante código de barras y dispositivos periféricos.
- Historial de movimientos y eventos de la pieza (trazabilidad y gestión documental de la pieza).
- Número ilimitado de almacenes diferentes, organización y fácil localización de piezas.
- Fácil creación de nuevos almacenes y contenedores.
- Informes personalizados de stock de piezas.
- Asistentes de precios, de entrada y salida de material de referencias...

- Avanzado sistema automático de gestión de precios.
- Administrador de la plataforma SIDEO. Integración completa.

*NOTA: Sobre esta área actúa la aplicación móvil CRV Almacén, que permitirá la actualización o creación de piezas y la descarga de los conjuntos de piezas pertenecientes a una orden de trabajo.*

### Pedidos

- Fácil gestión de pedidos gracias al asistente de selección de piezas.
- Gestión pedidos vía online con soporte para varios formatos y envío por correo.
- Procesamiento de pedidos , tickets, facturación , recuperación de pedidos...
- Terminal Punto de Venta avanzado y de fácil uso.
- Integración de los almacenes virtuales en la búsqueda de piezas.

### Ventas

- Gestión de albaranes, facturas, tickets, abonos, albaranes de pago, facturas de pago...
- Facturación electrónica.
- Gestión de tipos de pago, tipos de IVA, grupos de clientes, grupos de proveedores.
- Gestión de cascos.
- Impresión de facturas múltiple.
- Gestión contabilidad: cuentas contables, definición de asientos, guías de traspaso, remesas bancarias, exportación a programas contables.
- Gestión de cajas: movimientos de cajas, listados, informes.
- Módulo de devoluciones adaptado específicamente para desguaces.

### Logística

- Gestión de repartos: rutas de entrega, repartidores, vehículos de reparto.
- Seguimiento de repartos.
- Definición de flota. Enlace con almacén, facturación y clientes, definición de rutas. Informes y estadísticas de repartos.

### Estadísticas

- Estadísticas especializadas para el sector y personalizables.
- Rentabilidad de piezas y vehículos.
- Estadísticas e informes sobre inventarios, ingresos, gastos, beneficios, cobros...
- Rotación de almacén.
- Exportaciones de todos los informes a **Excel** y **PDF**.

### Portal RecambioVerde.es



- **www.recambioverde.es**, es un portal de venta de piezas recuperadas al que tiene acceso gratuito cualquier desguace con una implantación adecuada de CRVNet.
- Al contrario que otros modelos de negocio, es el comprador el que se pone en contacto directamente con el desguace proveedor de la pieza, permitiendo un modelo escalable que mantiene el interés con independencia del número de desguaces adscritos a la iniciativa.
- A través de la plataforma SIDEO, los clientes de CRVNet pueden controlar qué parte de su almacén aparece en el canal de RecambioVerde pudiendo modificar el precio u otras características.
- Las piezas elegidas por los clientes, suben al portal en tiempo real y automáticamente sin necesidad de trabajo por su parte.

*Almacenes Virtuales*

- Cualquier desguace que use CRVNet puede vender piezas que no pertenezcan a su almacén, o realizar pedidos con éstas.
- Obviamente, dichos desguaces deben haber llegado previamente a acuerdos para definir qué piezas y a qué precio se van a compartir.
- Los almacenes virtuales proveen de un sistema fácil y directo para aquellos desguaces que quieran llegar a acuerdos entre ellos compartiendo su stock de piezas.
- Los almacenes virtuales permiten una fácil configuración para elegir con toda precisión qué parte del stock quiere el usuario compartir, con qué desguace quiere compartirlo y a qué precio quiere hacerlo.

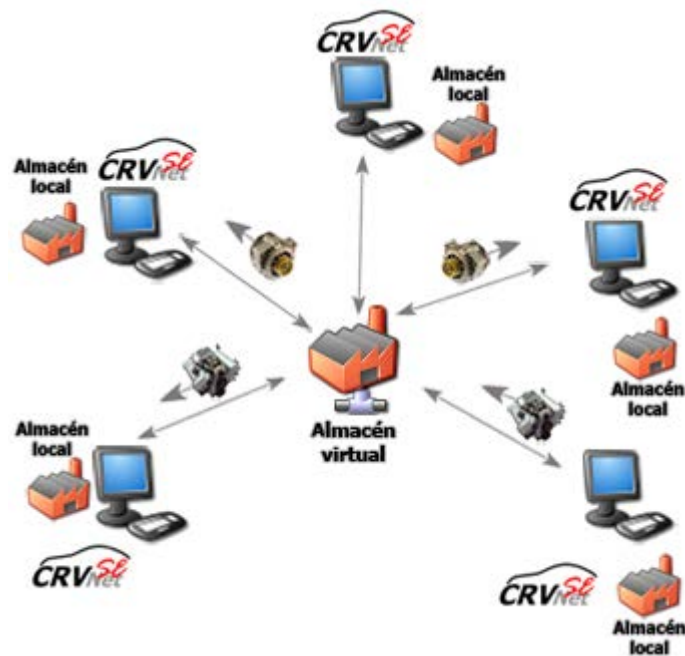


Ilustración 26: Diagrama del esquema de los almacenes virtuales

### 4.1.2 Arquitectura de la plataforma

Una vez queda bien definido el esquema de la plataforma, estamos en disposición de implantar nuestra propuesta de herramienta móvil.

La aplicación funcionará dentro del ámbito local de cada desguace, por lo que se deberán establecer protocolos de comunicación para que dicha herramienta sólo pueda interactuar dentro de las instalaciones donde ha sido configurada. Para ello se limitará a que la aplicación sólo permita acceder a los servicios a través de una conexión Wifi, que habilitará la conexión de la aplicación con el equipo servidor que almacena la información de stock del desguace.

Las aplicaciones móviles trabajarán paralelamente a las estaciones de trabajo (anteriormente ya disponibles en las instalaciones; portátiles, ordenadores de sobremesa) en la recogida de datos y gestión del material disponible.

Éstas aplicaciones realizarán una función complementaria al software de escritorio, permitiendo realizar operaciones similares pero con la virtudes a su favor de la movilidad, la sencillez y la eficacia que pueden proporcionar las aplicaciones móviles, fomentando con ello una mayor agilidad en la inserción, actualización y consulta del stock de la empresa.

El trabajo propiciado desde las aplicaciones móviles tendrá su efecto en el servidor local, que a su vez efectuará los cambios a modo de espejo en la plataforma SIDE0 y con ello se actualizarán otros canales como la web [recambioverde.es](http://recambioverde.es), almacenes virtuales.

Podemos ver el esquema propuesto a continuación:

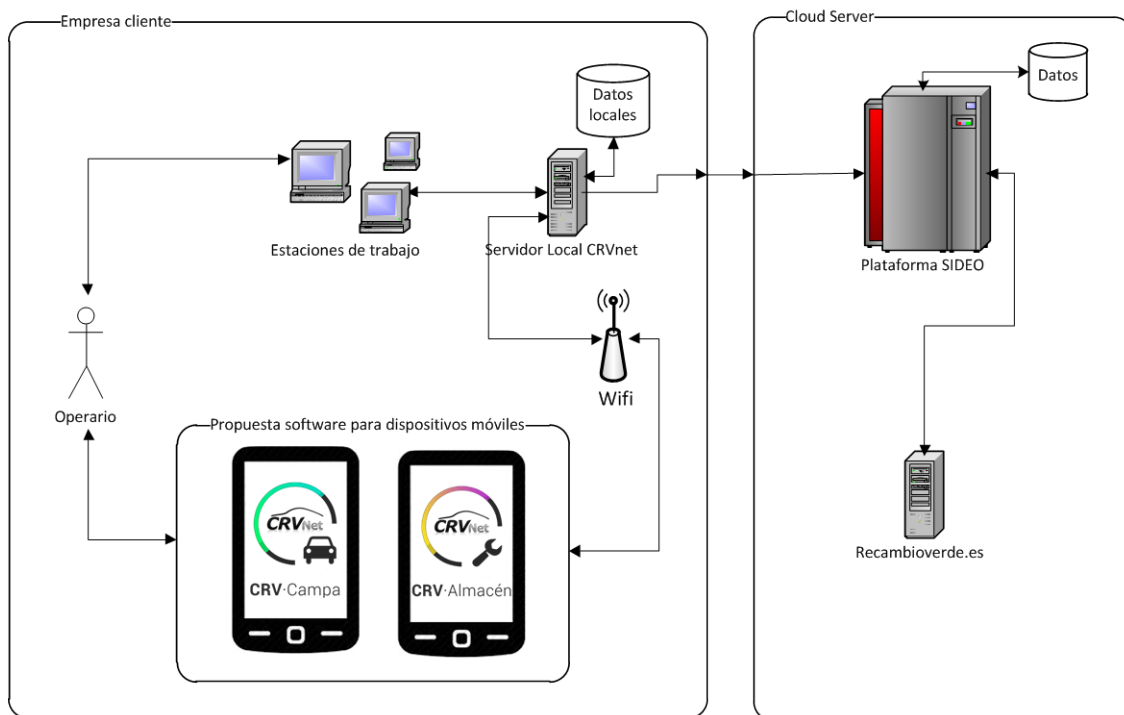


Ilustración 27: Situación de la APP dentro de la arquitectura de la plataforma CRVNet

## 4.2 Lógica de negocio.

Para la correcta especificación del sistema que se pretende implementar, se va a utilizar el estándar UML (Lenguaje Unificado de Modelado), por la amplia aceptación de la que dispone en la actualidad.

En el siguiente diagrama se pueden apreciar el actor (el operario que interactuará con la aplicación) y dos subsistemas diferenciados; los correspondientes a las aplicaciones CRV Almacén y CRV Campa.

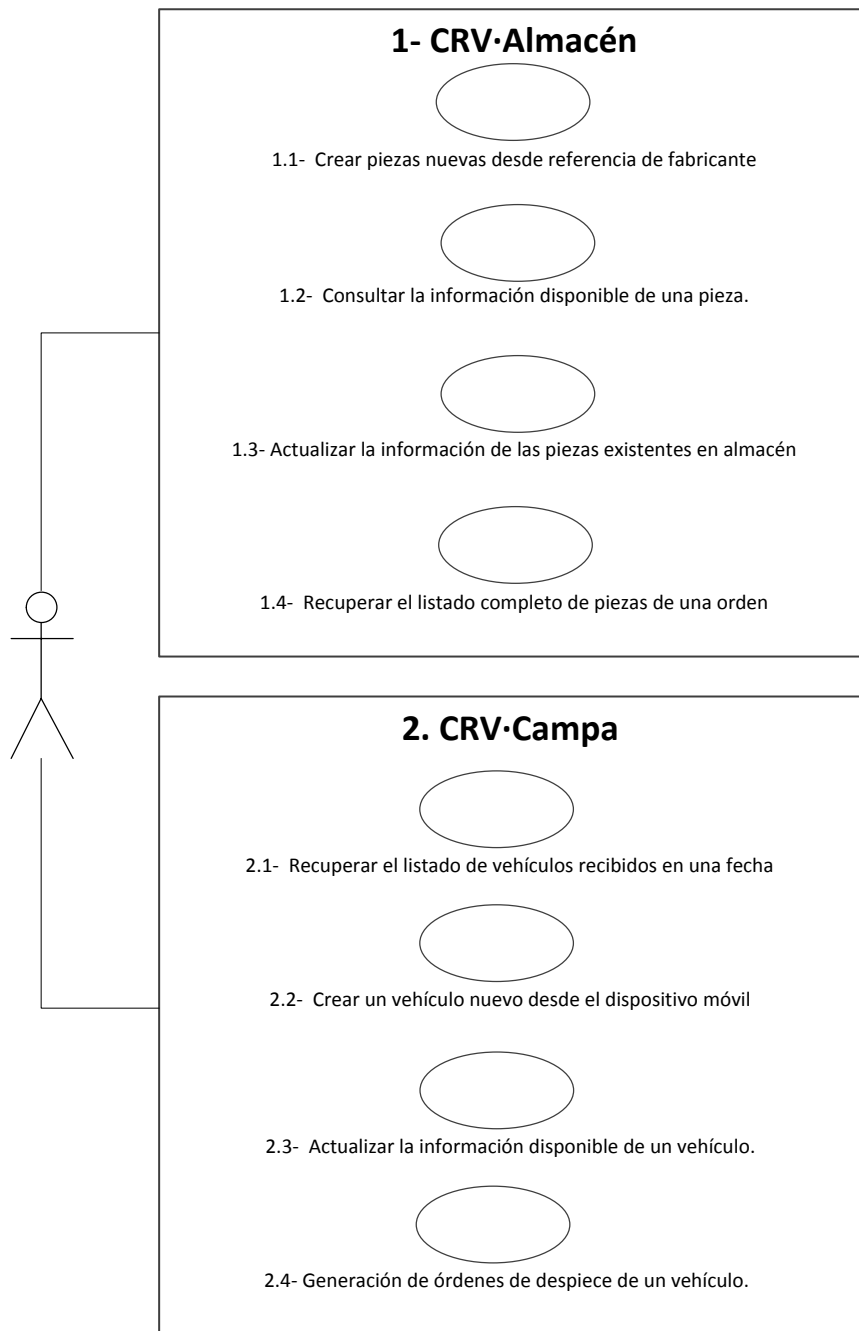


Ilustración 28: Diagrama de casos de uso

Los casos de uso describen el comportamiento de un sistema desde el punto de vista del usuario. Permitiendo definir los límites del sistema y las relaciones entre el sistema y el entorno. Además son descripciones de la funcionalidad del sistema e independientes de la implementación. A continuación detallaremos la descripción textual de los casos de uso identificados:

- Casos de uso relativos a CRV Almacén:

<b>Nombre</b>	<b>1.1 Crear piezas nuevas desde la referencia de fabricante</b>
<b>Actores</b>	USUARIO
<b>Propósito</b>	Facilitar la inserción de piezas en la base de datos a través de la consulta y resolución de una referencia de fabricante ya existente en la base de datos.
<p><b>Resumen:</b> El caso de uso se inicia cuando el Usuario introduce (ya sea a través del lector de código de barras o introduciendo manualmente) la referencia y consulta dicha referencia. El proceso genera una consulta en el servidor que informa al usuario de la existencia o no de dicha referencia. Devolviendo en caso positivo, una plantilla base del tipo de la referencia indicada preparada para que el usuario la complete. El caso de uso finaliza cuando el usuario ha finalizado la introducción de los datos y envía la pieza al servidor.</p>	
<b>CURSO NORMAL DE EVENTOS</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
<p>0. El usuario pulsa sobre el icono "Añadir" pieza.</p> <p>1. El usuario introduce una referencia de fabricante registrada en el sistema.</p> <p>2. El usuario consulta la referencia.</p> <p>4. El usuario introduce los datos referentes a la pieza que considera relevantes.</p> <p>5. El usuario guarda los cambios de la pieza y envía los datos al servidor.</p>	<p>3. El sistema consulta el tipo de pieza y proporciona al usuario la plantilla respectiva.</p> <p>6. El sistema detecta una nueva pieza sin código de pieza y con código de fabricante y genera un identificador interno que asigna al código de pieza para dar de alta una nueva pieza en el sistema.</p> <p>7. El sistema envía al usuario la verificación de que dicha pieza se ha generado con éxito.</p> <p>8. La aplicación elimina de la lista de piezas la citada pieza.</p>



<b>CURSOS ALTERNOS</b>	
En la acción 2	<p>El punto 2 no es una acción necesaria (por consiguiente tampoco lo es el 3). Por lo que se podrá trabajar Off-line con una plantilla por defecto, hasta el momento del envío al servidor que se realizarán las validaciones necesarias y se generará la nueva pieza fusionando los datos de la plantilla proporcionada por defecto y los de la plantilla perteneciente a la referencia en cuestión.</p> <p>Ante la posibilidad de que la referencia no exista se procederá como en el punto 3.</p>
En la acción 3	<p>Si la referencia no está en catálogo no se podrá generar la plantilla de la pieza hasta que la referencia se dé de alta en el mismo.</p> <ul style="list-style-type: none"> <li>▪ La consulta del usuario sobre la referencia proporcionará un aviso indicando la situación.</li> <li>▪ En intento de envío de la pieza al servidor dará un aviso mostrando la incapacidad para generar la pieza nueva.</li> <li>▪ La pieza permanecerá en el listado de piezas hasta que se corrija la referencia o se cree dicha referencia en la aplicación de escritorio.</li> </ul>
<b>Prioridad</b>	Alta
<b>Mejoras aportadas</b>	<ul style="list-style-type: none"> <li>▪ Facilita la creación tanto On-line como Off-line de piezas, desde la referencia de fabricante.</li> <li>▪ Permite crear piezas desde cualquier parte de las instalaciones del desguace, incluso desde aquellos puntos que se quedan fuera del alcance de la cobertura Wifi.</li> </ul>

<b>Nombre</b>	<b>1.2</b> Consultar la información disponible de una pieza
<b>Actores</b>	USUARIO
<b>Propósito</b>	Facilitar el acceso a la información de las piezas almacenadas en la base de datos a través de la consulta realizada desde el dispositivo móvil.
<p><b>Resumen:</b> El caso de uso se inicia cuando el Usuario introduce (ya sea a través del lector de código de barras o introduciendo manualmente) el código de pieza y consulta dicho código. El proceso realiza una consulta en el servidor que proporciona al usuario la información relativa a dicha pieza. El caso de uso finaliza tras la misma consulta. Pero en este punto el usuario podrá decidir si almacenar esa pieza en su dispositivo para su posterior acceso o descartar los datos recibidos.</p>	
<b>CURSO NORMAL DE EVENTOS</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
<p>0. El usuario pulsa sobre el icono "Añadir" pieza.</p> <p>1. El usuario introduce un código de barras registrado en el sistema.</p> <p>2. El usuario consulta el código de barras.</p> <p>4. El usuario revisa la información recibida de la pieza consultada.</p> <p>5. El usuario decide entre almacenar o descartar la información recibida.</p>	<p>3. El sistema consulta el código identificador de la pieza y proporciona al usuario la descripción respectiva.</p>
<b>CURSOS ALTERNOS</b>	
En la acción 3	Si el código de barras no está en catálogo no se devolverá información alguna de la pieza. Así mismo el usuario quedará notificado del resultado de la consulta.
<b>Prioridad</b>	Alta
<b>Mejoras aportadas</b>	<p>Facilita el acceso a datos relativos a la pieza que no podrían ser accedidos sin una consulta en un equipo de escritorio.</p> <ul style="list-style-type: none"> <li>▪ Permite verificar el contenedor de la pieza</li> <li>▪ Permite verificar el estado de la pieza: Vendida, Almacenada, Desechada, Estantería.</li> </ul>

<b>Nombre</b>	<b>1.3</b> Actualizar la información de las piezas disponibles en el almacén
<b>Actores</b>	USUARIO
<b>Propósito</b>	Facilitar la actualización de la información de las piezas almacenadas en la base de datos a través de la inserción de datos realizada desde el dispositivo móvil.
<p><b>Resumen:</b> El caso de uso se inicia a partir del caso de uso 1.2. Tras la consulta del código de pieza en el servidor y la recepción de la información, esta puede ser modificada (incluyendo nuevas fotos o borrando anteriores, modificando los campos textuales, o añadiendo nuevos campos) y reenviada al servidor. El caso de uso finaliza tras el envío de la información, que será un paso indispensable para la correcta actualización.</p>	
<b>CURSO NORMAL DE EVENTOS</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
<p>0. El usuario realiza las acciones pertenecientes al caso de uso 1.2</p> <p>1. El usuario realiza cambios sobre los campos disponibles del formulario de o sobre la galería de imágenes de la pieza.</p> <p>2. El usuario guarda los cambios de la pieza y envía la pieza modificada al servidor</p> <p>4. El usuario recibe el feedback resultado de la operación en el servidor</p>	<p>3. El servidor recibe los datos de una pieza modificada y tras recuperar los datos originales de la misma aplica un algoritmo de fusión de los datos (basado en reglas configurables desde el software de escritorio: que permitirán según el caso: sobrescribir el contenido completo, añadir a continuación o decidir sobre la información más relevante) para obtener una correcta actualización de la pieza en la base de datos. Posteriormente se envía el resultado de la operación al usuario.</p>

<b>CURSOS ALTERNOS</b>	
En la acción 0	Al igual que en el caso de uso 1.2, toda la operativa puede ser realizada sin disponer de conexión, menos el proceso final de envío de la información (que como su naturaleza indica; requeriría de acceso al servidor en el momento puntual del envío).
<b>Prioridad</b>	Alta
<b>Mejoras aportadas</b>	<ul style="list-style-type: none"> <li>▪ Facilita la incorporación de datos a las piezas desde el código de barras que se puede ver en su etiqueta (que se coloca cuando se extrae del vehículo).</li> <li>▪ Permite la modificación de piezas ya catalogadas.</li> </ul>

<b>Nombre</b>	<b>1.4</b> Recuperar el listado completo de piezas de una orden
<b>Actores</b>	USUARIO
<b>Propósito</b>	Facilitar la recuperación de piezas hacia el dispositivo móvil a través de un código de orden (de revisión, de extracción...), de manera que todas las piezas pertenecientes a esa orden se descarguen directamente en el dispositivo (con todos los datos recopilados en el servidor) con una sola acción.
<p><b>Resumen:</b> El caso de uso se inicia cuando el usuario introduce (ya sea a través del lector de código de barras o introduciendo manualmente) el código de una pieza y consulta dicha referencia. El proceso genera una consulta en el servidor que informa al usuario de la existencia de dicha orden. Devolviendo en caso positivo, el conjunto de piezas asociadas a dicha referencia, de manera que quedan almacenadas en el dispositivo del usuario para facilitar su acceso, consulta o actualización. El caso de uso está formado por un conjunto de operaciones (tanto en el lado del cliente, como del servidor) que se realizan de manera atómica con una sola acción en la aplicación. Esta operación sólo se podrá realizar con acceso al servidor.</p>	
<b>CURSO NORMAL DE EVENTOS</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
0. El usuario pulsa sobre el botón físico del dispositivo "Menú" y tras el	



