



# **GESTIÓN DEL SAT DE UN TALLER CON ANDROID.**

*Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles*

**Alumno: Raúl Roca Sanz**

**Consultor: Francesc D'Assís Queralt Giralt**

**7/6/2017**

© (Raúl Roca Sanz)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>Gestión del SAT de un taller con Android</i>
<b>Nombre del autor:</b>	<i>Raúl Roca Sanz</i>
<b>Nombre del consultor:</b>	<i>Francesc D'Assís Queralt Giralt</i>
<b>Fecha de entrega (mm/aaaa):</b>	<i>06/2017</i>
<b>Titulación:</b>	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>
<b>Resumen del Trabajo (máximo 250 palabras):</b>	
<p>Con la finalidad de facilitar la gestión a los trabajadores de un servicio de asistencia técnica (SAT) se presenta una aplicación desarrollada para dispositivos Android. Con el uso de esta aplicación los trabajadores podrán prescindir de sus PCs para actualizar la ficha de los artículos que se encuentran en reparación.</p> <p>Sus funciones principales serán las de dar de alta artículos en los que se podrán incluir fotografías, descripción, observaciones del técnico, datos del cliente y fecha de entrada en taller. También dispondrán de un buscador para encontrar un artículo que ya se ha dado de alta, una vez encontrado podrán editar los valores de sus atributos para mantener la ficha actualizada. Todo ello se almacenará en una base de datos a la que podrá acceder cualquier usuario registrado y validado por el administrador.</p>	
<b>Abstract (in English, 250 words or less):</b>	
<p>In order to facilitate the management of a technical assistance service, an application developed for Android devices is presented. The workers can dispense their PCs to update the data of the articles that are under repair with the use of this application.</p> <p>Its main functions will be to enter articles that may include pictures, description, observations of the technician, customer data and date of entry into the commerce. They will have a search engine to find an article that has already been registered. They can edit the values of their</p>	

attributes to update the database. All data will be stored in a database that can be accessed by any user registered and validated by the administrator.

**Palabras clave (entre 4 y 8):**

Gestión SAT, Android, reparaciones.

# Índice

1	Introducción.....	1
1.1	Contexto y justificación del trabajo.....	1
1.2	Objetivos del trabajo.....	3
1.3	Enfoque y método elegido.....	4
1.4	Planificación del trabajo .....	5
1.5	Breve resumen del trabajo obtenido. ....	7
1.6	Breve descripción de otros capítulos de la memoria.....	7
2	Usuarios y contexto de uso .....	8
2.1	Investigación.....	8
2.2	Perfiles de usuario.....	12
2.3	Contexto de uso .....	12
2.4	Análisis de tareas .....	13
3	Diseño conceptual .....	14
3.1	Escenarios de uso .....	14
3.2	Diagrama de flujo .....	16
4	Prototipado.....	17
4.1	Sketches.....	17
4.2	Prototipo alta fidelidad .....	18
4.2.1	Login.....	18
4.2.2	Registro .....	19
4.2.3	Menú.....	19
4.2.4	Alta artículo .....	20
4.2.5	Añadir imagen .....	21
4.2.6	Buscar artículo .....	21
4.2.7	Editar artículo encontrado .....	22
5	Evaluación .....	23
6	Definición casos de uso .....	24
6.1	Diagrama UML.....	24
6.2	Casos de uso .....	25
7	Diseño de la arquitectura .....	27
8	Implementación.....	29
8.1	Servidor backend.....	29
8.2	Estructura de la aplicación .....	29
8.2.1	Actividades de carpeta com.edu.uoc.gestionsat .....	29
8.2.2	Carpeta res .....	30
8.3	Modificaciones en diseño de la aplicación .....	31
8.3.1	Pantalla alta de nuevo artículo.....	31
8.3.2	Pantalla búsqueda y edición de artículo .....	32
9	Pruebas .....	34
9.1	Entorno de pruebas.....	34
9.2	Pruebas realizadas .....	34
10	Conclusiones.....	37
11	Bibliografía.....	38
12	Anexos .....	39
12.1	Instalación .....	39
12.2	Manual de uso.....	39
12.2.1	Login .....	39
12.2.2	Registro.....	40