Casos de uso relativos a CRV Campa:

<b>Nombre</b>	<b>2.1</b> Recuperar el listado de vehículos recibidos en una fecha
<b>Actores</b>	USUARIO
<b>Propósito</b>	Dirigir el trabajo del usuario a través de la recuperación de vehículos de una fecha indicada (por defecto la fecha del día presente) para proporcionar una lista de trabajos pendientes que el operario debe completar y enviar consiguiendo así sincronizar el estado del almacén real con la información de la base de datos
<p><b>Resumen:</b> El caso de uso se inicia cuando el Usuario inicia la aplicación CRV Campa, automáticamente recibe los vehículos pendientes de revisión que han sido introducidos ese día desde el registro de la aplicación de escritorio. Dicha operación puede ser realizada también indicando cualquier día que quiera ser consultado. Esta operación de consulta podrá configurarse también para obtener también los vehículos que ya han sido revisados dentro de la fecha indicada.</p>	
<b>CURSO NORMAL DE EVENTOS</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
<p>0. <i>El usuario inicia la aplicación y pulsa sobre el control de fecha.</i></p> <p>1. El usuario introduce la fecha deseada en el calendario mostrado.</p> <p>3. El usuario recibe en su dispositivo los datos de los vehículos disponibles.</p>	<p>2. El sistema consulta todos los vehículos que han sido registrados en esa fecha y devuelve una lista asociada a la configuración que ha recibido en la petición.</p>

<b>CURSOS ALTERNOS</b>	
En la acción 0	El caso de uso es ejecutado automáticamente en el arranque obteniendo los vehículos registrados en la fecha actual.
En la acción 1	La acción se puede realizar también indicando si se desea recibir todos los vehículos o sólo los que están pendientes de revisión en esa fecha.
<b>Prioridad</b>	Alta
<b>Mejoras aportadas</b>	<ul style="list-style-type: none"> <li>▪ Proporciona al Usuario el acceso a la información de los vehículos recibidos en una fecha sin tener que acercarse a la oficina.</li> <li>▪ Permite consultar la información de los vehículos ya revisados para poder corregir características y actualizar dicha información en el servidor.</li> </ul>

<b>Nombre</b>	<b>2.2</b> Crear un vehículo desde el dispositivo móvil
<b>Actores</b>	USUARIO
<b>Propósito</b>	Proporciona al usuario la capacidad de recoger información de cualquier vehículo, ya sea identificado o no, aunque no disponga de conexión de datos con el servidor, para la posterior clasificación desde la aplicación de escritorio.
<p><b>Resumen:</b> El caso de uso se inicia cuando el Usuario crea desde un formulario el nuevo vehículo introduciendo la información de la que dispone o la que desea aportar. En el vehículo se podrán realizar las funciones de “Identificación”, “Recopilación de fotografías” y “Validación de piezas” (sobre un listado de piezas por defecto, que podrá ser configurado desde la aplicación de escritorio). Tras la incorporación de información, la nueva ficha del vehículo podrá ser enviada al servidor, donde se creará el registro del nuevo vehículo.</p>	
<b>CURSO NORMAL DE EVENTOS</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
0. El usuario pulsa sobre el icono “Añadir vehículo” obteniendo una vista	

<p><i>resumen del nuevo vehículo.</i></p> <p><i>{1. El usuario selecciona en la vista resumen el tipo de acción que realizará [Identificación   Realización de fotografías   Validación de piezas].</i></p> <p>2. El usuario introduce la información que considera oportuna y guarda los cambios.}n</p> <p>3. El usuario marca el vehículo como “TERMINADO”.</p> <p>4. En usuario envía los vehículos terminados.</p> <p>[6. El usuario accede a la aplicación de escritorio y vincula el nuevo registro con una referencia de vehículo existente, lo que forzará a que con posterioridad el nuevo registro pueda ser tratado por igual que otros vehículos con esta referencia, ante acciones masivas.]</p>	<p>5. El servidor recibe una petición y crea un vehículo con los datos introducidos. Posteriormente envía la confirmación del proceso que se mostrará al Usuario en la aplicación móvil.</p>
---	--



<b>CURSOS ALTERNOS</b>	
<p>En la acción 2</p> <p>En la acción 2</p> <p>{Acciones 1 y 2}<sup>n</sup></p> <p>[Acción 6]</p>	<p>Si el Usuario sustituye el código de vehículo autogenerado por un código de vehículo existente, el efecto de esta acción convertirá este caso de uso en un caso de uso de actualización de los datos del vehículo indicado.</p> <p>Consideraciones:</p> <ul style="list-style-type: none"> <li>▪ Si el código de vehículo no existiera se trataría como en la primera situación, es decir, se crearía el vehículo nuevo.</li> </ul> <p>Si entre los datos aportados en este vehículo se encontrara una matrícula ya existente en el sistema, se procedería exactamente que en el caso anterior.</p> <p>La notación {}<sup>n</sup> indica que la secuencia de acciones que contiene puede realizarse un número indeterminado de veces.</p> <p>La notación [] indica que las acciones que contiene pueden estar presentes o no en este caso de uso.</p>
<b>Prioridad</b>	Alta
<b>Mejoras aportadas</b>	<ul style="list-style-type: none"> <li>▪ Proporciona una herramienta al Usuario para introducir vehículos desde el dispositivo móvil aunque carezca de conexión de datos con el servidor y pese a que no tenga ninguna referencia de dicho vehículo.</li> </ul>

Nombre	2.3 Actualizar la información disponible de un vehículo
Actores	USUARIO
Propósito	Facilitar la actualización de vehículos de la base de datos a través del código de vehículo o la matrícula.
<p><b>Resumen:</b> El caso de uso se inicia cuando el Cliente accede a alguno de los vehículos que tiene almacenados localmente en su dispositivo (a través de la descarga de vehículos de una fecha o creados manualmente). Éste podrá realizar modificaciones sobre los mismos y enviarlos al servidor, haciendo efectivos los respectivos cambios.</p>	
<b>CURSO NORMAL DE EVENTOS</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
<p>0. El usuario pulsa sobre uno de los vehículos disponibles en su dispositivo.</p> <p>{1. El usuario selecciona en la vista resumen el tipo de acción que realizará [Identificación   Realización de fotografías   Validación de piezas].</p> <p>2. El usuario introduce la información que considera oportuna (pero <b>exigiéndose disponer de código de vehículo o matrícula en el proceso de "Identificación"</b>), y guarda los cambios.}^n</p> <p>3. El usuario marca el vehículo como "TERMINADO".</p> <p>4. En usuario envía los vehículos terminados.</p>	<p>5. El servidor recibe la petición y actualiza los datos del vehículo con los datos introducidos. Posteriormente envía la confirmación del proceso que se mostrará al Usuario en la aplicación móvil.</p>

<b>CURSOS ALTERNOS</b>	
{Acciones 1 y 2} <sup>n</sup>	La notación {} <sup>n</sup> indica que la secuencia de acciones que contiene puede realizarse un número indeterminado de veces.
<b>Prioridad</b>	Alta
<b>Mejoras aportadas</b>	<ul style="list-style-type: none"> <li>▪ Permite actualizar cualquier vehículo que disponga de código de vehículo o matrícula en el sistema.</li> </ul>

<b>Nombre</b>	<b>2.4</b> Generación de órdenes de despiece de un vehículo
<b>Actores</b>	USUARIO
<b>Propósito</b>	Facilitar la generación de órdenes de despiece desde el dispositivo móvil.
<p><b>Resumen:</b> El caso de uso se inicia cuando el Usuario tras la creación o descarga de un vehículo accede a la herramienta de validación de piezas y selecciona del listado de piezas asociado al vehículo aquellas que considera de interés para su extracción. Esta orden podrá ser recuperada desde la aplicación móvil CRV Almacén en su caso de uso 1.4.</p>	
<b>CURSO NORMAL DE EVENTOS</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
<p>0. El usuario pulsa sobre uno de los vehículos disponibles en su dispositivo.</p> <p>1. El usuario selecciona de la vista resumen el tipo de acción "Validación de piezas".</p> <p>2. El usuario estudia el estado del vehículo y de sus piezas y selecciona en el formulario aquellas que considera de interés para su extracción.</p> <p>3. El usuario marca el vehículo como "ORDEN DE DESPIECE".</p> <p>4. El usuario envía el vehículo.</p>	<p>5. El servidor recibe la petición y genera la orden de despiece con las piezas seleccionadas de este vehículo. Posteriormente envía la confirmación del proceso que se mostrará al Usuario en la aplicación móvil.</p>

<b>CURSOS ALTERNOS</b>	
	<ul style="list-style-type: none"> <li>▪ Solo existe un procedimiento.</li> </ul>
<b>Prioridad</b>	Alta
<b>Mejoras aportadas</b>	<ul style="list-style-type: none"> <li>▪ Hace independiente de la aplicación de escritorio la generación de órdenes.</li> <li>▪ Transforma por completo la dinámica actual de funcionamiento (basada en la impresión física de formularios en papel con los listados de piezas del vehículo, donde el Usuario marca las piezas), evitando un proceso intermedio en el que un segundo Usuario digitalizaba la información recogida por el primero.</li> </ul>

Restricciones no funcionales que afectan a los casos de uso:

- Por razones de privacidad, se ha decidido que no sea posible el acceso a los servidores a través de conexiones de datos de operadores móviles. Por lo que únicamente se podrá conectar con los servidores dentro del alcance de los repetidores Wifi de la empresa.

### 4.3 GUI (Interfaz Gráfica de Usuario)

La interfaz gráfica de una aplicación móvil a menudo encauza el éxito o el fracaso de una propuesta de software, por lo que se ha dedicado a este apartado una parte importante del esfuerzo de desarrollo de la aplicación.

Para el desarrollo de interfaz de la aplicación móvil se ha obviado cualquier aspecto gráfico de la aplicación de escritorio, es decir, se ha evitado ceñirse a portar una aplicación de una plataforma a otra, con lo que se ha propiciado un rediseño total de la interacción del usuario, todo ello con la finalidad de propiciar una interacción natural y sencilla, practicable sin apenas recibir ninguna formación para el uso.

En la herramienta propuesta se pueden diferenciar dos aplicaciones autónomas interrelacionadas, a las que llamaremos CRV Almacén y CRV Campa, ambas diseñadas con un aspecto muy similar “a modo de Suite”, que comparten modo de uso, pero que a su vez permiten el acceso a funcionalidades diversas.

Para la correcta visualización de los contenidos, se ha diseñado formularios, listados detallados, galerías de imágenes y matrices de contenidos, siempre con la pretensión de utilizar en cada situación los controles que permiten maximizar la eficacia de la función encomendada. A continuación se mostrarán algunas capturas de pantalla de sendas aplicaciones y se definirán algunas de sus características.

#### CRV Almacén



Ilustración 29: Captura de pantalla de la APP CRV Almacén - Listado de piezas

A la izquierda podemos observar la pantalla inicial de CRV Almacén.

El control principal de esta vista es la implementación de una lista de piezas seleccionables que permite el acceso individual a cada una de las piezas que van a ser actualizadas en el servidor.

Además se muestran en esta misma interfaz un contador de piezas que mantiene informado al usuario de esta característica y dos botones inferiores para gestionar las dos actividades más frecuentes que se pueden realizar desde esta perspectiva: “Añadir” piezas nuevas o “Enviar” algunas de las que ya hemos documentado

El principio fundamental de ofrecer una interfaz limpia y sencilla, en la que sólo se dispone de controles para la realización de las acciones fundamentales, no puede reducir el conjunto de funcionalidades de la herramienta, motivo por el que aquellas acciones menos frecuentes son incorporadas también, pero esta vez a través de submenús accesibles a través de la tecla “Menú” física del dispositivo.

Las citadas características se pueden observar en la imagen de la derecha y son: “Vaciar la lista de piezas disponibles”, “Descargar piezas desde orden” y “Configuración”



Ilustración 30: Captura de pantalla de la APP CRV Almacén - Descargar piezas desde orden



Ilustración 31: Captura de pantalla de la APP CRV Almacén - Selección

Por otro lado también se puesto especial interés en aprovechar los eventos más comunes de interacción como puedan ser pulsaciones simples, pulsaciones largas, deslizamientos... eventos a los que el usuario está acostumbrado y le permiten acceder a funciones extra que no están visibles a primera vista.

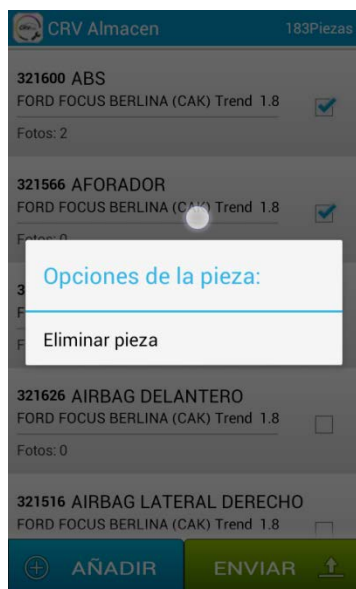
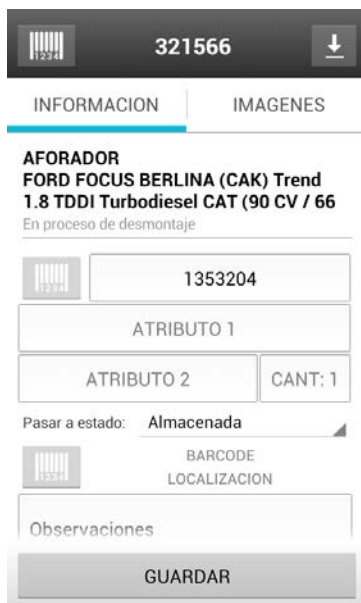


Ilustración 32: Captura de pantalla de la APP CRV Almacén - Pulsación prolongada

Se ha hecho uso de animaciones ligeras en ciertos casos, que permiten afianzar a través de un feedback positivo las acciones del usuario.

A la izquierda se puede observar un menú descubierto por el efecto producido de una pulsación larga sobre una pieza concreta.



**Ilustración 33: Captura de pantalla de la APP CRV Almacén - Menú Información de la pieza**

El formulario principal de recogida de datos muestra una disposición en pestañas que diferencia claramente los dos ámbitos permitidos: La recogida de “Información” textual y la recogida de “Imágenes” de la pieza.

En la pestaña de información podemos observar un conjunto de campos textuales editables sin ningún título informativo que muestre la finalidad del mismo. La descripción de cada campo se muestra dentro del mismo control, y únicamente cuando el mismo está vacío. Esta descripción desaparece en el momento en que el usuario teclea cualquier carácter.

Esta decisión de diseño basa su existencia en la optimización del espacio disponible en pantalla, generalmente reducido, que

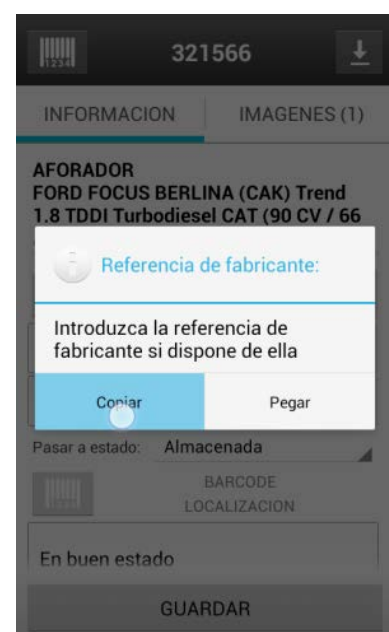
favorece la exposición del mayor número de contenidos para minimizar el número de interacciones del usuario con la pantalla para realizar su trabajo.

Aunque en una primera instancia la decisión pudiera comprometer el nivel de dificultad para el usuario, hay que considerar que en un entorno de producción donde el usuario explota la herramienta a diario, la familiarización con cada campo está garantizada, premiando así la disposición de los contenidos explícita, para garantizar una mayor agilidad en la introducción de los datos.

Pese a al esfuerzo dedicado para asegurar la cabida de todos los controles dentro del formulario en un solo vistazo, la gran diferencia de tamaños de pantalla de los dispositivos objetivo donde se implantará esta aplicación, se ha caracterizado a dicho formulario con la posibilidad de hacer “scroll”, es decir, permite mostrar únicamente la parte del formulario que cabe en pantalla, permitiendo acceder al contenido restante deslizando el formulario de abajo hacia arriba.

Una de las funcionalidades estrella en nuestro diseño, fue el hecho de rediseñar las funciones conocidas en todas las plataformas como “Copiar/Pegar”, que en este caso permiten trasladar información entre distintos campos de formulario o incluso entre distintas piezas.

Aunque el sistema operativo ya las incorporaba, el hecho de redefinirlas ha asegurado el tipo de interacción independientemente de la versión del sistema operativo.



**Ilustración 34: Captura de pantalla de la APP CRV Almacén - Copiar/Pegar**



La pestaña de “Imágenes” permite el acceso a los controles de adición o eliminación de fotografías, así como al control galería que permite el acceso al conjunto de imágenes recopiladas sobre la pieza que se está procesando.

Para la navegación entre las distintas imágenes se ha optado por una interacción del tipo desplazamiento horizontal (a izquierdas o derechas según se pretenda acceder a la imagen anterior o posterior) sobre el control galería, lo que permite disponer en cada momento de una miniatura de la imagen que puede ser ampliada o rotada a través de la interacción de usuario consistente en una pulsación prolongada fija sobre la imagen seleccionada en ese momento.

A continuación se muestran capturas de pantalla de las situaciones comentadas

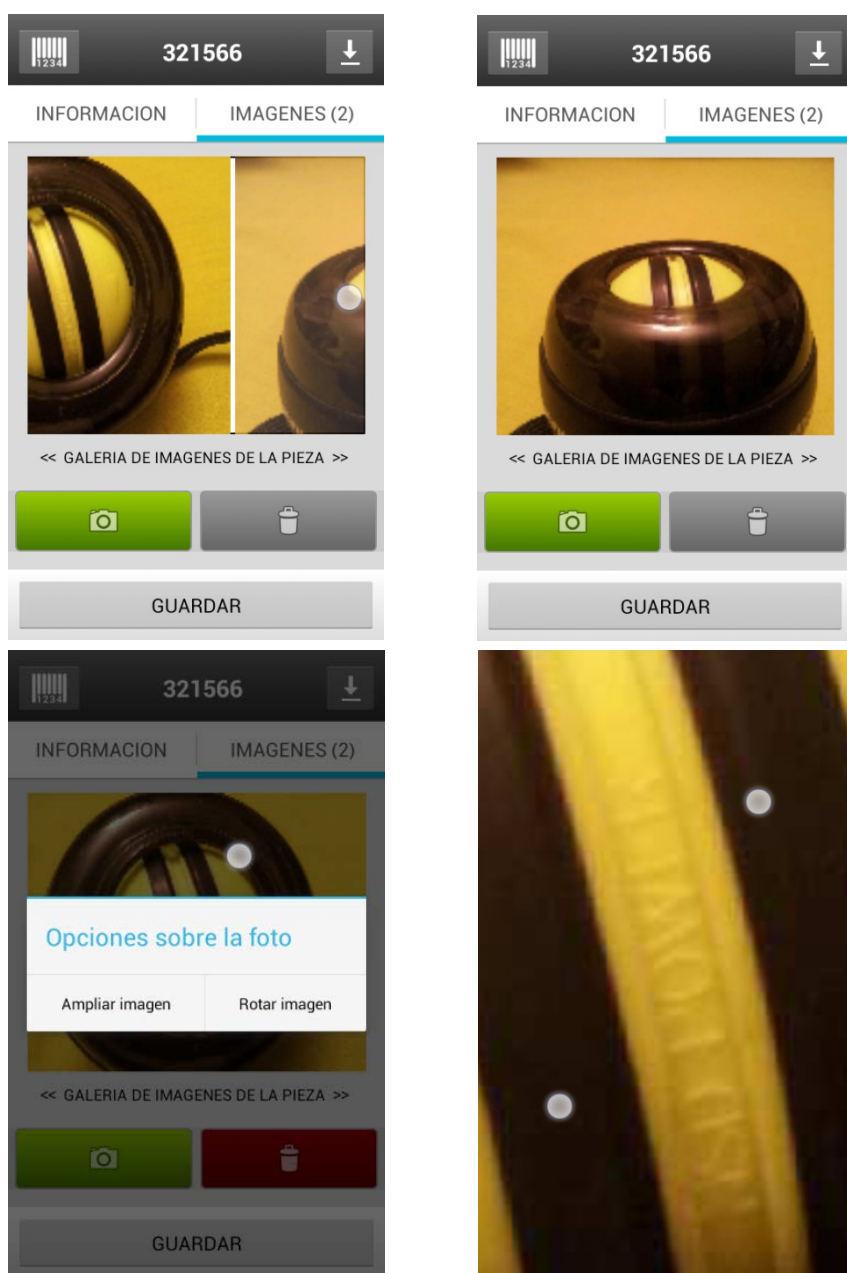


Ilustración 35: Captura de pantalla de la APP CRV Almacén – Capturas de la galería de Imágenes

Una vez definidas de manera independiente las vistas que caracterizan la aplicación CRV Almacén, pasaremos a señalar las interacciones que pueden suceder entre las mismas, según la finalidad pretendida, se puede consultar en el cuadro de la siguiente página.

En dicho diagrama se entrecruzarán las distintas funcionalidades de CRV Almacén (horizontal) con las distintas fases de las mismas (vertical), facilitando al usuario la identificación de las fases comunes entre distintas operativas y simplificando así el aprendizaje y la familiarización de los procedimientos.

Ilustración 36: Diagramas de interacción de la interfaz de CRV Almacén

CRV Almacén

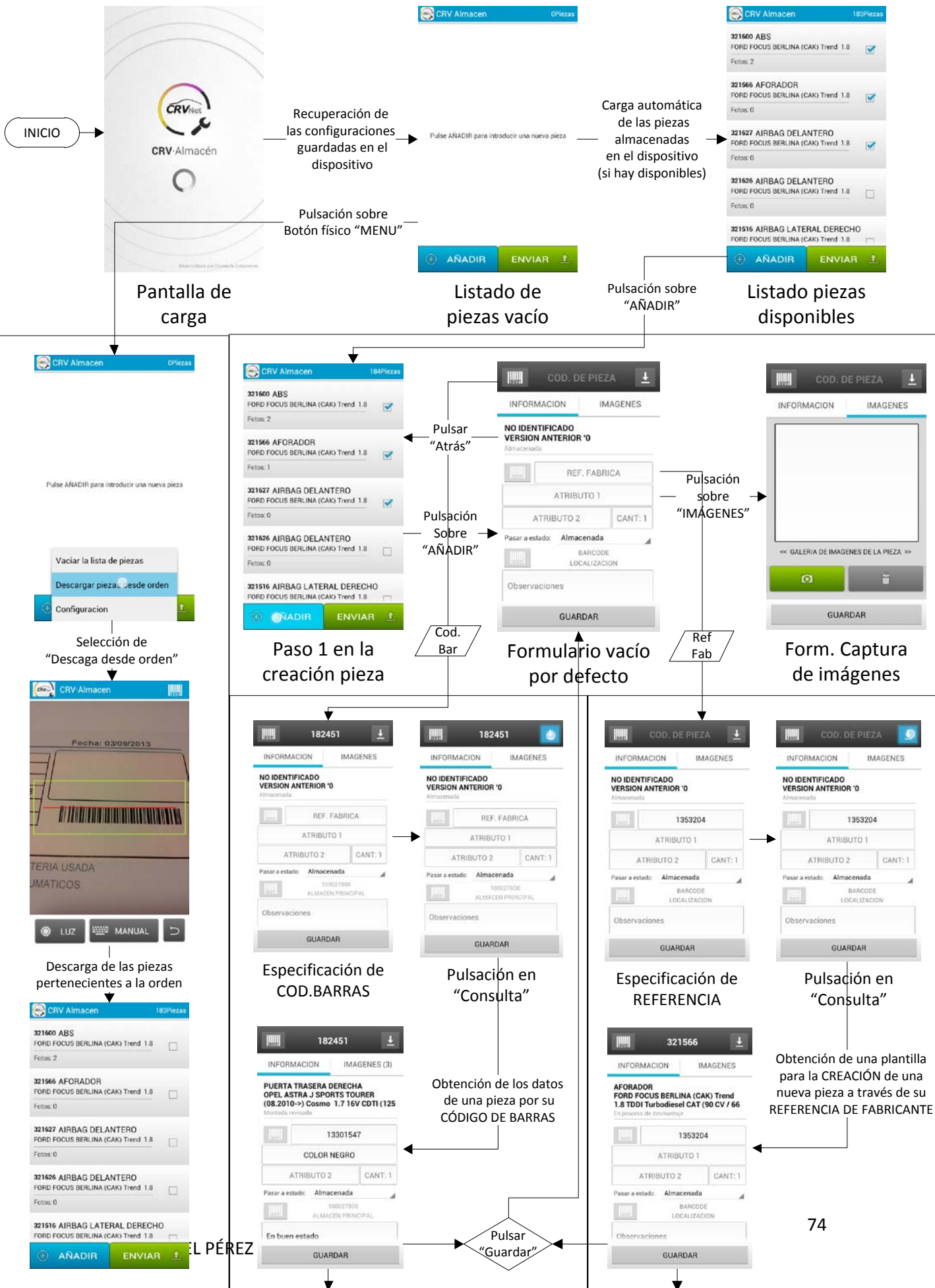
Inicialización de la aplicación

Recepción de piezas múltiple (desde orden)

Actualización de datos de piezas desde <Código de barras>

Creación de piezas desde <Referencia de fabricante>

Fase de Recuperación/Creación de piezas



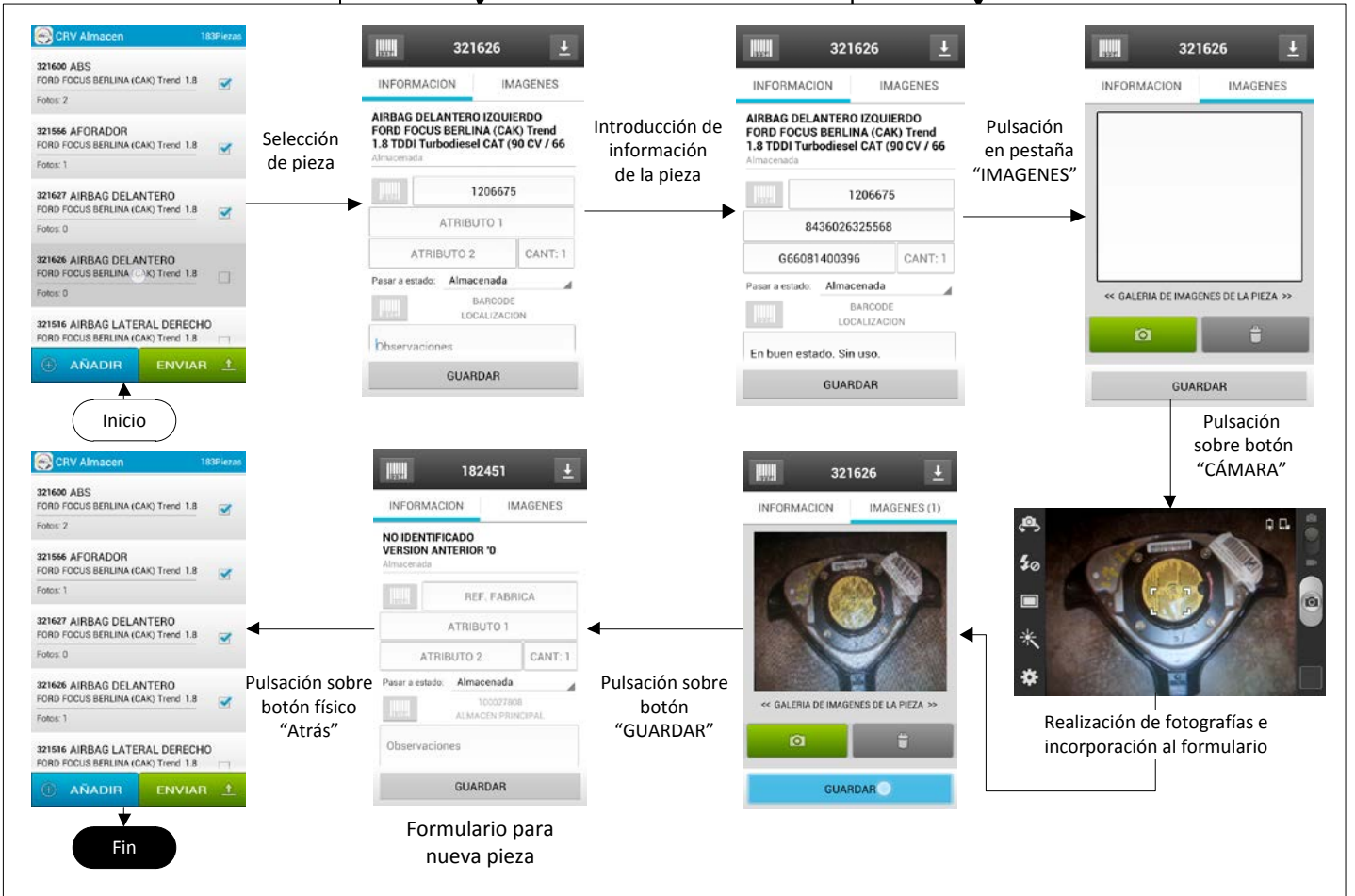
# CRV Almacén

Recepción de piezas múltiple (desde orden)

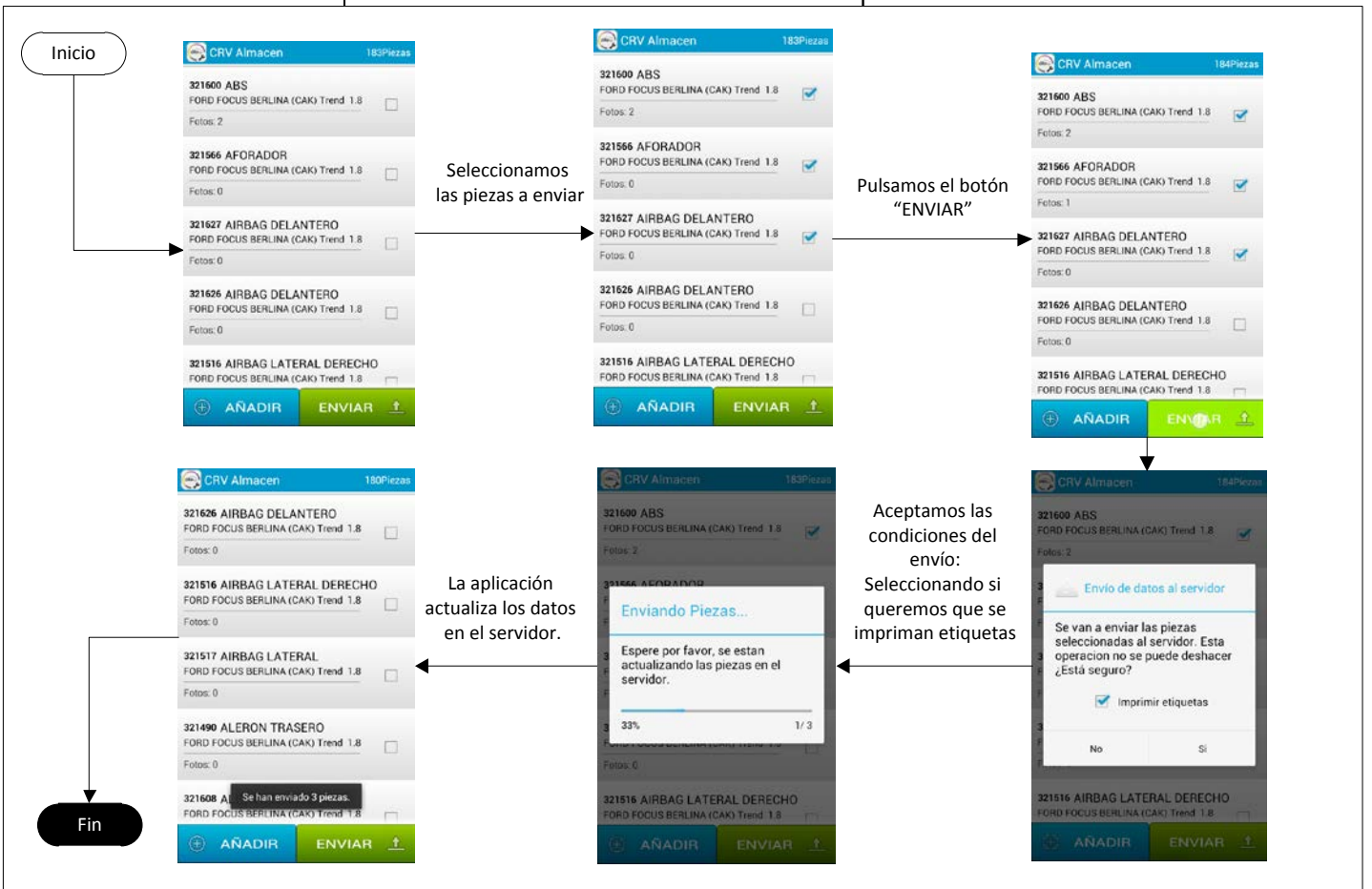
Envío/recepción de piezas individual <Código de barras>

Envío/recepción de piezas individual <Referencia fabricante>

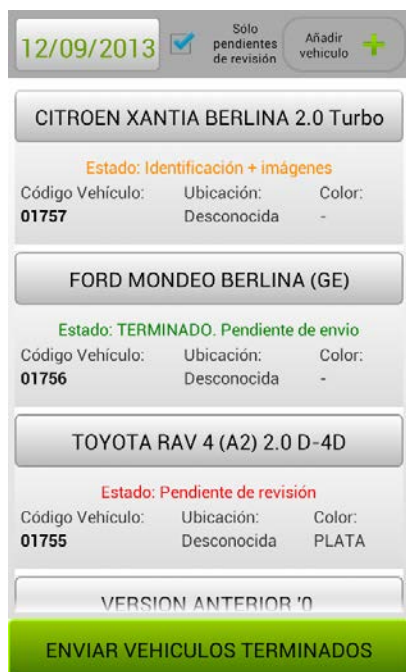
Cumplimentación de datos de las piezas



Envío de datos al servidor



## CRV Campa



A la izquierda podemos observar la pantalla inicial de CRV Campa.

El control principal de esta vista está formado por la lista de vehículos a la que podemos acceder sin conexión, que permite acceder al detalle de cada uno de ellos. Cada elemento de la lista proporciona información resumida del estado y de características de un vehículo distinto.

Además se muestran en esta misma interfaz herramientas para la adición de vehículos como un botón con una fecha, que permite descargar los vehículos recibidos de un día concreto y un botón para introducir vehículos manualmente.

Ilustración 37: Captura de pantalla de la APP CRV Campa - Pantalla principal



Ilustración 39: Captura de pantalla de la APP CRV Campa - Pantalla principal (Horizontal)

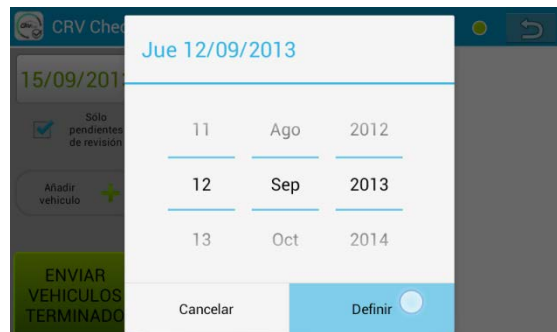


Ilustración 38: Captura de pantalla de la APP CRV Campa - Descarga de vehículos por fecha

Desde esta misma vista se pueden enviar los vehículos terminados a través del control correspondiente.

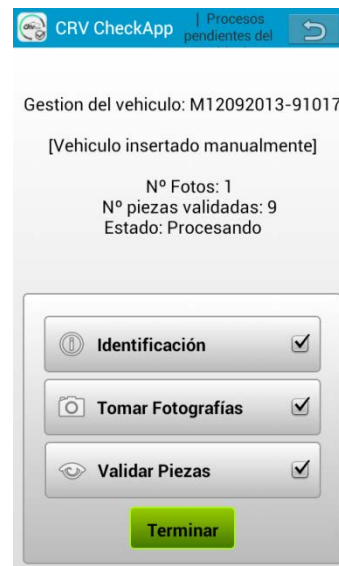
Como se puede observar en las ilustraciones superiores, la APP CRV Campa redefine las vistas dependiendo de la orientación física del dispositivo, lo que permite aprovechar las características de tabletas y teléfonos de grandes pantallas.



## INTEGRACIÓN DE TECNOLOGÍAS MÓVILES EN EL NEGOCIO DE LOS DESGUACES DE VEHÍCULOS



**Ilustración 41: Captura de pantalla de la APP CRV Campa – Vista resumen (previo)**



**Ilustración 40: Captura de pantalla de la APP CRV Campa – Vista resumen (Operaciones finalizadas)**

La pulsación sobre un vehículo de la lista de la pantalla principal genera la vista del detalle del ítem seleccionado.

En dicha vista se muestra un resumen del trabajo realizado sobre este vehículo y se posibilita a través de controles de tipo botón el acceso a las funciones de “Identificación”, “Toma de fotografías” y “Validar piezas”, los cuales mantienen información de estado del paso del usuario por los mismos.



**Ilustración 42: Captura de pantalla de la APP CRV Campa - Vista de identificación**



**Ilustración 43: Captura de pantalla de la APP CRV Campa - Vista de identificación (scrolled)**

## INTEGRACIÓN DE TECNOLOGÍAS MÓVILES EN EL NEGOCIO DE LOS DESGUACES DE VEHÍCULOS

En la ilustración anterior se pueden observar varias capturas de la vista de identificación del vehículo, la cual permite la consulta e introducción de información textual en diversos campos editables, que se anidan dentro de un panel capaz de mostrar un subescenario de la vista principal a través de interacciones con la interfaz de tipo arrastre vertical.

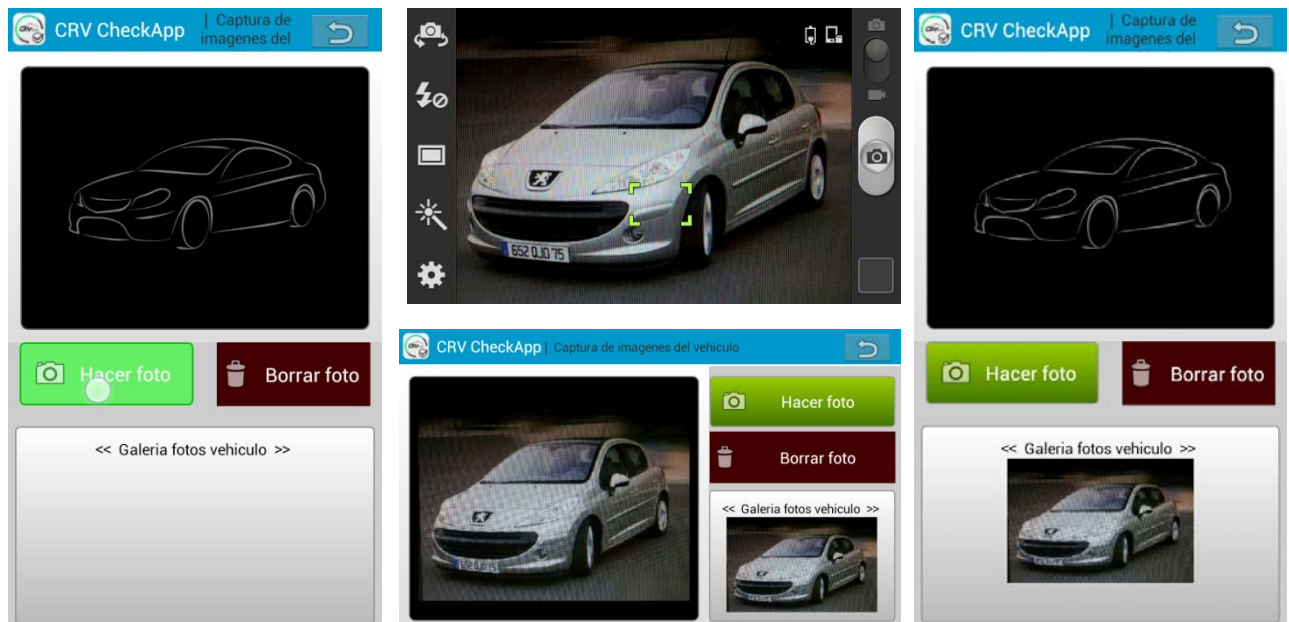


Ilustración 44: Captura de pantalla de la APP CRV Campa - Captura de imágenes

Las anteriores capturas definen el aspecto de la interfaz vertical y horizontal de la vista de realización de fotografías. En ellas encontramos 3 zonas diferenciadas; un visor de imágenes, donde podemos ver la imagen con su calidad original, botonera donde encontramos las acciones principales de la vista y una galería donde podemos realizar movimientos de arrastre horizontal para desplazar entre las distintas miniaturas de las fotos realizadas a este vehículo.

Por último, a continuación incluimos varias capturas de las vistas de revisión de piezas, esta parte de la herramienta consta de elementos tipo matriz cuya selección implica; bien la selección de familia (en el primer nivel) o el cambio de estado de pieza “Montada en vehículo” a pieza “Revisada” preparada para su extracción (tras la selección de familia).