<b>12.2.3-Menú .....</b>	<b>40</b>
<b>12.2.4-Nuevo artículo.....</b>	<b>41</b>
<b>12.2.5-Buscar artículo.....</b>	<b>41</b>
<b>12.2.6-Editar artículo.....</b>	<b>43</b>
<b>12.2.7-Cerrar sesión.....</b>	<b>44</b>

## Lista de figuras

Ilustración 1. SimplyGest SAT móvil .....	2
Ilustración 2. Información IDE Android Studio .....	4
Ilustración 3. Diagrama de Gantt .....	6
Ilustración 4. Diagrama de flujo .....	16
Ilustración 5. Sketches .....	17
Ilustración 6. Prototipo login .....	18
Ilustración 7. Prototipo registro .....	19
Ilustración 8. Prototipo menú .....	19
Ilustración 9. Prototipo alta artículo .....	20
Ilustración 10. Prototipo añadir imagen .....	21
Ilustración 11. Prototipo buscar artículo .....	21
Ilustración 12. Prototipo editar artículo .....	22
Ilustración 13. Diagrama casos de uso .....	24
Ilustración 14. Diagrama explicativo del sistema .....	27
Ilustración 15. Entidades y atributos .....	28
Ilustración 16. Diagrama UML entidades y clases .....	28
Ilustración 17. Estructura carpeta java .....	30
Ilustración 18. Estructura de res .....	30
Ilustración 19. Prototipo y resultado de pantalla artículo nuevo .....	31
Ilustración 20. Prototipo de buscar .....	32
Ilustración 21. Prototipo de editar .....	32
Ilustración 22. Resultado buscar y editar con un resultado encontrado .....	32
Ilustración 23. Resultado buscar con varios resultados .....	33
Ilustración 24. Versiones emuladas .....	34
Ilustración 25. Pruebas unitarias .....	34
Ilustración 26. Pantalla inicio .....	39
Ilustración 27. Pantalla registro .....	40
Ilustración 28. Pantalla menú .....	40
Ilustración 29. Pantalla artículo nuevo .....	41
Ilustración 30. Pantalla buscar .....	42
Ilustración 31. Pantalla artículo encontrado .....	42
Ilustración 32. Pantalla para varios artículos encontrados .....	43
Ilustración 33. Pantalla editar artículo .....	43
Ilustración 34. Cerrar sesión .....	44

## Lista de tablas

Tabla 1. Escenario de uso 1 .....	14
Tabla 2. Escenario de uso 2 .....	15
Tabla 3. Escenario de uso 3 .....	15
Tabla 4. Caso de uso 1 .....	25
Tabla 5. Caso de uso 2 .....	25
Tabla 6. Caso de uso 3 .....	25
Tabla 7. Caso de uso 4 .....	26
Tabla 8. Caso de uso 5 .....	26
Tabla 9. Casos de prueba .....	36

# 1-Introducción

## 1.1-Contexto y justificación del trabajo.

Debido al alto volumen de trabajo que se acumula en un taller o SAT, se hace necesario tener una gestión mediante la utilización de herramientas informáticas que nos ayuden a tener el control de todos los datos que se almacenan (inventario de artículos, datos de clientes, estado de las reparaciones, datos de proveedores, etc.).

Es muy habitual utilizar un PC con acceso a una base de datos donde se almacenan todos los datos relevantes sobre el negocio. Este caso, siendo el más habitual, hace que el técnico o el gestor de este negocio dependan de un PC con conexión a internet en caso de que la base de datos esté disponible online para hacer las consultas necesarias.

El objetivo de este trabajo será el de desarrollar una aplicación que dote al técnico de una aplicación con la que gestionar sus trabajos con total movilidad y sin la necesidad de necesitar ningún PC con el que actualizar los datos de la base de datos.

Si analizamos las opciones disponibles podemos encontrar multitud de aplicaciones para un entorno que se centra en el uso de un PC. Sin embargo, al estudiar el de las aplicaciones para móviles observamos que en el caso de Android no hay ninguna aplicación que cubra esta necesidad de una forma eficiente.

En el caso de iOS si que disponemos de una opción que se podría ajustar a nuestras necesidades. Este es el caso de [SimplyGest Sat Móvil](#), aplicación que podemos encontrar en la App