La interfaz reacciona marcando con un tono verdoso aquellas familias que tienen alguna de sus piezas marcada como “Revisada” para facilitar al mecánico su labor



Ilustración 45: Captura de pantalla de la APP CRV Campa - Validación de piezas

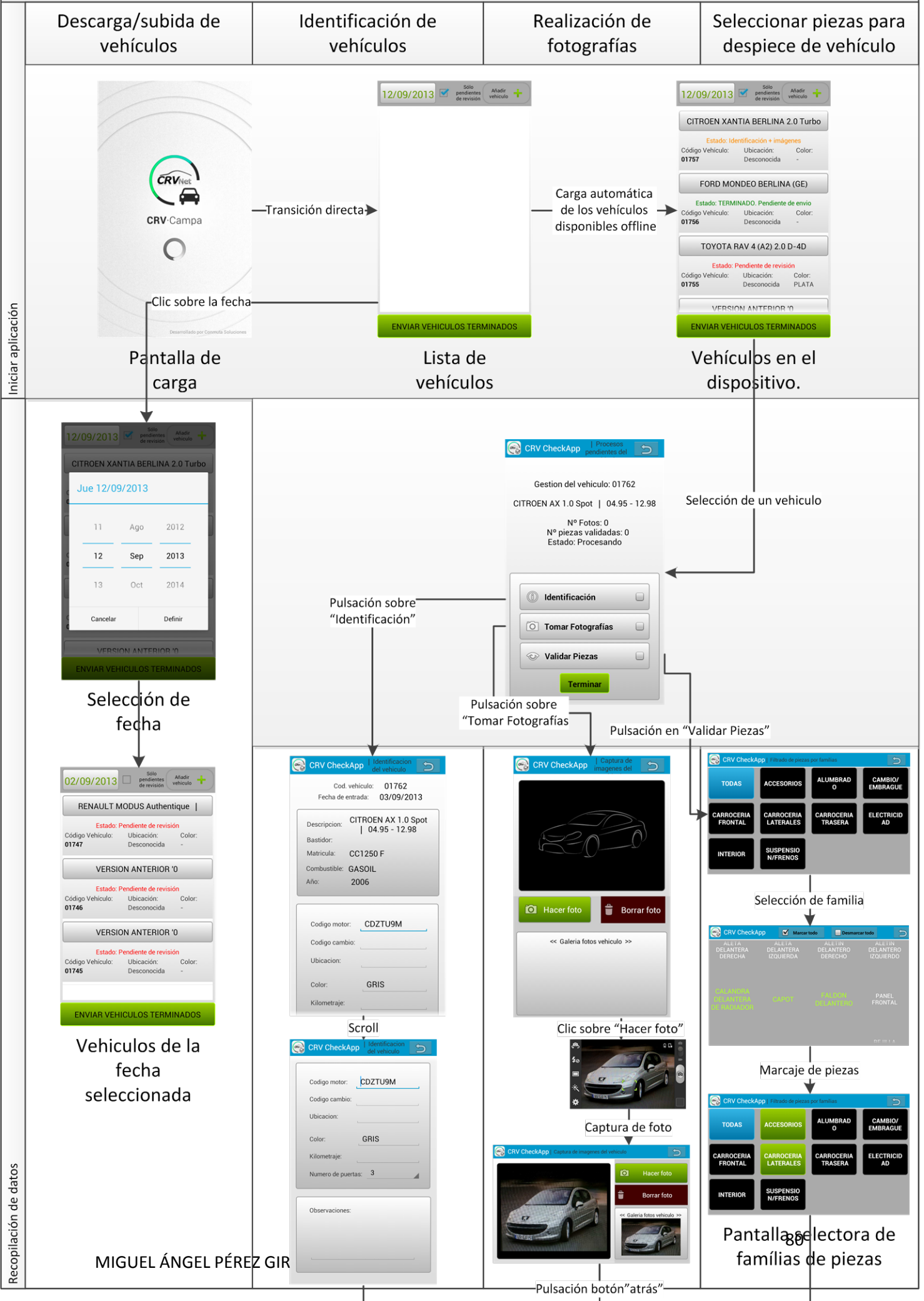
Una vez definidas de manera independiente las distintas vistas de que caracterizan la aplicación CRV Campa, al igual que hicimos en la aplicación CRV Almacén, pasaremos a señalar las principales interacciones que pueden suceder entre las distintas vistas, según la finalidad pretendida, se puede consultar en el cuadro de la siguiente página.

En dicho diagrama se entrecruzarán las distintas funcionalidades de CRV Campa (horizontal) con las distintas fases de las mismas (vertical), facilitando al usuario la identificación de las fases comunes entre distintas operativas y simplificando así el aprendizaje y la familiarización de los procedimientos.



**Ilustración 46: Diagramas de interacción de la interfaz de CRV Campa**

**CRV Campa**



# CRV Campa

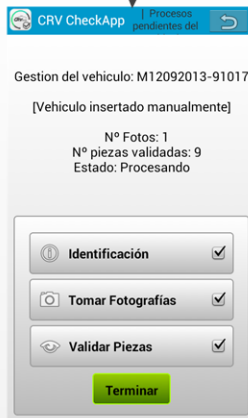
Descarga/subida de vehículos

Identificación de vehículos

Realización de fotografías

Seleccionar piezas para despiece de vehículo

Continuación diagrama anterior



Vista detallada resum. vehículo

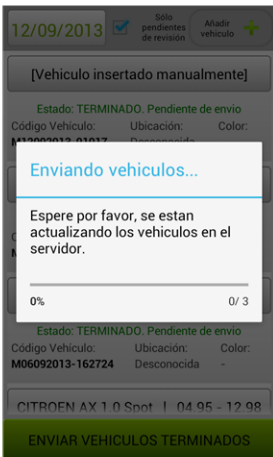


Listado de vehículos preparados para envío

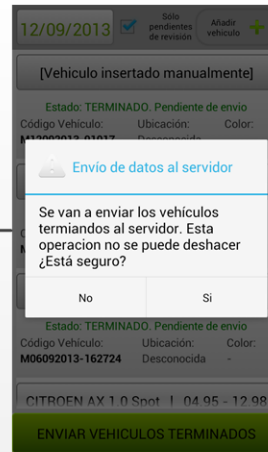
Finalizar la recogida de datos del vehículo

Pulsación sobre "Terminar"

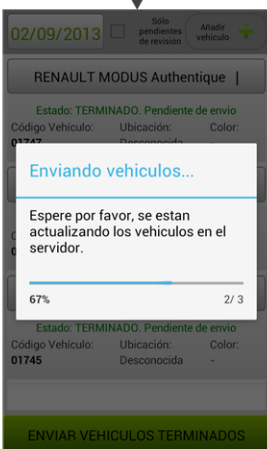
Pulsación sobre "ENVIAR VEHICULOS TERMINADOS"



Envío de todos los vehículos terminados

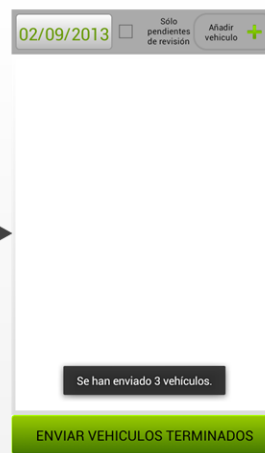


Aceptación



Medición del porcentaje completado

Proceso completado elimina del dispositivo los vehículos enviados satisfactoriamente



Lista de vehículos pendientes

Actualización de información en el servidor

### 4.4 Modelo de datos.

La finalidad de todo sistema es realizar un trabajo concreto. Dicho trabajo requiere de un tiempo de proceso y a su fin, obtiene unos resultados que deberán ser almacenados para que su información sea persistente en el tiempo.

No bastando con el simple almacenamiento de los datos, todo sistema necesita crear relaciones entre los datos, para con ello representar entidades y sus respectivos atributos, capaces de modelar objetos tangibles o intangibles de la vida real. Dichas entidades también podrán ser relacionadas entre sí, formando complejas estructuras de datos que convierten datos en información.

Para el entorno que nos concierne; el desarrollo de la aplicación móvil, se ha creado un modelo reducido del modelo real, para aislar las características que conciernen a este desarrollo del de la aplicación de escritorio ya implantada.

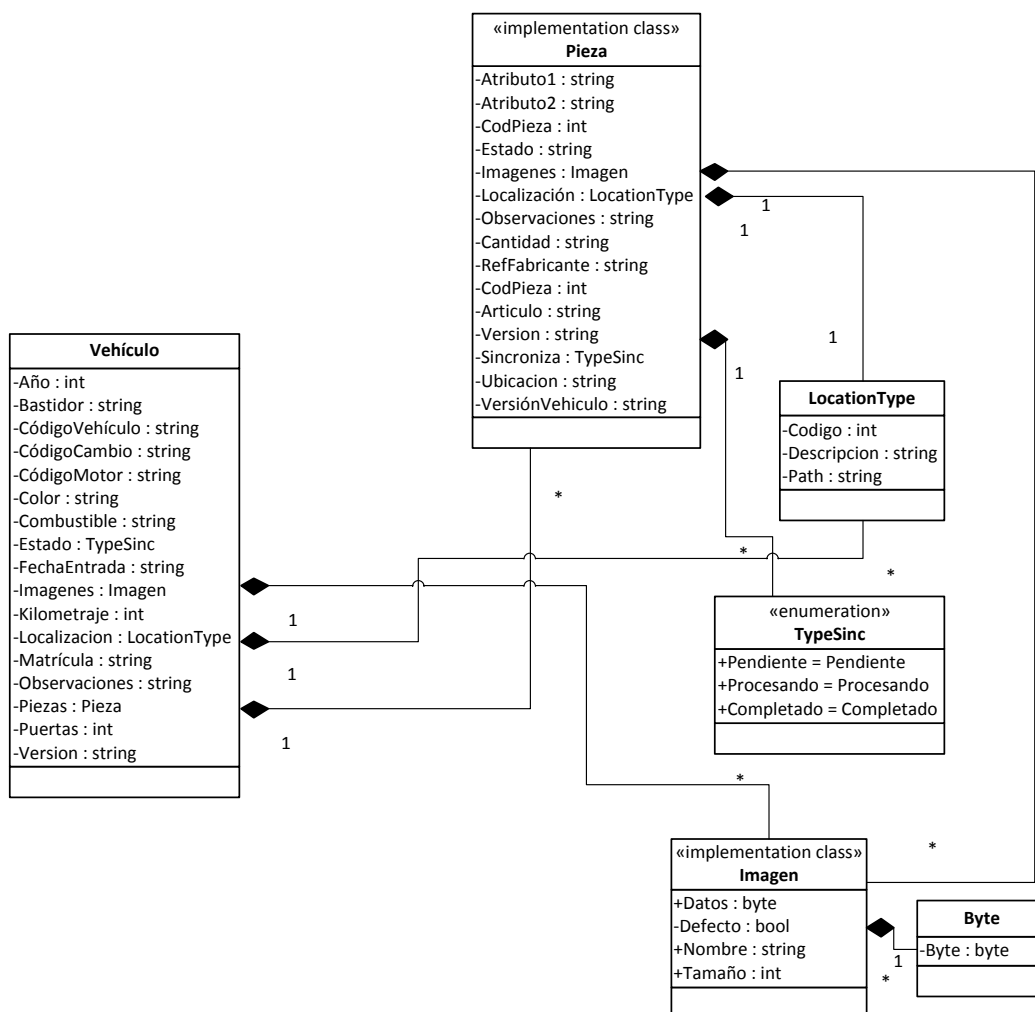


Ilustración 47: Diagrama de entidades y relaciones

La persistencia de la información de esta aplicación se ha logrado a través de procesos de serialización de las entidades descritas en la memoria de almacenamiento del dispositivo móvil. Dicho proceso en nuestro caso se ha basado en la codificación de buffers y objetos en memoria a archivos de texto XML.

Para el almacenamiento de configuraciones locales de la aplicación (como dirección del servidor y otras preferencias de usuario), se ha hecho del API "*Android.Preference*", y en concreto de la clase "*SharedPreferences*" que permiten que todos los ajustes realizados para una preferencia determinada se guarden automáticamente en la instancia de la aplicación.

Por otro lado, el mayor beneficiario del uso de nuestro modelo de datos ha sido el protocolo de comunicación utilizado SOAP, que se ha utilizado para transmitir datos complejos como instancias de objetos completos. ¿Por qué no hemos utilizado REST para las comunicaciones en nuestra aplicación? Lo explicaremos en el punto 5.1.3 cuando hablemos de las tecnologías utilizadas en el proyecto.

## 5 Implementación y desarrollo

El primer enigma a resolver en la fase de análisis del contexto en el que se va a desarrollar este proyecto, consta en discernir entre los distintos entornos de desarrollo que podemos encontrar en el mercado, aquél en que favorezca en mayor medida la implementación y dé mayores garantías en su uso.

Con el fin de hacer uso del equipamiento del que ya dispone la empresa, vamos a realizar el desarrollo sobre un equipo con un procesador Dual Core 2.8 GHz, 4GB RAM y Windows Vista, donde instalaremos nuestro centro trabajo y lo acondicionaremos para cubrir con todas las necesidades que requiera el proyecto; como aplicaciones de diseño gráfico, editores de texto, herramientas de descompresión de archivos...

Como bien indicamos en la introducción de este PFC, la aplicación pretendida debería funcionar en dispositivos móviles y teniendo en cuenta el alcance de mercado de las principales plataformas, así como sus perspectivas de crecimiento, el desarrollo inicial se realizará en Android, pudiendo extenderse más adelante hacia IOS, Windows Phone y BlackBerry.



**Worldwide Smartphone Platforms Market Share, 2013 Q2 (Unit Share)**

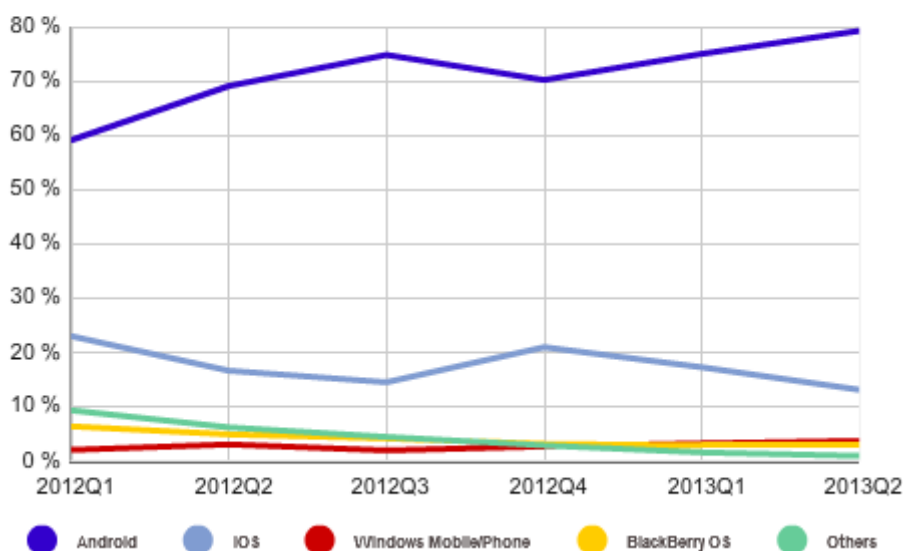


Ilustración 48: Gráfico comparativo del reparto de mercado de sistemas operativos móviles. Fuente: IDC

( IDC Worldwide Mobile Phone Tracker, 2013)

## 5.1 Herramientas utilizadas.

Como planteábamos en la introducción de éste punto; una de las decisiones más importantes (previas al desarrollo), consiste en la adecuada elección del entorno de desarrollo integrado (IDE) que permita la correcta implementación de las aplicaciones encomendadas, en perspectiva, no sólo a las especificaciones actuales, sino en vistas a afrontar nuevos retos en el futuro.

### 5.1.2 El entorno de desarrollo

A continuación, estudiaremos las alternativas más comunes de las que disponemos en la actualidad para el desarrollo de aplicaciones nativas en Android. Entre ellas podemos encontrar Eclipse, Xamarin y Android Studio.

#### 5.1.2.1 Eclipse

Eclipse es un IDE de código abierto y multiplataforma programado principalmente en JAVA y ampliamente utilizado para el desarrollo de aplicaciones Android y otros productos.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto. (Wikipedia, 2013)

La definición que da el proyecto Eclipse acerca de su software es: "*una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular*", lo que pone de manifiesto la intención globalizadora que ha pretendido el equipo de trabajo.

El entorno de desarrollo integrado de Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad al frente de la plataforma de cliente enriquecido, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente a permitirle a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python, permite a Eclipse trabajar con lenguajes para procesado de texto como LaTeX, aplicaciones en red como Telnet y Sistema de gestión de base de datos.

Eclipse hace uso de widgets; que son pequeños programas a modo de herramientas que permiten dar fácil acceso a funciones frecuentemente usadas y proveer de información visual al usuario.

Los widgets de Eclipse están implementados por una herramienta de widget para Java llamada Standard Widget Toolkit, a diferencia de la mayoría de las aplicaciones Java, que usan las opciones estándar Abstract Window Toolkit (AWT) o Swing.

Eclipse dispone de un Editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes para creación de proyectos, clases, pruebas, etc.

En cuanto al desarrollo de aplicaciones Android se refiere, Eclipse es quizás la herramienta más extendida y más probada. El pack mínimo para empezar a trabajar se consigue a través de la instalación de Java SE (actualmente en su versión 7 Update 40), más el SDK de Android y la instalación del plug-in de Eclipse para Android (<https://dl-ssl.google.com/android/eclipse/>)

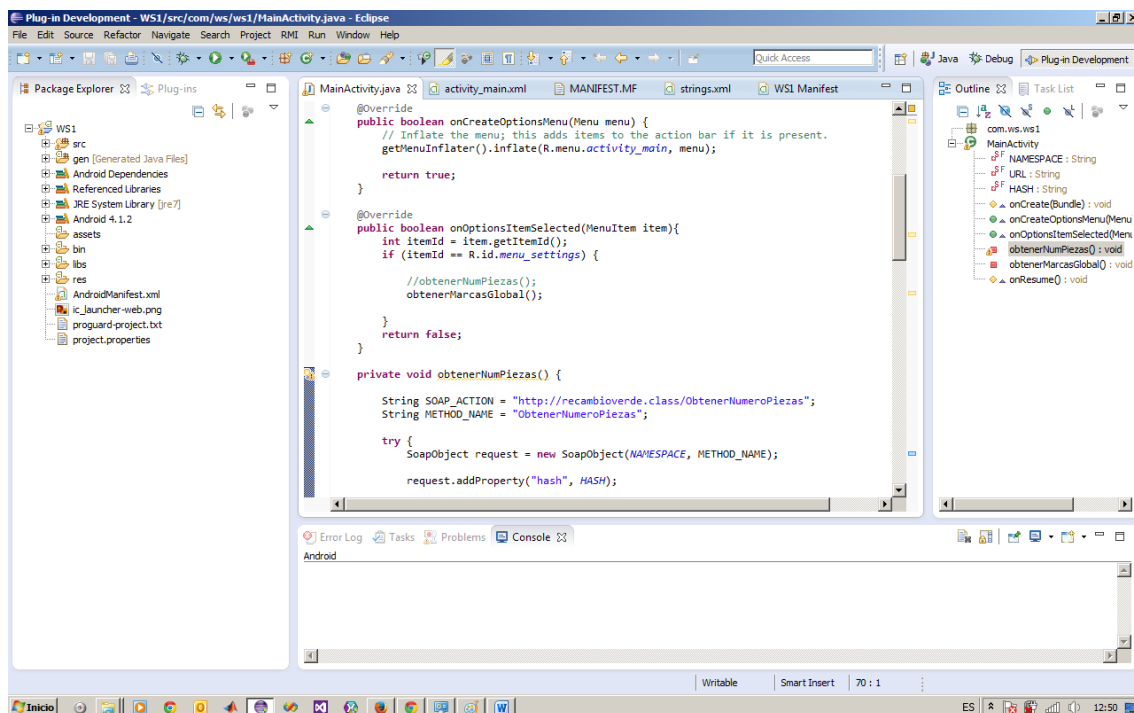


Ilustración 49: Captura de pantalla de Eclipse

## 5.1.2.2 Xamarin Studio

Xamarin es una compañía establecida en mayo de 2011 por los ingenieros que crearon Mono, una implementación libre de la plataforma de desarrollo .NET para dispositivos Android, iOS y GNU/Linux. En junio de 2000, Microsoft anunció por primera vez su .NET Framework. Miguel de Icaza, de Ximian, comenzó a investigar si una versión para Linux era factible.

Posteriormente, el proyecto Mono fue lanzado el 19 de junio de 2001. Ximian fue entonces adquirida por Novell el 4 de agosto de 2003. (Wikipedia, 2013)

Después de la subsecuente adquisición de Novell por parte de Attachmate en abril de 2011, esta última empresa anunció el despido de cientos de trabajadores de Novell, incluyendo a los desarrolladores de Mono, poniendo así el futuro del proyecto en problemas. Sin embargo, el 16 de mayo, Miguel de Icaza anunció en su blog que Mono sería desarrollado y soportado por Xamarin, empresa que planeaba lanzar una serie de productos para dispositivos móviles.

Hoy la empresa de De Icaza, comandada por Nat Friedman, anuncia Xamarin 2.0, que incluye un plug-in que permite que los desarrolladores de Android e iOS puedan escribir aplicaciones nativas en C# usando Visual Studio de Microsoft. También se introduce el IDE de Xamarin Studio así como una colección de componentes. En esta tienda de componentes, la cual viene integrada a Xamarin Studio y Visual Studio, es un catálogo de controles para la interfaz gráfica, gráficas, charts, temas y servicios en la nube. Algunos componentes no tienen costo, otros sí, y van de los rangos de precios entre 50 y 400 dólares.

El entorno de desarrollo, Xamarin Studio, puede ser usado para crear aplicaciones cruzadas con C#, esto quiere decir que se pueden escribir en una plataforma y generar código para otra. Xamarin Studio está fuertemente integrado con iOS y el SDK de Android por lo que se pueden crear, probar y depurar aplicaciones en emuladores así como dispositivos reales.

Xamarin Studio es también multiplataforma (Windows y Mac), pero al contrario que otras alternativas, pese a tener una alternativa gratuita limitada por un tamaño de código compilado bajo (32KB), obtener una versión capacitada para afrontar un proyecto real tiene un coste, comenzando por 299 dólares por plataforma/año por persona y 1899 dólares por la plataforma/año para las grandes organizaciones.



Ilustración 50: Viñeta del conjunto de aplicaciones de Xamarin



Algunas de las características más interesantes de Xamarin Studio que promete el mismo proveedor son:

- Capacidad para reutilizar hasta el 90% del código utilizado en traducciones de aplicaciones Android a iOS y a la inversa.
- Facilidad de instalación, proporcionando un solo instalador que se aprovisiona automáticamente de todas las características necesarias: el IDE, Android SDK, GTK#, Java...
- Capacidad para incorporar las librerías y Frameworks .NET y de terceros.
- Acceso completo al API de Android estándar.
- Resaltado de sintaxis y autocompletado de código.
- Integración automática con el IDE de Visual Studio de Microsoft.
- Documentación muy extensa de todas las características de Xamarin; como acceso a los métodos, y modos de uso. También completos tutoriales paso a paso de integración de componentes y ejemplos varios.
- Generación de código nativo, soporte para LINQ, trabajo con XML.
- Comunidad de usuarios (Disponible en [forums.xamarin.com](http://forums.xamarin.com))

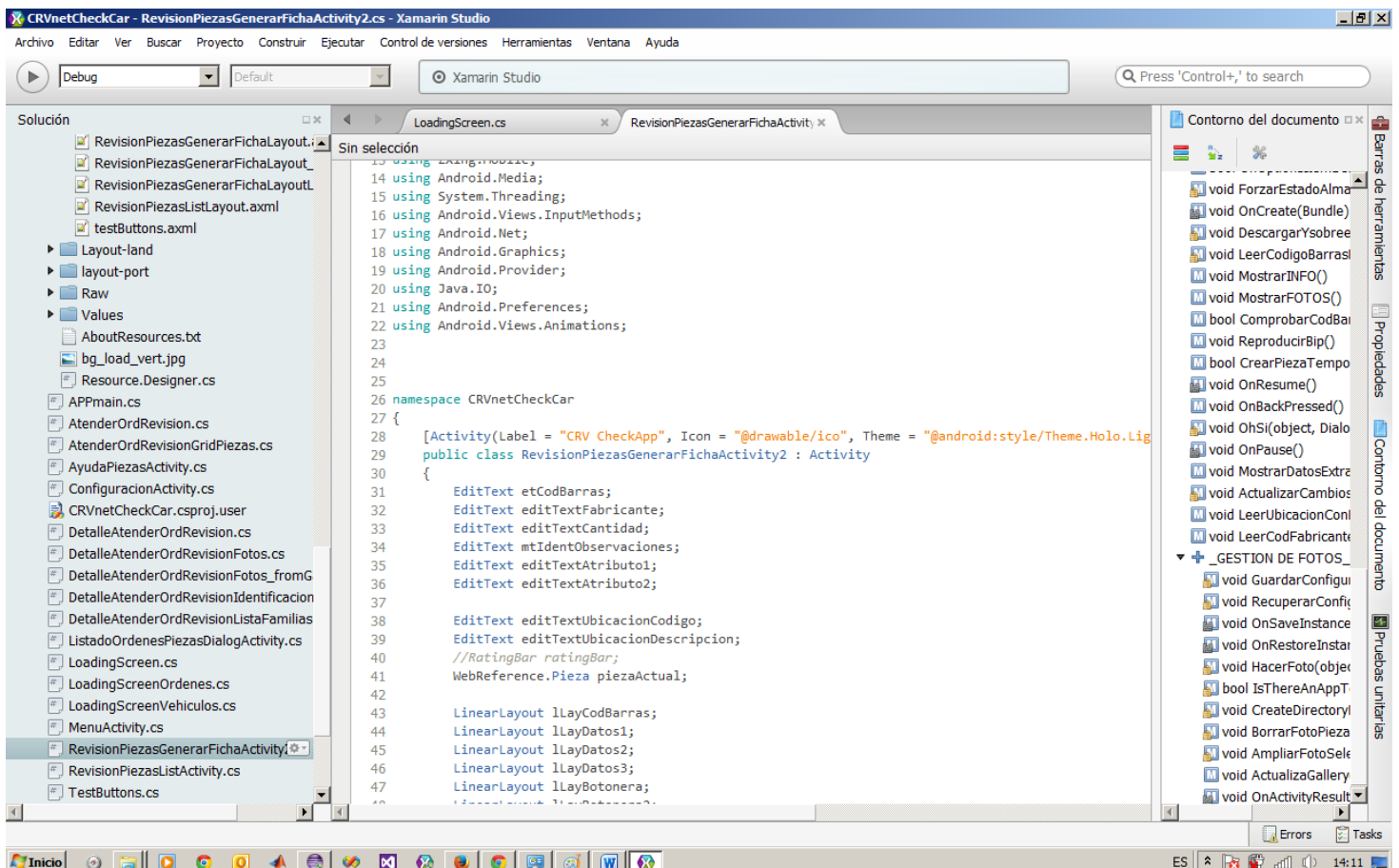


Ilustración 51: Captura de pantalla de Xamarin Studio

### 5.1.2.3 Android Studio

Ésta es la propuesta de código abierto y multiplataforma de Google, un IDE basado en *IntelliJ IDEA Community Edition* y que promete simplificar las tareas de desarrollo.

Android Studio está basado en Gradle, una herramienta para automatizar la construcción de nuestros proyectos, por ejemplo las tareas de compilación, testing, empaquetado y el despliegue de los mismos. Es muy flexible para la configuración, pero además ya tiene preconfiguradas las tareas para la mayoría de los proyectos por defecto. Esta herramienta se usa también en otros proyectos “Open Source” como “Spring”, “Hibernate”, y “Grails”. Utiliza un DSL (Lenguaje Específico de Dominio) basado en Groovy.

Características:

- Utiliza un DSL (Lenguaje Específico de Dominio) basado en Groovy (nada de XMLs complejos)
- Nos permite añadirlo a nuestros servidores de integración continua
- Construcción por convención (al estilo Maven)
- Compatible con dependencias Maven

Una de las características más novedosas, es la de poder previsualizar los recursos. Por ejemplo si estamos en un “drawable”, podemos poner el cursor sobre el identificador, pulsar Ctrl + Q (comando para ver la documentación) y visualizar automáticamente los distintos recursos.

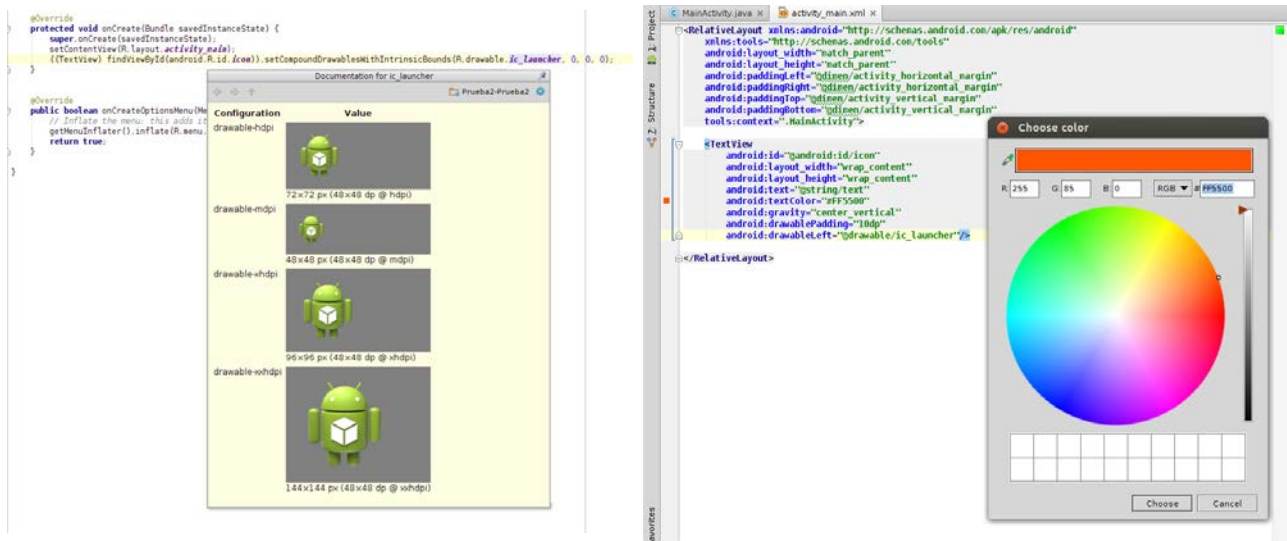


Ilustración 52: Captura de Android Studio - Previsualización de recursos

Android Studio incluye widgets para facilitar la generación de recursos para distintas configuraciones. Uno de los mejores ejemplos es a la hora de crear una cadena de texto “string”. Podemos seleccionar el desplegable de región y se nos mostrará una lista de países que podemos ir seleccionando. Esto resulta tremendamente útil porque incorpora todos los identificadores con sus variantes, así que no tendremos que recordarlos de memoria.

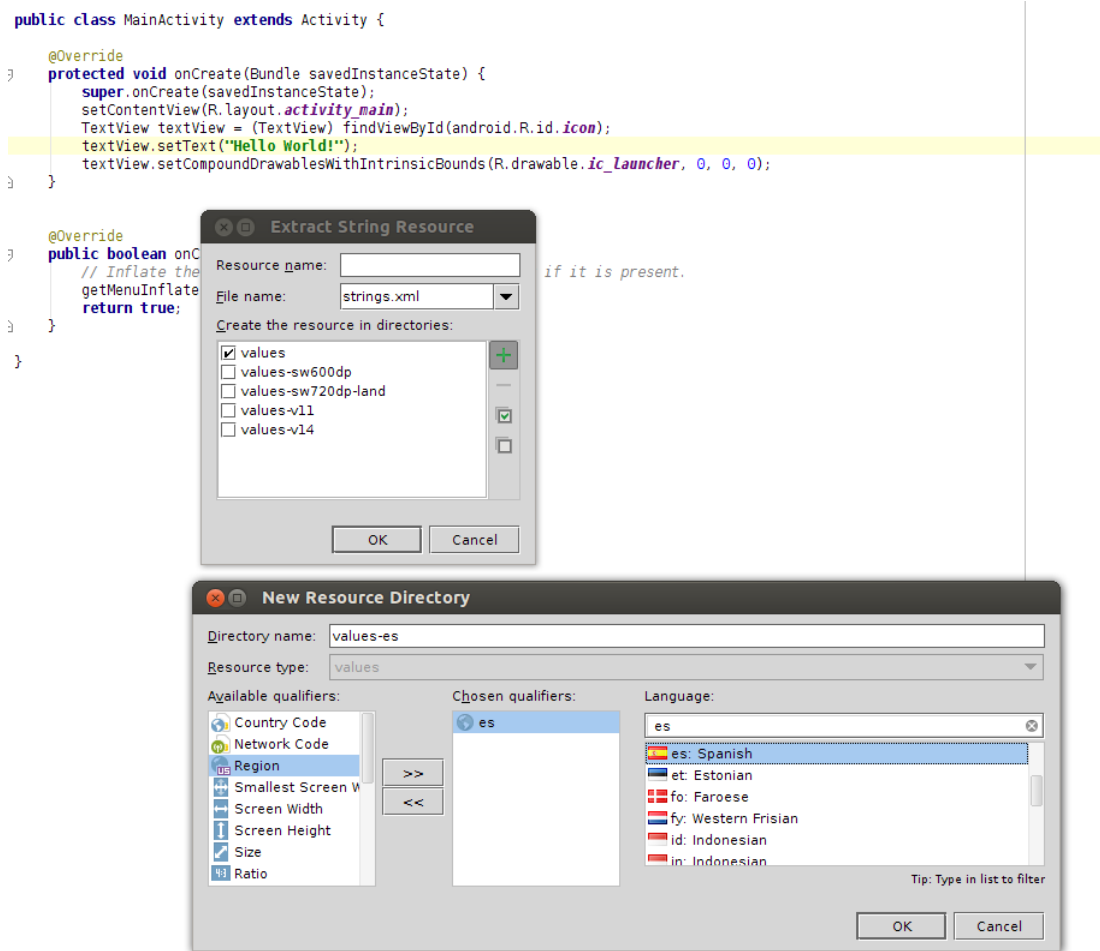


Ilustración 53: Captura de Android Studio – Generación de recursos de idioma

Este IDE incorpora de forma transparente una herramienta de análisis de código que nos ayuda a mejorar la calidad del mismo. Además se realiza de forma automática, de forma que conforme vas escribiendo te va indicando posibles fallos o mejoras que puedes corregir en el código. Métodos sin utilizar, lógica demasiado compleja o condiciones que nunca van a darse serán detectadas por el IDE y nos permitirá con pocas pulsaciones de ratón corregir dichos problemas.

La refactorización es muy potente en Android Studio. Ahora podemos cambiar el nombre de una imagen y automáticamente se cambiará el nombre del identificador. A la inversa funcionaría exactamente igual. IntelliJ posee un sistema de refactorización muy avanzado y Android Studio se beneficia de eso.

La previsualización de los layouts está bastante cuidada, marcando los elementos sobre los que estamos trabajando y permitiendo elegir entre distintos dispositivos, configuraciones, temas, etc.

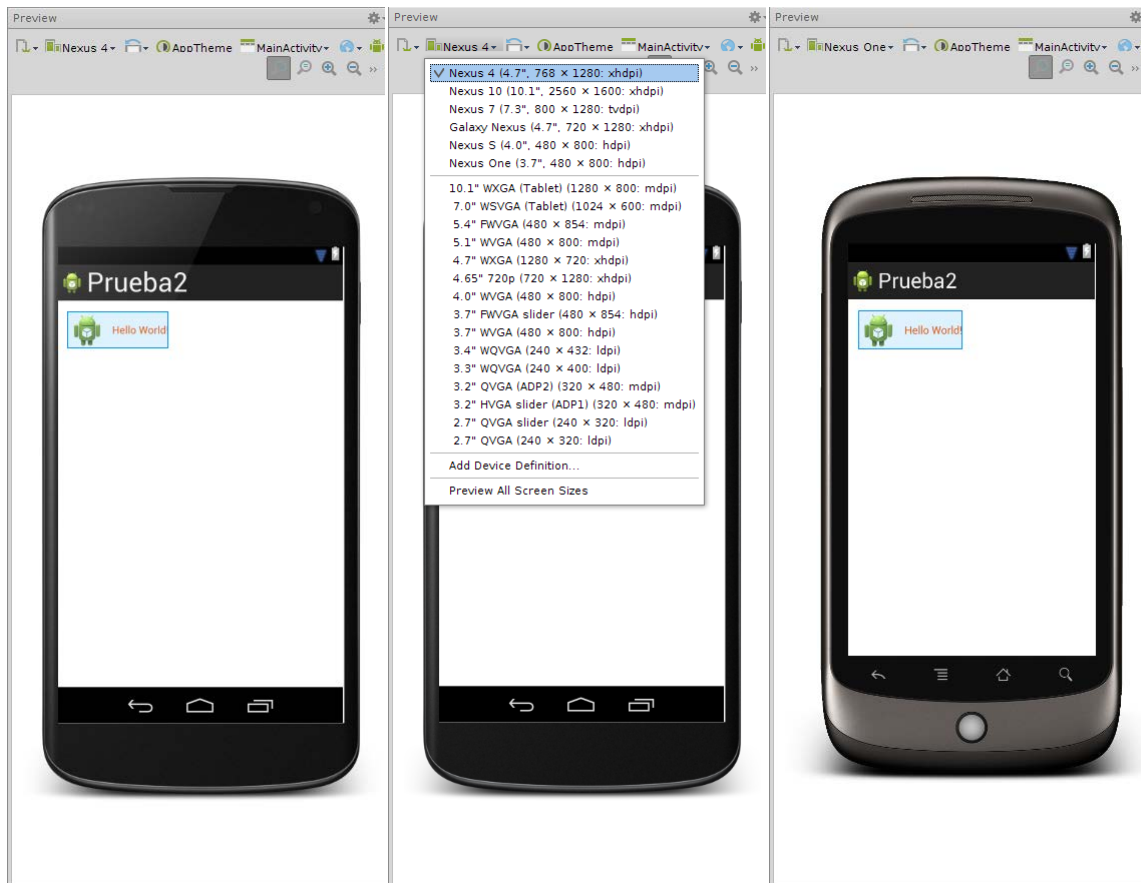


Ilustración 54: Captura de Android Studio - Previsualización de Layouts

Quizá el punto donde Google ha invertido mayor esfuerzo, ha sido en facilitar el uso de sus servicios en nuestras aplicaciones, como por ejemplo en el Google Cloud Messaging (GCM); un servicio de mensajería para comunicar aplicaciones cliente con servidor, o en la presentación de Google Play Games.

### *5.1.2.4 Selección del entorno de desarrollo.*

Tras el estudio y prueba de las ventajas e inconvenientes de cada uno de los entornos de desarrollo mencionados anteriormente, a continuación expondremos las conclusiones a las que hemos llegado.

Desde el punto de vista de la alternativa de Eclipse, tenemos una herramienta que disfruta de un nivel de madurez alto, es decir, un entorno más que probado por una comunidad de desarrolladores durante un largo periodo de tiempo y que además distribuye nuevas versiones anualmente (Callisto 2006, Europa 2007, Ganymede 2008, Galileo 2009, Helios 2010, Indigo 2011, Juno 2012, Kepler 2013) y mejoras con bastante frecuencia. Hemos observado que es fácil encontrar desde cualquier buscador soluciones a problemas comunes en procesos de configuración del entorno o de su uso habitual.

La alternativa de Google, Android Studio está cercana en lo que en poco tiempo será la mejor alternativa para el desarrollo de aplicaciones Android, una herramienta muy potente respaldada por quizá la empresa más puntera de la actualidad y que conoce mejor que nadie su propio producto. Por otro lado, hay que destacar que en este momento la herramienta se encuentra en una fase beta, en concreto en la versión Android Studio 0.2.0 y es algo que hay que considerar de cara a realizar una implementación.

Finalmente hemos concluido que aunque cualquiera de ellos podría ser un estupendo candidato como plataforma de desarrollo de aplicaciones Android, concretamente Xamarin Studio reúne el mejor conjunto de características para el desarrollo de nuestro proyecto.

La anterior conclusión se ha tomado en base a la capacidad para reutilizar librerías internas de nuestra empresa (librerías C# ya utilizadas en la aplicación de escritorio) y así acelerar el desarrollo de la aplicación móvil. Además otra de las características de mayor peso ha sido su capacidad para portar la aplicación a la plataforma iOS, minimizando los cambios en el desarrollo.

Como en toda decisión, la elección que da acceso a algún tipo de ventajas, proporciona a su mismo tiempo un conjunto inconvenientes o problemas a tratar, que describiremos en un posterior apartado, cuando tratemos el detalle de la implementación.

### 5.1.3 Documentación y herramientas internas y de terceros

Una de las fuentes nada despreciables de documentación y ejemplos la hemos encontrado en la misma página web del proveedor de la plataforma Xamarin (<http://docs.xamarin.com/samples/Android/all>), otra de las páginas donde hemos encontrado gran parte de las soluciones han sido los foros de StackOverflow, y el mismo foro oficial de Xamarin. Por otro lado la documentación del API utilizado la hemos podido consultar tanto en:

- <http://developer.android.com/guide/components/index.html>
- <http://developer.android.com/reference/packages.html>
- <http://androidapi.xamarin.com/>

Ya inmersos en el desarrollo del proyecto, como adelantábamos con anterioridad, hemos hecho uso de librerías en C# ya implementadas en la aplicación de escritorio CRVNet (para compresión de datos e imágenes, tratamiento de cadenas, generación de ficheros...) y por otro lado, ha sido posible incorporar todas las funcionalidades creadas en la aplicación móvil a la aplicación de escritorio. Por lo que podremos señalar que el avance con la elección de C# como lenguaje de programación en nuestro desarrollo ha sido bidireccional.

Por otro lado, para la implementación de una de las funcionalidades estrella de nuestra aplicación, en concreto la de leer códigos de barras desde la cámara de los dispositivos móviles, hemos incorporado el proyecto de código abierto ZXing, que cumple con eficacia el propósito de realizar esta tarea sin basarse en una resolución en el lado del servidor, es decir, es el mismo dispositivo móvil en que realiza la tarea sin la necesidad de estar conectado a la red, característica imprescindible dadas las circunstancias de la necesidad de poder disponer de nuestra herramienta aun estando sin conexión de datos.

El proyecto ZXing (pronunciado "zebra crossing"). Es una librería de código abierto de lectura de códigos multi-formato 1D/2D, incluyendo el Code 128, que es el que utilizaremos para la lectura de etiquetas de las piezas de los clientes. El objetivo del proyecto, desarrollado en el entorno de Google Code, es poder decodificar códigos QR, Data Matrix y OPC. Provee clientes para Java, iOS y Android. Lo podemos encontrar en <http://code.google.com/p/zxing/> y lo que nos permite incorporarlo a nuestro proyecto es que pese a estar desarrollado en Java, está portado (traducido) a C#, lo que han llamado ZXing .NET. (Google Project, 2013)

Otra de las Tecnologías utilizadas en el desarrollo del proyecto ha sido SOAP, para las comunicaciones entre la aplicación ¿Por qué no hemos utilizado REST para las comunicaciones en nuestra aplicación?

Detengámonos brevemente a ver las diferencias:

- REST: suele utilizar HTTP como método de comunicación y XML o JSON para intercambiar datos. Cada URL representa un objeto sobre el que se pueden utilizar los métodos POST, GET, PUT y DELETE.
- SOAP: es toda una infraestructura basada en XML, cada objeto puede tener métodos definidos por el programador con los parámetros que sean necesarios.

Con lo que se puede deducir claramente que REST es más ligero, con poca configuración, se lee fácilmente (son URLs) y no hace falta nada especial para implementarlo. El protocolo SOAP, mucho más ambicioso, consume mayor ancho de banda por la característica forma de enviar los paquetes, con un envoltorio o *"envelope"*, que define su uso) y tiene un tipado fuerte (que se puede observar en su WSDL, el Lenguaje de Definición del Servicio Web).

Por otro lado, el protocolo estricto al que se somete, garantiza una mayor robustez por el hecho de no permitir el acceso a los parámetros directamente desde la URI, como REST, pero los motivos de mayor peso por el que hemos elegido SOAP es la transparencia con la permite el envío de objetos completos al servidor (directamente desde la llamada), y que dada la situación en la que las conexiones de datos se realizan, en un ámbito local a través de la conexión Wifi de la empresa, el sobrecoste de las cabeceras de datos que incorpora SOAP frente a REST, se hace irrelevante.

## 5.2 Plataforma objetivo donde se explotará.

Tras la decisión inicial de desarrollar la aplicación para la plataforma móvil Android, encontramos las consideraciones de producir una aplicación que se pudiera explotar en la mayoría de los dispositivos del mercado, y la de basar el desarrollo en una API capaz de aprovechar las funciones más novedosas de las últimas versiones y de ofrecer un aspecto gráfico novedoso y agradable.

Conociendo de antemano que ambas circunstancias estaban enfrentadas, realizamos un estudio de mercado para asegurar la suficiente cuota de mercado evitando dejar escapar el mayor número de mejoras posibles.

Podemos comprobar en “developer.android.com”

<http://developer.android.com/about/dashboards/index.html> la información siguiente:

En dicha página Google ofrece información sobre el número relativo de los dispositivos que comparten una determinada característica, como la versión de Android o tamaño de pantalla. Esta información es de vital importancia para obtener una aproximación (dado que los cálculos se realizan sobre los dispositivos que se han conectado a Google Play en un periodo de 7 días durante el mes de septiembre de 2013) y puede ayudar a priorizar los esfuerzos de manera eficaz para garantizar nuestro acceso a la cuota de mercado que esperamos.

*Nota: A partir de septiembre de 2013, los dispositivos que ejecutan versiones anteriores a Android 2.2 no aparecen en esta información porque esos dispositivos no son compatibles con el nuevo Google Play*

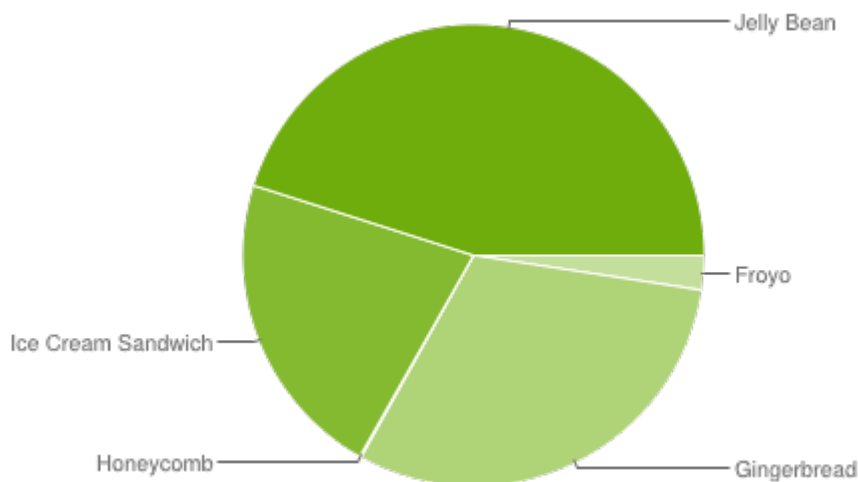


Ilustración 55: Gráfica de porciones reparto de versiones Android

Sabiendo que Google ha decidido descartar aquellos usuarios que utilizan versiones inferiores a la 2.2 nos podemos hacer una idea de las especificaciones de software por las que se mueve el mercado.



Version	Codename	API	Distribution
<a href="#">2.2</a>	Froyo	8	2.4%
<a href="#">2.3.3 - 2.3.7</a>	Gingerbread	10	30.7%
<a href="#">3.2</a>	Honeycomb	13	0.1%
<a href="#">4.0.3 - 4.0.4</a>	Ice Cream Sandwich	15	21.7%
<a href="#">4.1.x</a>	Jelly Bean	16	36.6%
<a href="#">4.2.x</a>		17	8.5%

Ilustración 56: Tabla de reparto de versiones de Android en septiembre de 2013

En la tabla anterior podemos observar que el reparto de versiones Android, la que nosotros traducimos respondiendo a la pregunta de ¿qué *CUOTA* de mercado (porcentaje) de dispositivos tienen acceso a nuestra aplicación si garantizamos una versión mínima de Android? A continuación vemos los porcentajes acumulados.

Versión	Codename	API	Distribución	CUOTA
<a href="#">2.2</a>	Froyo	8	2,40%	100%
<a href="#">2.3.3 - 2.3.7</a>	Gingerbread	10	30,70%	97,60%
<a href="#">3.2</a>	Honeycomb	13	0,10%	66,90%
<a href="#">4.0.3 - 4.0.4</a>	Ice Cream Sandwich	15	21,70%	66,80%
<a href="#">4.1.x</a>	Jelly Bean	16	36,60%	45,10%
<a href="#">4.2.x</a>		17	8,50%	8,50%

Ilustración 57: Cuotas de mercado según versión de Android

De la anterior tabla podemos extraer que el 66,80% de los dispositivos ya utilizan versiones 4.0 o superior (Ice Cream Sandwich + Jelly Bean), lo que supone un importante avance en la lucha de Android contra la fragmentación (dejando de lado la fragmentación debida al hardware). Y por otro lado, el siguiente gran escalón lo encontramos asegurando la compatibilidad hasta la versión 2.3 (Gingerbread), lo que deja un estupendo 97,6% de dispositivos a su alcance.

Por otro lado exponemos los factores que nos han condicionado en la elección del SDK mínimo de Android:

- El uso de la librería de ZXing para la funcionalidad de reconocimiento de códigos de barras nos exige funciones del API 10, por lo que trasladando dichos requerimientos a nuestro proyecto, nos vemos forzados a exigir una versión 2.3 o superior.
- Otro factor, lo encontramos en la característica del acceso al portapapeles, para implementar la función de copiar y pegar. Descubrimos que el portapapeles al que tenemos acceso en versiones superiores a la 3.0 (HoneyComb) dispone de acceso a una pila de contenidos de manera que es posible recuperar un dato anterior al último copiado.

En este caso esta característica pudo tratarse diferenciando los casos para evitar que el acceso al recurso en las versiones superiores, no limitara el SDK mínimo de nuestra aplicación. La resolución consistió en ofrecer un método alternativo para el acceso al portapapeles que sí estuviera implementado en la versión del dispositivo del cliente, sacrificando en las versiones anteriores las características extra, pero no la funcionalidad. A continuación podemos ver un ejemplo de la distinción:

```
Android.Text.ClipboardManager clipboardMenor11;  
Android.Content.ClipboardManager clipboardSDKHoney;  
  
int sdkNumber = (int) Android.OS.Build.VERSION.SdkInt;  
if (sdkNumber < (int) Android.OS.BuildVersionCodes.Honeycomb){  
    clipboardMenor11.Text = etCodBarras.Text;  
}else{  
    ClipData clip = ClipData.NewPlainText("ETIQUETA", objetoAcopiar.Text);  
    clipboardSDKHoney.PrimaryClip = clip;  
}
```

Por lo anteriormente citado, hemos podido asegurar una compatibilidad a partir de la versión 2.3, lo que nos garantiza un acceso al 97,6% del mercado en septiembre de 2013, con tendencias a incrementarse en los próximos meses. Este dato afianza uno de los requisitos iniciales que demandaban una compatibilidad de al menos el 95% de los dispositivos existentes en el mercado.

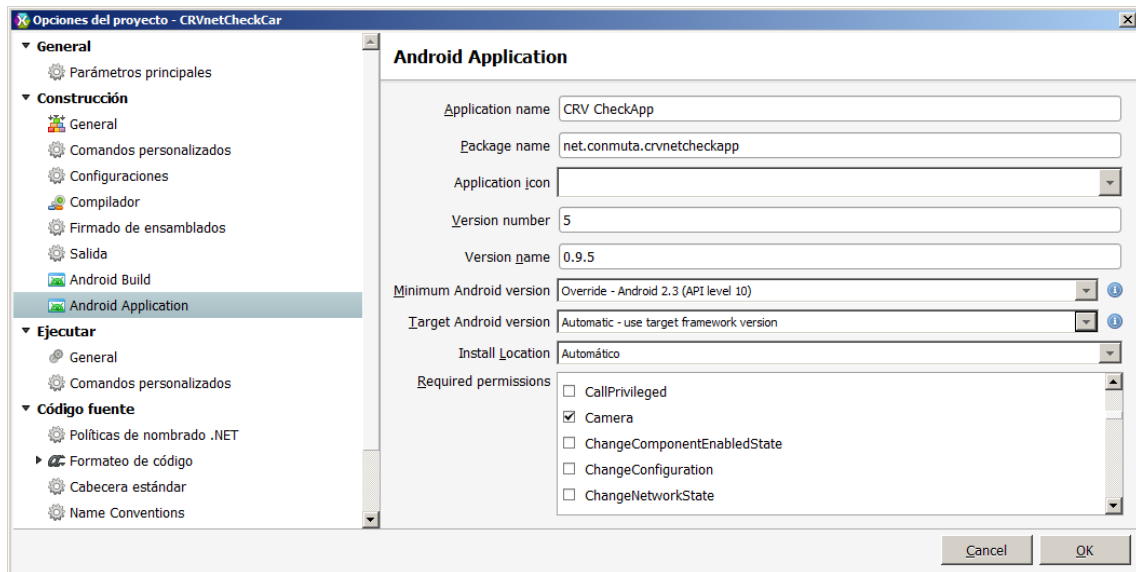


Ilustración 58: Captura de la especificación de opciones del proyecto en Xamarin Studio

## 5.3 Detalle.

### 5.3.1 Metodología de desarrollo

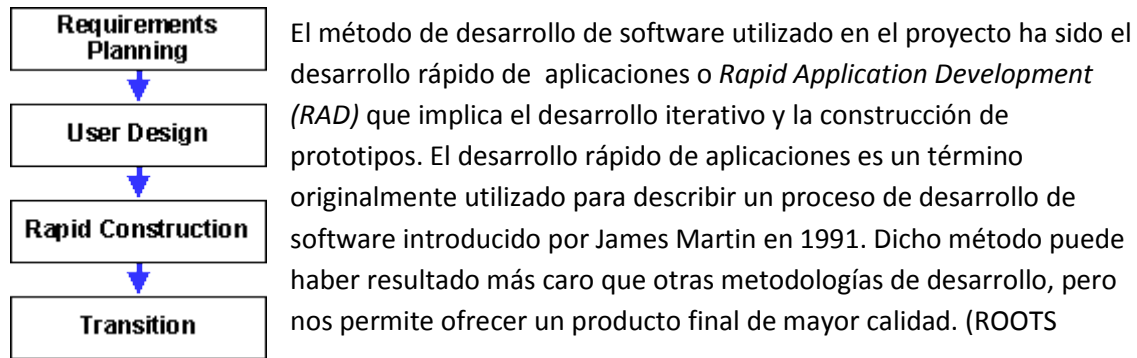


Ilustración 59: Flujo de trabajos RAD

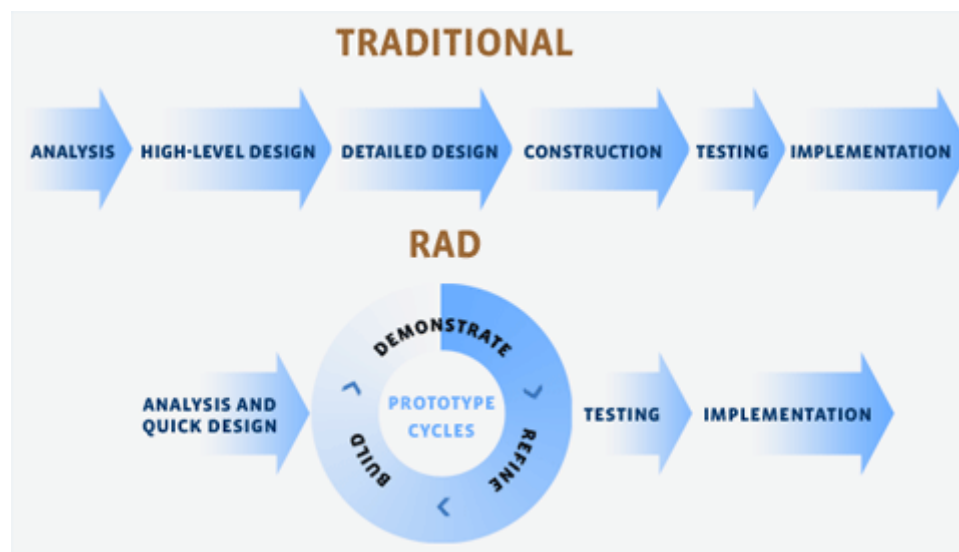


Ilustración 60: Comparación de RAD frente a las metodologías tradicionales. Fuente: [rootservices.com](http://rootservices.com)

Si comparamos RAD frente a metodologías tradicionales, podemos observar que incurre en la posibilidad de entrar en un ciclo interminable de prototipos insatisfactorios. Éste es un riesgo que hay que vigilar y controlar a través de la especificación detallada de los casos de uso.

Por otro lado, en entornos donde los requisitos están correctamente establecidos, pero corren el riesgo de ser alterados considerablemente, como ocurre en nuestro caso, hacemos uso de ese tipo de método para poner a la disposición del cliente los prototipos lo antes posible, para con ello, recibir el feedback de los usuarios en una fase temprana, y así redefinir los requisitos antes de que dicho proceso genere un coste demasiado alto.

Las características fundamentales por las que hemos elegido esta metodología son las siguientes:

- A través de RAD, se intenta reducir el riesgo inherente del proyecto partiéndolo en segmentos más pequeños y proporcionar más facilidad de cambio durante el proceso de desarrollo.
- Una orientación dedicada a producir sistemas de alta calidad con rapidez, principalmente mediante el uso de iteración por prototipos (en cualquier etapa de desarrollo), promoviendo la participación de los usuarios y el uso de herramientas de desarrollo computarizadas. Estas herramientas pueden incluir constructores de Interfaz gráfica de usuario (GUI), Computer Aided Software Engineering (CASE) las herramientas, los sistemas de gestión de bases de datos (DBMS), lenguajes de programación de cuarta generación, generadores de código, y técnicas orientada a objetos.
- Hace especial hincapié en el cumplimiento de la necesidad comercial, mientras que la parte tecnológica acotada ya por la plataforma decidida Android y las herramientas adquiridas, queda en un segundo plano.
- El control de proyecto implica el desarrollo de prioridades y la definición de los plazos de entrega. Si el proyecto empieza a aplazarse, se hace hincapié en la reducción de requisitos para el ajuste, no en el aumento de la fecha límite. Aunque en nuestro caso, esta característica la hemos ignorado al tratarse de un producto propio, cuyos requisitos son inamovibles.
- Se facilita una producción activa de software en lugar de mostrar prototipos inacabados, poniendo rápidamente al alcance del cliente una herramienta que garantiza la participación activa del mismo y un feedback que permite a los usuarios participar iterativamente en el diseño del sistema.
- Tras la finalización del producto, el cliente ya está familiarizado con la herramienta y se dispone de la documentación necesaria para facilitar el futuro desarrollo y mantenimiento.

Podemos encontrar más información de esta metodología de desarrollo en

<http://www.projectmanagement.com/content/processes/11306.cfm>

[http://en.wikibooks.org/wiki/Introduction\\_to\\_Software\\_Engineering/Process/Rapid\\_Application\\_Development](http://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Process/Rapid_Application_Development)

### 5.3.2 Estructura del proyecto

Desde el punto de vista de la distribución de clases, funciones de apoyo, plantillas y otros recursos de proyecto, hemos apostado por la distribución en varias colecciones básicas que aglutinan contenidos pertenecientes al mismo ámbito. A continuación detallaremos la distribución interna de las aplicaciones construidas.

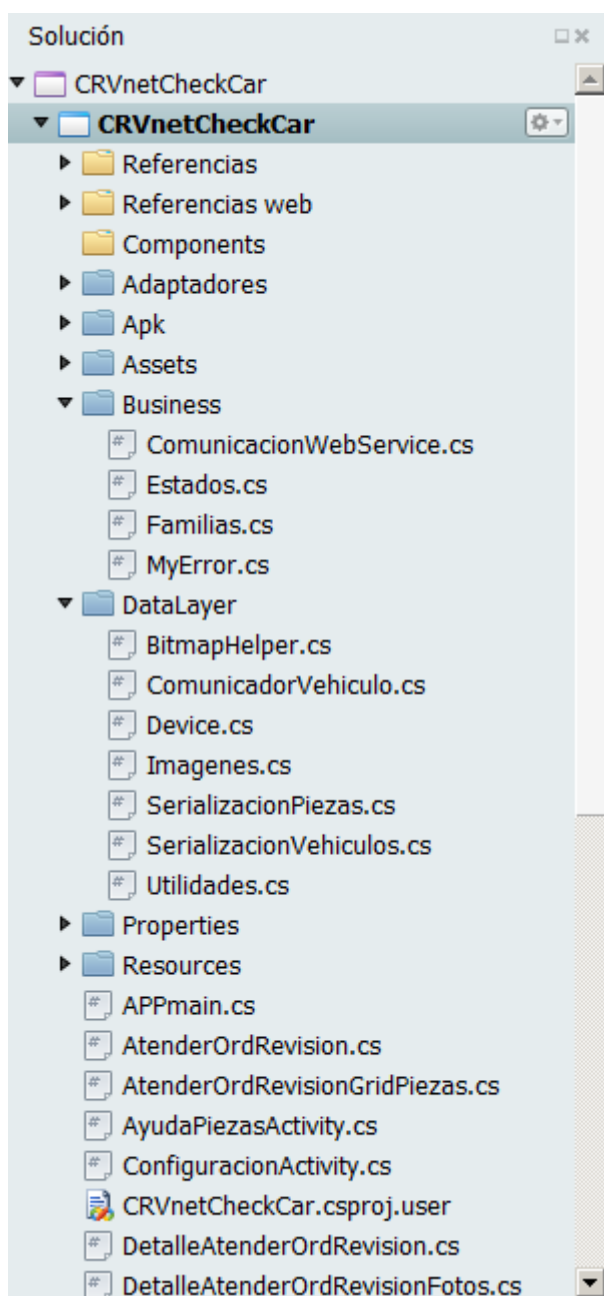


Ilustración 61: Explorador de archivos de Xamarin - Visión general

Basándonos en la similitud de los métodos utilizados en las aplicaciones CRV Campa y CRV Almacén, hemos decidido distribuir los recursos según el ámbito de aplicación en lugar de dividirlos por aplicaciones. Lo que nos permitirá tener una buena disponibilidad para el caso de optimizaciones y mejoras sobre funcionalidades parejas de ambas funciones. A continuación analizaremos las principales colecciones de funciones.

- **Funciones de negocio** (llamada *Business* en nuestro proyecto). En esta capa hemos almacenado las clases y métodos que tienen una estrecha relación con el entorno de CRVNet, donde podemos encontrar configuraciones y accesos a los servicios Web, así como el tratamiento y notificación de errores. También podemos encontrar clasificaciones y clasificadores de familias de piezas y estados o ubicaciones de las piezas. Por último también incluimos en este apartado el módulo de errores más frecuentes que solemos llamar desde cualquiera de los formularios.

- **Funciones de acceso y tratamiento de datos.** (Colección de funciones y clases que hemos llamado en nuestro proyecto *DataLayer*). Esta colección está formada por todas aquellas funciones que han requerido del acceso, uso o tratamiento de datos. Entre ellas podemos encontrar; funciones para trabajar con imágenes o la compresión de las mismas, herramientas para el paso de parámetros y configuraciones entre Activities, todas las funciones relativas a la serialización y la creación y recuperación de preferencias de la APP...
- **Colección de recursos de diseño.** Estos recursos podemos encontrarlos en la carpeta *Resources*. Xamarin, al igual que otros IDEs, en su implementación de Android, exige una estructura de directorios bastante concreta para garantizar la optimización lo recursos según las características físicas de los dispositivos móviles donde se ejecutan. Lo que permite dejar en manos del sistema operativo Android la selección de los recursos adecuados para cada dispositivo.

En el caso de los recursos gráficos como, imágenes, fotografías (.jpeg, .jpg...), y otros archivos con o sin transparencias (.png) se deben ubicar en la carpeta “drawable”, colocando según la resolución de los mismos en las carpetas -xhdpi, -hdpi, -mdpi, -ldpi según dispongamos de recursos de mayor a menor densidad de píxeles respectivamente.

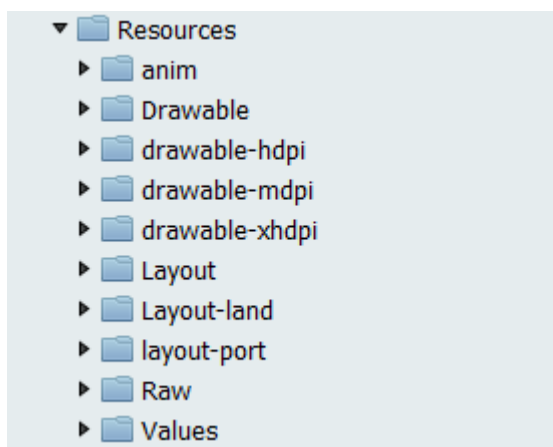


Ilustración 62: Explorador de archivos de Xamarin - Recursos del proyecto

En la carpeta Layout, tenemos todas las definiciones de las vistas que enlazamos a cada Activity, diferenciando también aquellas que encontramos en modo “-land” (en inglés, viene de “landscape”, refiriéndose a vistas apaisadas), o en “-port” (en inglés, de “portrait”, refiriéndose a la vista vertical). Esta característica propia de Android, permite redefinir distintas vistas según la orientación en la que colocamos el dispositivo.

Animaciones, en la carpeta “anim”: Se trata de ficheros XML (de texto plano), donde definimos las animaciones, en nuestro caso, se trata de animaciones básicas de tipo “Tween”; basadas en la interpolación de efectos básicos como rotaciones, translaciones, escalados y efectos de transparencias.

**Colección de recursos “Values”:** Esta carpeta recoge varios ficheros XML, que según el nombre específico de los ficheros, contienen información de recursos de tamaños creados por nosotros

(dimens.xml) o recursos de cadenas de texto con sus traducciones a los idiomas que hayamos elegido (string-es.xml, string-en.xml...).

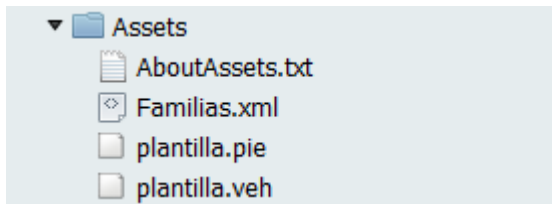


Ilustración 63: Explorador de archivos de Xamarin - Assets

- **Colección de recursos Assets**, los recursos almacenados en esta carpeta no se procesarán, se trata de recursos que cargaremos directamente desde la aplicación, como son las plantillas de vehículo y de pieza o las familias de las piezas (dado que es un recurso estático).

- **Colección de “Adaptadores”:**  
Hemos creado esta colección por la importancia que tienen dentro de nuestro software los elementos de tipo contenedor, como puedan ser las listas, las galerías de imágenes o las matrices de contenidos. Dado

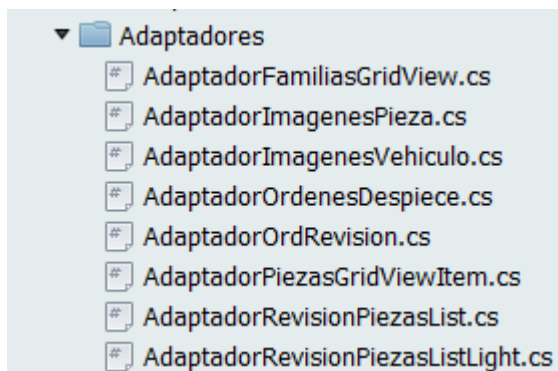


Ilustración 64: Explorador de archivos de Xamarin - Adaptadores

que en nuestro caso, hemos requerido crear elementos de listas muy concretos especificándolos en su plenitud, por lo que hemos tenido que crear un tipo “Adapter” o adaptador para cada uno de estos elementos.

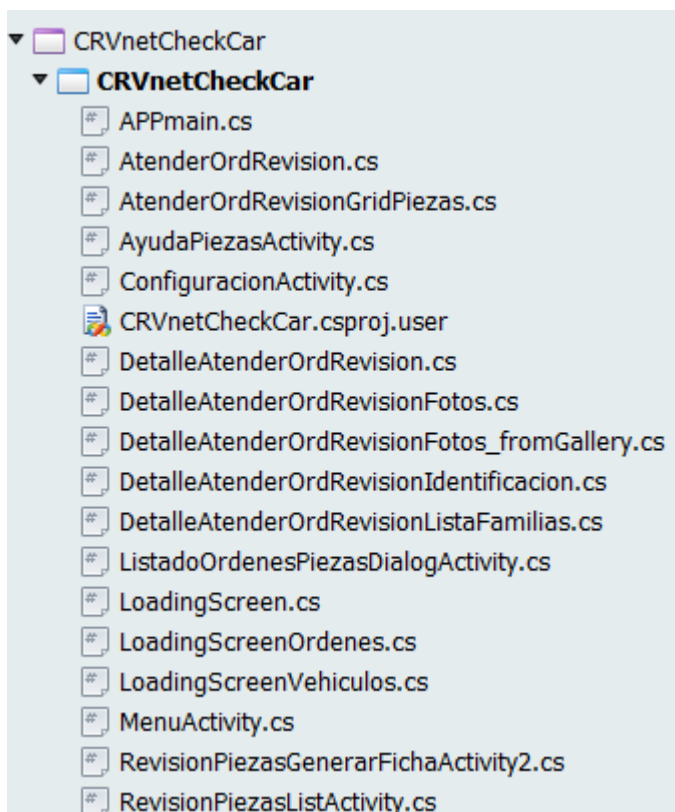


Ilustración 65: Explorador de archivos de Xamarin -Ficheros fuente

- **Colección de Actividades:** Por último haremos referencia también a los ficheros que definen el comportamiento de las actividades, que son los elementos en Android encargados de enlazar la parte visual del programa (la vista), con las acciones que se les atribuyen, como por ejemplo, la creación del evento de la pulsación de un botón, y las sucesivas acciones que se puedan realizar tras la generación de dicho evento, como la llamada a cargar otra vista, la lectura de un fichero, el intercambio de datos con el servidor, etc.



### 5.3.3 Ejemplos de recursos del proyecto

A continuación mostraremos el contenido real de ficheros existentes en la aplicación, pudiendo encontrar; vistas, actividades, plantillas, definiciones de animaciones,

#### 5.3.3.1 Ejemplo de definición de una vista:

Como ejemplo de vista hemos elegido “RevisionPiezasListLayout.xml”, que define el aspecto visual de la pantalla principal de CRV Almacén, donde encontramos una ListView (lista donde mostraremos las distintas piezas disponibles en el dispositivo), 2 Buttons ( un botón para añadir piezas nuevas y otro para enviar al servidor las piezas seleccionadas). Un TextView (en la parte superior derecha, podemos observar un rótulo que nos indica el número de piezas que tenemos disponibles en la aplicación).

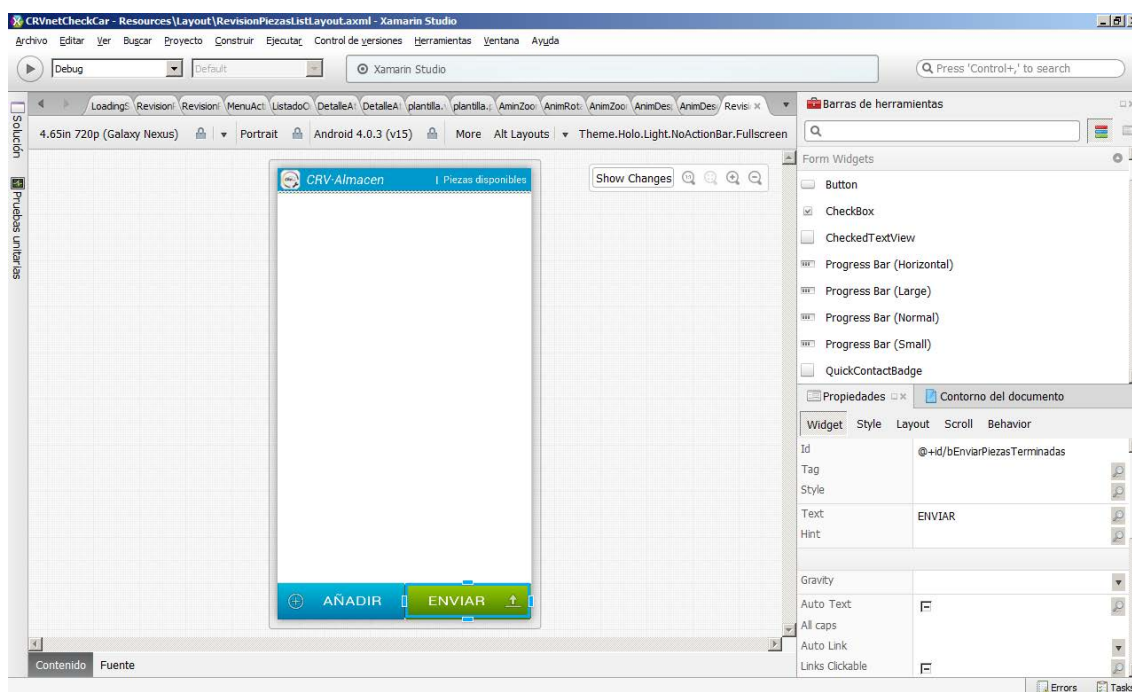


Ilustración 66: Explorador de archivos de Xamarin - Definición de una vista

A continuación mostramos el código que define esta vista; donde podemos comprobar que hemos apostado por los LinearLayout en su modo horizontal y vertical, para la colocación de

los distintos elementos dentro de la vista. En los casos en los que nos ha sido posible, hemos hecho uso de las propiedades “wrap\_content” y “fill\_parent” para asegurar que las dimensiones proporcionadas se adaptan a las características de los dispositivos donde se utiliza esta aplicación.

**Resources/Layout/RevisionPiezasListLayout.axml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:minWidth="25px"
    android:minHeight="25px"
    android:weightSum="1"
    android:background="@drawable/degradadoFondoNegro">
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="2dp"
        android:minWidth="25px"
        android:minHeight="25px"
        android:background="#ffffffff" />
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:minWidth="25px"
        android:minHeight="25px"
        android:weightSum="1"
        android:background="@drawable/degradadoFondoNegro">
        <LinearLayout
            android:orientation="vertical"
            android:minWidth="25px"
            android:minHeight="25px"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/linearLayout2"
            android:layout_weight="1"
            android:layout_gravity="fill"
            android:background="#ffcbccb">
            <ListView
                android:minWidth="25px"
                android:minHeight="25px"
                android:layout_width="fill_parent"
                android:layout_height="match_parent"
                android:id="@+id/piezasPendientes"
                android:layout_marginTop="1.7dp"
                android:layout_marginLeft="0.0dp"
                android:layout_marginRight="0.0dp"
                android:layout_marginBottom="0.0dp"
                android:fastScrollEnabled="true"
                android:fadeScrollbars="true"
                android:choiceMode="singleChoice"
                android:background="#ffffffff"
                android:layout_weight="1"
                android:clickable="true"
                android:longClickable="true"
```

```

        android:hapticFeedbackEnabled="true" />
<TextView
    android:text="Pulse AÑADIR para introducir una nueva pieza"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:layout_width="fill_parent"
    android:layout_height="match_parent"
    android:id="@+id/etNota"
    android:gravity="center"
    android:layout_marginRight="0.0dp"
    android:layout_marginLeft="0.0dp"
    android:background="@android:color/white"
    android:visibility="gone" />
</LinearLayout>
<LinearLayout
    android:orientation="horizontal"
    android:minWidth="25px"
    android:minHeight="25px"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/linearLayoutBotonera"
    android:gravity="center"
    android:layout_gravity="center"
    android:background="#ffffffff">
<Button
    android:text="AÑADIR"
    android:id="@+id/bMasPieza"
    android:background="@drawable/btnAnadirPiezas"
    android:textColor="#FFFFFF"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="0.0dp"
    android:layout_marginBottom="0.0dp"
    android:layout_marginTop="0.0dp"
    android:layout_weight="1"
    android:drawableLeft="@android:drawable/ic_menu_add"
    android:shadowColor="#000000"
    android:textScaleX="1.3"
    android:layout_marginRight="0.0dp" />
<Button
    android:id="@+id/bEnviarPiezasTerminadas"
    android:background="@drawable/btnEnviarPiezas"
    android:text="ENVIAR"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="#ffffffff"
    android:layout_marginLeft="0.0dp"
    android:layout_marginTop="0.0dp"
    android:layout_marginBottom="0.0dp"
    android:layout_marginRight="0.0dp"
    android:layout_weight="1"
    android:drawableRight="@android:drawable/ic_menu_upload"
    android:textScaleX="1.3" />
</LinearLayout>
</LinearLayout>
</LinearLayout>

```

### 5.3.3.2 Ejemplo de definición de una actividad

A lo largo del proyecto hemos definido multitud actividades con funciones muy distintas entre ellas, algunas de ellas cuentan con cerca de 3000 líneas de código, por lo que pese a ser un ejemplo estupendo para mostrar las distintas funcionalidades que se han implementado en este proyecto, es inviable su incorporación a este PFC. Como representante de actividad hemos elegido “**DetalleAtenderOrdRevisionFotos.cs**”, que es el archivo fuente encargado de definir las acciones asociadas a la creación de fotografías de los vehículos, lo que es un estupendo ejemplo para mostrar la manera de solucionar uno de los grandes hándicaps de este proyecto, la insuficiencia de memoria a la hora de realizar operaciones con las imágenes, detectada sobretodo en dispositivos de baja gama, donde la memoria RAM es muy limitada.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Android.App;
using Android.Content;
using Android.Content.PM;
using Android.Graphics;
using Android.OS;
using Android.Provider;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using Java.IO;
using System.Threading;
using Android.Preferences;

namespace CRVnetCheckCar {

    [Activity(Label = "Captura de imágenes del vehículo", Theme = "@android:style/Theme.Holo.Light.NoActionBar.Fullscreen")]
    public class DetalleAtenderOrdRevisionFotos : Activity {

        private WebReference.Vehiculo veh = null;
        static Java.IO.File _file;
        static Java.IO.File _dir;
        ImageView _imageView;
        Gallery gallery;
        int posSelected = -1;

        protected override void OnCreate(Bundle bundle) {
            base.OnCreate(bundle);

            SetContentView(Resource.Layout.DetalleAtenderOrdRevision);

            Window.SetFlags(WindowManagerFlags.Fullscreen, WindowManagerFlags.Fullscreen);

            _imageView = (ImageView)findViewById<ImageView>(Resource.Id.imageView1);
            gallery = (Gallery)findViewById<Gallery>(Resource.Id.gallery1);
        }
    }
}
```

```

string codVeh = Intent.GetStringExtra("CodVehiculo");
veh = DataLayer.Comunicador.getVehicule();
if (veh.Codigo != codVeh) {
    return;
}

Button makeFoto = findViewById<Button>(Resource.Id.bHacerFoto);
Button deleteFoto = findViewById<Button>(Resource.Id.bBorrarFoto);

makeFoto.Click += HacerFoto;
deleteFoto.Click += delegate { BorrarFotoVehiculo(); };

ImageButton ibGuardar5 = findViewById<ImageButton> (Resource.Id.ibGuardar5);
ibGuardar5.Click += delegate {
    OnBackPressed();
};

//Para evitar que la transicion pierda la referencia al filename (path físico de la foto) si Android lo
//cierra por falta de memoria.
try{
    if (bundle != null){
        _file = new File(bundle.GetString ("filename"));
    }
}catch(Exception e){
    if (Business.EstadoDispositivo.MODO_DEBUG_ACTIVADO){
        Toast.MakeText (this, "Error en la recarga\n" + e.Message, ToastLength.Short).Show ();
    }
}
}

protected override void onSaveInstanceState(Bundle bundle) {
    base.OnSaveInstanceState(bundle);
    try{
        if (_file!=null && _file.AbsolutePath!=null){
            bundle.PutString("fileName", _file.AbsolutePath);
            GuardarConfiguracion (_file);
        }
    }catch{
    }
}

protected override void OnRestoreInstanceState (Bundle savedInstanceState) {
    base.OnRestoreInstanceState (savedInstanceState);

    //Para evitar que la transicion no pierda la referencia al filename si se cierra.
    if (savedInstanceState != null){
        try {
            _file = new File(savedInstanceState.GetString ("filename"));
        }catch(Exception e){
            if (Business.EstadoDispositivo.MODO_DEBUG_ACTIVADO){
                Toast.MakeText (this, "Error en la recarga\n" + e.Message, ToastLength.Short).Show ();
            }
        }
    }
}

```

```

        if (_file == null){
            _file = RecuperarConfiguracion ();
        }
    }
}

private void BorrarFotoVehiculo() {
    if (posSelected != -1) {
        DataLayer.Imagenes.EliminarImagenVehiculo(veh, posSelected);
        Toast.MakeText(this, "Se ha eliminado la imagen", ToastLength.Short).Show();
        actualizaGallery();

        //Colocamos una imagen por defecto
        _imageView.SetImageResource (Resource.Drawable.coche14);
        posSelected = -1;
    }
    else {
        Toast.MakeText(this, "No se ha seleccionado ninguna imagen a borrar", ToastLength.Short).Show();
    }
}

public void actualizaGallery(){
    //Recargamos los valores del adaptador
    gallery.Dispose ();
    gallery = (Gallery)FindViewById<Gallery>(Resource.Id.gallery1);
    Adaptadores.AdaptadorImagenesVehiculo.RecargaFotos();

    gallery.Adapter = new Adaptadores.AdaptadorImagenesVehiculo(this);
    gallery.ItemClick += delegate(object sender, AdapterView.ItemClickEventArgs args) {
        posSelected = args.Position;
        _imageView.SetImageBitmap(Adaptadores.AdaptadorImagenesVehiculo.GetBitmap(args.Position));
    };

    if (gallery.Count > 0){
        //mostramos la última foto tomada
        gallery.SetSelection(gallery.Count - 1);
    }
}

protected override void OnResume() {
    base.OnResume();
    actualizaGallery();
}

private void HacerFoto(object sender, EventArgs eventArgs) {
    if (IsThereAnAppToTakePictures()) {
        CreateDirectoryForPictures();
        Intent intent = new Intent(MediaStore.ActionImageCapture);
        //Creamos el fichero temporal que actua como contenedor de la foto
        _file = new File(_dir, String.Format("tempPhoto_{0}.jpg", Guid.NewGuid()));
    }
}

```

```

        if (_file == null || _file.AbsolutePath.Equals(null)){
            Toast.MakeText(this, "Camera: Error 0x07: " + "No se ha podido crear el fichero temporal. Reintentelo de nuevo.", ToastLength.Long).Show();

            //Si no podemos crearlo, devolvemos al usuario al formulario de la camara.
            return;
        }

        try{
            intent.PutExtra(MediaStore.ExtraOutput, Android.Net.Uri.FromFile(_file));
            StartActivityResult(intent, 0); //Llama a OnActivityResult
        }catch (Exception e){
            Toast.MakeText(this, "Camera: Error 0x01: " + e.Message, ToastLength.Long).Show();
        }
    }
    else return;
}

private bool IsThereAnAppToTakePictures() {
    Intent intent = new Intent(MediaStore.ActionImageCapture);
    IList<ResolveInfo> availableActivities = PackageManager.QueryIntentActivities(intent, PackageManager.ResolveInfoFlags.MatchDefaultOnly);
    return availableActivities != null && availableActivities.Count > 0;
}

private void CreateDirectoryForPictures() {
    try {
        _dir = new File(Android.OS.Environment.GetExternalStoragePublicDirectory(Android.OS.Environment.DirectoryDcim), "CRVnet");

        if (!_dir.Exists()) {
            _dir.Mkdirs();
        }
    }catch (Exception e){
        Toast.MakeText(this, "Camera: Error 0x04: " + e.Message, ToastLength.Long).Show();
    }
}

protected override void OnActivityResult(int requestCode, Result resultCode, Intent data) {

    if (resultCode.Equals(Result.Canceled)) return;

    int height = 600;
    int width = 800;

    string mens = "";
    if (!resultCode.Equals (Result.Ok)) {
        mens += "No se ha podido generar la foto.";
        Toast.MakeText (this, mens, ToastLength.Long).Show ();
    }
}

```

```

Intent mediaScanIntent = new Intent(Intent.ActionMediaScannerScanFile);
Android.Net.Uri contentUri = Android.Net.Uri.FromFile(_file);
mediaScanIntent.SetData(contentUri);
SendBroadcast(mediaScanIntent);

try{
    using (Bitmap bitmap = DataLayer.BitmapHelper.LoadAndResizeBitmap(_file.AbsolutePath, width, height)) {

        //Mostramos la imagen en el Preview
        try{
            _imageView.SetImageBitmap (bitmap);
        }catch (Exception e){
            Toast.MakeText(this, "Camera: Error 0x02.0: " + e.Message, ToastLength.Long).Show();
        }

        //Guardamos la nueva foto en el vehículo que tenemos en memoria
        try {
            WebReference.Vehiculo veh = DataLayer.Comunicador.getVehicle ();
        }catch (Exception e){
            Toast.MakeText(this, "Camera: Error 0x02.1: " + e.Message, ToastLength.Long).Show();
        }

        try {
            DataLayer.Imagenes.AgregarImagenVehiculo (veh, DataLayer.Imagenes.ConvertBitmapToImageType (bitmap, "fotoAndroid"));
        }catch (Exception e){
            Toast.MakeText(this, "Camera: Error 0x02.2: " + e.Message, ToastLength.Long).Show();
        }

    }
}catch (Exception e){
    string mensaje = "Camera: Error 0x02: " + e.Message;

    if (_file != null && _file.AbsolutePath != null) {
        mensaje += "\nABSOLUTE PATH: " + _file.AbsolutePath;
    } else {
        mensaje += "\n\nNo se ha podido guardar la foto. Reintentelo por favor.";
    }

    Toast.MakeText(this, mensaje, ToastLength.Long).Show();
}

//Borramos la foto de disco.
try{
    if (_file!=null && _file.AbsolutePath!=null){
        string ruta = _file.AbsolutePath;
        DataLayer.Imagenes.BorrarImagenDisco (ruta);
    }
}catch (Exception e){

    string mensaje = "Camera: Error 0x03: " + e.Message;
    if (_file != null && _file.AbsolutePath != null) {
        mensaje += "\nABSOLUTE PATH: " + _file.AbsolutePath;
    } else {
        mensaje += "\n\nNo se ha podido acceder a la foto.";
    }
}

```



```

        Toast.makeText(this, mensaje, ToastLength.Long).Show();
    }
    base.OnActivityResult(requestCode, resultCode, data);
}

void GuardarConfiguracion (File f) {
    ISharedPreferences prefs = PreferenceManager.DefaultSharedPreferences (this);
    ISharedPreferencesEditor editor = prefs.Edit ();
    editor.PutString ("_tempFile", f.AbsolutePath);
    editor.Commit ();
}

File RecuperarConfiguracion () {
    //Recuperamos las configuraciones guardadas anteriormente.
    ISharedPreferences prefs = PreferenceManager.DefaultSharedPreferences (this);
    return new File(prefs.GetString ("_tempFile", ""));
}

}
}

```

### 5.3.3.3 Ejemplo de plantilla de piezas:

Para asegurar la persistencia de los datos, hacemos uso de plantillas que serializamos y deserializamos para consultar o añadir información. Estas plantillas se definen desde la aplicación de escritorio, y son configurables por los usuarios, por lo que pueden ser actualizadas y son descargadas al dispositivo cada vez que se detectan cambios. Cada vez que se crea una nueva pieza, se crea en memoria un objeto, con las propiedades inicializadas que extrae de esta plantilla.

#### **Resources/Assets/plantilla.pie**

```
<?xml version="1.0" encoding="utf-8"?>
<Pieza xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" RefId="0" RefPieza="" Atributo1="" Atribu
to2="" PrecioCompra="1" Precio="0" Estado="Material de segunda mano" Ubicacion="Almacenada" Ob
servaciones="" Sincroniza="Procesando">
  <Clase IdRef="0" Interes="0" PrecioLocal="0" Pvr="0" PrOE="0" PrIA="0" PrSIDE="0" xmlns="http://re
cambioverde.class/">
    <Articulo>
      <Codigo>016001</Codigo>
      <Descripcion>NO IDENTIFICADO</Descripcion>
      <CodCompat>000000</CodCompat>
    </Articulo>
  </Clase>
  <Version xmlns="http://recambioverde.class/">
    <Codigo>060001000</Codigo>
    <Descripcion>VERSION ANTERIOR '0 </Descripcion>
    <CodCompat>01000000</CodCompat>
  </Version>
  <Localizacion xmlns="http://recambioverde.class/">
    <Codigo>100027808</Codigo>
    <Nombre>ALMACEN PRINCIPAL</Nombre>
    <Domicilio />
    <CodigoPostal />
    <Pais />
  </Localizacion>
</Pieza>
```

### 5.3.3.4 Ejemplo de definición de una animación:

Para ofrecer al usuario un feedback de mayor calidad, hemos hecho uso de animaciones de tipo “Tween”, que son animaciones que requieren poco esfuerzo computacional, basadas en un serie de transformaciones como translaciones, escalados, rotaciones y transparencias... Estas animaciones pueden ser realizadas a través de ficheros XML o por código, aunque nosotros hemos elegido la primera opción por resultar de esta manera más legibles y reutilizables.

A continuación podemos ver un ejemplo de la animación propuesta para simular el efecto de descargar los datos de una pieza del servidor. La animación consiste en mostrar la vista del formulario donde se especifican los datos de la pieza, desplazándose verticalmente desde el exterior de la pantalla hasta ocupar el lugar original en la vista. El efecto simula la situación caída de arriba (el servidor) abajo (la aplicación) de la información consultada.

#### **Resources/anim/efectoDescargaPieza.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">

  <translate
    android:interpolator="@android:anim/linear_interpolator"
    android:fromXDelta="0%"
    android:toXDelta="0"
    android:fromYDelta="-100%"
    android:toYDelta="0%"
    android:duration="400"
    android:fillEnabled="true"
    android:fillBefore="true"
    android:fillAfter="true" >
  </translate>

</set>
```

### 5.3.4 Problemas surgidos durante la implementación

#### Cambios en las especificaciones.

La necesidad de satisfacer las necesidades de la mayor cantidad posible de clientes hace replantearse las especificaciones de la aplicación constantemente y readaptar en la medida de lo posible la interfaz y las funciones desarrolladas. A veces, los cambios pueden ser tan bruscos que hagan descartar un prototipo por completo.

Durante el desarrollo de este producto, en ocasiones se han desarrollado varias aplicaciones para un mismo propósito, se han implementado funcionalidades que con posterioridad han sido descartadas, y se ha dedicado un gran esfuerzo para ofrecer un usuario una interfaz al mismo tiempo que una aplicación estable.

Uno de los grandes cambios en las especificaciones de requisitos, se ha podido observar al transformar una aplicación que se orientaba principalmente a dispositivos de gran tamaño, pasando de tabletas a Smartphones, dado que desde la experiencia del usuario, se pudo comprobar que un teléfono móvil era un dispositivo más manejable y adecuado para realizar actividades que en muchas ocasiones precisaban de disponer de mayor maniobrabilidad, y permitiendo también, poder ser llevado encima con mayor naturalidad, cuando en el pretratamiento de algún proceso, el usuario necesitaba tener las manos libres.

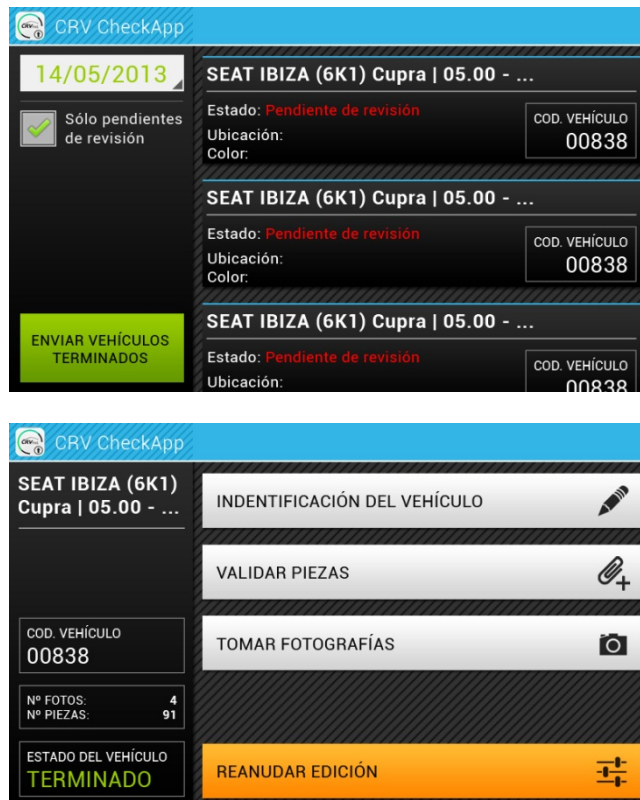


Ilustración 67: Capturas de un prototipo descartado de aplicación de CRV Campa

### Durante la documentación y codificación

Una de las situaciones más repetidas cuando llevábamos a cabo el proceso de documentación previo a la implementación de una característica requerida, ha reiterado con la dificultad para encontrar documentación específica para la plataforma Xamarin con C#, lo que ha repercutido en un sobreesfuerzo en la localización de información fiable y en la implementación en C# respecto a lo que hubiera supuesto la misma situación en Java.

Durante el transcurso del desarrollo, ha sido frecuente acudir a la fuente de documentación del algún API de Xamarin y encontrarnos con que la página no está disponible todavía.

**Mono.Xml.MiniParser.IMutableAttrList**

*Documentation for this section has not yet been entered.*  
 See Also: [MiniParser.IMutableAttrList Members](#)

[Open in Native Browser](#)

---

**Syntax**

```
public interface MiniParser.IMutableAttrList : MiniParser.IAttrList
```

---

**Remarks**

*Documentation for this section has not yet been entered.*

---

**Requirements**

**Namespace:** Mono.Xml  
**Assembly:** Mono.Security (in Mono.Security.dll)

Ilustración 68: Captura de una notificación de API no documentada en Xamarin

Otro de los aspectos negativos ha sido que pese a que los métodos y atributos en muchas ocasiones comparten nomenclaturas bastante similares con Java, a veces los parámetros requeridos han sido bastante distintos, por lo que algunos ejemplos encontrados para Java han sido difícilmente reutilizables en nuestra situación.

### Problemas con el entorno de Xamarin Studio

Tras empezar el desarrollo en una versión de prueba de 30 días de Xamarin Studio que permitía desarrollar sobre Visual Studio 2012, adquirimos la versión con licencia *Indie* (para un solo desarrollador), tras adaptar la licencia al proyecto para poder seguir trabajando, descubrimos que la versión *Indie* no permite incorporación con Visual Studio, por lo que a partir de ese momento tuvimos que descartar esta característica y trabajar únicamente desde Xamarin Studio.

	STARTER FREE	INDIE \$299 / year <small>Per platform, per developer</small>	BUSINESS \$999 / year <small>Per platform, per developer</small>	ENTERPRISE \$1899 / year <small>Per platform, per developer</small>
<u>Permitted Use</u>	Individual	Individual	Organization	Organization
<u>Deploy to Device</u>	✓	✓	✓	✓
<u>Deploy to App Stores</u>	✓	✓	✓	✓
<u>Xamarin Studio</u>	✓	✓	✓	✓
<u>Unlimited App Size</u>		✓	✓	✓
<u>Visual Studio Support</u>			✓	✓
<u>Business Features</u>			✓	✓
<u>Prime Components</u>				✓
<u>Email Support</u>			✓	✓
<u>One Business Day SLA</u>				✓
<u>Hotfixes</u>				✓
<u>Technical Kick-off Session</u>				✓
<u>Code Troubleshooting</u>			At Extra Cost	At Extra Cost
	<a href="#">Download</a>	<a href="#">Subscribe</a>	<a href="#">Subscribe</a>	<a href="#">Subscribe</a>

Ilustración 69: Plan de precios y características incluidas de Xamarin Studio

Un hecho aislado, pero suficientemente desafortunado, transcurrió mientras realizábamos una actualización de las denominadas “Estables” proporcionada por Xamarin, que tras la finalización del proceso, dejó el entorno inutilizable, impidiendo el acceso a la herramienta, ni siquiera tras su desinstalación completa y su posterior reinstalación. Necesitamos alrededor de dos días completos para recuperar el sistema, tras la inspección y modificación minuciosa del registro de Windows y la eliminación de cachés que se almacenaban en nuestro ordenador de desarrollo.

Una característica no deseable encontrada en Xamarin, la ubicamos al descubrir que no hemos tenido manera alguna de forzar el cambio de codificación de *UTF8* a *ISO 8859 Latin 1*, para poder incorporar a nuestros textos vocales con acentos (´). Finalmente este problema no ha podido ser resuelto configurando Xamarin, aunque sí ha sido posible incorporar estos caracteres no permitidos mediante la escritura de los textos en un editor de textos estándar y la copia y pegado de dicho contenido a Xamarin. Ésta es una característica que sí era realizable desde Visual Studio directamente a través del IDE.

## 6 Despliegue, explotación y evaluación.

La primera iteración de producción se realizó en las instalaciones de un cliente en el que se adaptó manualmente el servidor para que pudiera servir las peticiones de nuestra aplicación móvil, que todavía en un estado semifuncional, no tardó en recordarnos que se trataba de un prototipo de pruebas en fase de evaluación.

En general, la primera toma de contacto con los clientes, nos sirvió para testear, con usuarios no contaminados por nuestra experiencia de usuario y observar como respondía nuestra aplicación ante comportamientos y modos de uso para los que la aplicación no había sido preparada. Por otro lado, estas excursiones nos sirvieron para observar de primera mano cómo respondía el usuario ante las reacciones provocadas en la aplicación, lo que nos permitió rediseñar unos casos de uso que técnicamente eran ideales, a comportamientos que en la práctica minimizaban el número de interacciones requeridas por parte del usuario, de manera que el proceso resultaba más ligero para el usuario.

Otro de los puntos fuertes de la interacción con los usuarios, lo descubrimos al encontrar en ellos una fuente de dispositivos Android con características físicas muy diversas. Encontrando, por fortuna para nosotros, algunos dispositivos que no respondían correctamente ante ciertas acciones en el programa. Habitualmente éstos dispositivos eran de una gama relativamente baja, y nos permitió optimizar nuestra aplicación para que controlara fallos de insuficiencia de memoria (que ocurrían con frecuencia cuando se procesaban imágenes en crudo al realizar las fotografías), o situaciones en las que Android expulsa nuestra actividad de la pila de procesos.

El incómodo proceso provocado por las continuas actualizaciones (para el testeo de nuevas características y corrección de errores), realizadas mediante la descarga del contenedor *APK*, a través de una URL en el navegador, propició el desarrollo de un sistema de actualizaciones automático que es capaz de dispararse al detectar una nueva versión de la aplicación y ofrecer al usuario la posibilidad de actualizarla sin hacer uso de Google Play.

La llegada de nuestro software hizo plantearse a muchos clientes la adquisición de terminales móviles o tabletas para la explotación de las aplicaciones, con lo que podría crear un futuro mercado de venta terminales directamente desde nuestra empresa. Pese a ello, y mientras se evalúa la iniciativa, hemos decidido especificar unas configuraciones hardware mínimas para ofrecer a nuestros clientes la posibilidad de elegir el terminal, sin arriesgarse a que no puede cumplir alguna de las funciones del software. A continuación detallamos las dos configuraciones especificadas:

### Configuración mínima:

*Android 2.3*

*Procesador: 1GHz*

*Resolución pantalla: 480 x 800 o similar*



*RAM 512MB (con almenos 200MB disponibles)*

*Cámara con **Autofocus** y 3 Mega píxeles*

*Conexión WIFI*

*Nota: los requisitos mínimos relativos a la RAM y procesador deben incrementarse si el dispositivo dispone de un sistema operativo superior a Android 2.3. El motivo de ello es porque el sistema operativo consumirá mayor cantidad de estos recursos disminuyendo el rendimiento en la aplicación.*

### **Configuración recomendada:**

*Android 4.0 o superior*

*Procesador: Dual Core 1.2 GHz o superior*

*Resolución pantalla: 480 x 800 o superior*

*RAM 1 GB o superior (con almenos 300MB disponibles)*

*Cámara con **Autofocus**, 5 Mega píxeles y **Flash***

*Conexión WIFI*

Tras el estudio de las especificaciones, pudimos comprobar un motivo más, por el cual debíamos recomendar a nuestros clientes el uso de móviles en lugar de tabletas; dicho motivo se basa en la calidad de la cámara, y en la característica indispensable de disponer de autofocus, dado que esta característica es la que permite la lectura de códigos de barras. Tras analizar las alternativas de dispositivos disponibles en el mercado, observamos que las tabletas que cumplen estos requisitos, se encuentran en una gama de precio muy superior, de manera que requieren realizar una mayor inversión para disponer de las características que podemos encontrar en un móvil precio muy inferior.

Desde el punto de vista de las comunicaciones, la puesta en producción descubrió que la velocidad del envío de información, que en un principio no podía ser una limitación técnica, dado que las conexiones se realizarían por Wifi, cabía la posibilidad de que la distancia al punto de acceso fuera lo suficiente como para que se limitara la velocidad, hasta el punto de alcanzar los tiempos de espera límites (*timeouts*) desde el lado del cliente o del servidor, impidiendo así que ciertas operaciones pudieran procesarse con normalidad.

Estas limitaciones sólo se daban en circunstancias de envío de objetos con grandes cargas, pero en la actualidad se han corregido cambiando el modo de comunicación con el servidor permitiendo en envío fragmentado de la misma información de manera dichos envío aportan información de control en ambos lados (cliente y servidor), asegurando la atomicidad del proceso o descartando la comunicación entera y restableciendo el estado anterior en los dos lados.

La fase de evaluación y pruebas concluye el ciclo de vida de nuestro proyecto y lo prepara para ponerlo al alcance de la comunidad clientes. Tras las numerosas pruebas realizadas en varios clientes y tras la satisfacción expresada por los mismos, podemos asumir que la herramienta está lista para ser distribuida, y con ello termina este PFC.

## 7 Conclusión

Para terminar, expondremos las conclusiones a las que hemos podido llegar a lo largo del desarrollo de este Proyecto Final de Carrera. Dejando de lado el ámbito académico, tenemos que ser conscientes de que el desarrollo realizado se incorpora inmediatamente a un software comercial y a una imagen de marca donde el mínimo error devalúa la percepción de calidad del usuario de la herramienta global. Bajo esta premisa y bajo la responsabilidad que ella comporta, se ha invertido un gran porcentaje del esfuerzo total, en la realización de pruebas y en la reespecificación de requisitos y remodelación de casos de uso, con el objetivo de proporcionar un producto fiable y capaz de solventar la mayor parte de los procesos de negocio identificados en los clientes. Por ello, el tiempo de desarrollo ha podido verse afectado, alargando la fecha de publicación por encima del periodo previsto para una implementación directa de unas especificaciones de volumen similar.

Pese a la demora en el tiempo comentada, ha sido gratificante observar la aceptación que han tenido el pack de herramientas en los clientes, respondiendo por completo a las especificaciones determinadas y proporcionando a la plataforma CRVNet, la ventaja de ser la primera de las alternativas de su sector, en disponer de herramientas para Android integradas en su software de escritorio.

Durante la fase de análisis de mercado, a través del estudio de la competencia, pudimos comparar la potencia de nuestra solución frente a alternativas de software para móviles de otras empresas, que estando estancadas en plataformas retrógradas únicamente podían hacer uso de dispositivos con especificaciones hardware contemporáneas.

Desde el punto de vista de la codificación/implementación del producto, el sobreesfuerzo de desarrollar en C#, deja a su paso un balance positivo al descubrir una curva de aprendizaje que permite llegar a un nivel medio-alto con rapidez, gracias a la similitudes entre Java y C# y a la consideración de los incursores del proyecto Mono, de mantener la nomenclatura de los métodos, permitiendo así al programador, encontrar la vinculación de las funciones entre ambos lenguajes. Por dicha circunstancia, por la acumulación librerías y de herramientas de fabricación propia que se han desarrollado durante el transcurso del proyecto, y por la experiencia adquirida, podemos asumir que nos encontramos en un nivel óptimo para afrontar las nuevas propuestas de mercado.

Durante el transcurso del proyecto, las numerosas ocasiones en las que nos hemos reunido con los clientes, han ido propiciando la aparición de nuevos requisitos y funcionalidades que se han ido recopilando y que en un futuro próximo, evaluaremos con la intención de producir nuevas herramientas que poner a la disposición de los clientes.

Concluyo a título personal, que la experiencia acaecida a lo largo del tiempo dedicado a este proyecto, ha sido más que satisfactoria; habiendo alcanzado no sólo las metas personales (como la integración en el equipo de toma de decisiones), sino también las profesionales; habiendo logrado el desarrollo completo de las especificaciones definidas en la especificación de la aplicación.

*Cuando algunos dicen “el saber no ocupa lugar”,  
yo no coincido, ya que a algo tienes que renunciar,  
porque cuando en el mercado te pretendes iniciar,  
la primera pregunta es ¿cuánto has dedicado a trabajar?*

*Por eso yo recomiendo; el trabajo y estudio compaginar,  
para que al terminar los estudios, a esa pregunta podamos contestar,  
y tal y como está el mercado hoy, no nos la vamos a jugar.*

*Y para terminar, despido este contenido,  
incidiendo, en lo que en la carrera he aprendido,  
fruto de la paciencia, el esfuerzo y de lo vivido.*

*Y así mismo, con empeño, ruego al olvido,  
que no cuente conmigo, para llevarse lo retenido,  
al menos, hasta que yo mismo, lo haya decidido.*

## 8 Referencias.

- IDC Worldwide Mobile Phone Tracker. (7 de 8 de 2013). *Apple Cedes Market Share in Smartphone Operating System Market as Android Surges and Windows Phone Gains, According to IDC*. Recuperado el 14 de 9 de 2013, de Smartphone Operating System Highlights: <http://www.idc.com/getdoc.jsp?containerId=prUS24257413>
- Android Developers. (s.f.). *The official site for Android developers*. Recuperado el 15 de 09 de 2013, de <http://developer.android.com/develop/index.html>
- EU. (2002). *TITULO De la pagina*. Recuperado el 2013, de Comisión europea: <http://www.upv.es>
- EUROPA, E. E. (2013). *Waste indicators on generation and landfilling - monitoring sustainable development*. Recuperado el 08 de 2013, de <http://epp.eurostat.ec.europa.eu/portal/page/portal/waste/data/wastestreams/elvs>
- EUROPE DIRECT. (04 de 03 de 2009). *Directive on waste*. Recuperado el 09 de 2013, de [http://europa.eu/legislation\\_summaries/environment/waste\\_management/ev0010\\_en.htm](http://europa.eu/legislation_summaries/environment/waste_management/ev0010_en.htm)
- Google Project . (15 de 6 de 2013). *ZXing: Multi-format 1D/2D barcode image processing library with clients for Android*. Recuperado el 15 de 9 de 2013, de <http://code.google.com/p/zxing/>
- PARLAMENTO EUROPEO Y DEL CONSEJO. (01 de 07 de 2005). *El acceso al Derecho de la Unión Europea* . Recuperado el 09 de 2013, de <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CONSLEG:2000L0053:20050701:ES:PDF>
- ROOTS INFOCOMM LTD. (s.f.). *Software Development Life Cycle*. Recuperado el 15 de 09 de 2013
- SIGRAUTO: Asociación Española para el tratamiento medioambiental de los vehículos fuera de uso. (04 de 09 de 2013). *Página web de SIGRAUTO*. Recuperado el 10 de 09 de 2013, de <http://www.sigrauto.com/redlis.html>
- Wikipedia. (26 de 7 de 2013). *Eclipse (software) - Wikipedia, la enciclopedia libre*. Recuperado el 14 de 09 de 2013, de [http://es.wikipedia.org/wiki/Eclipse\\_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software))
- Wikipedia. (17 de 7 de 2013). *Xamarin - Wikipedia, la enciclopedia libre*. Recuperado el 14 de 9 de 2013, de <http://es.wikipedia.org/wiki/Xamarin>

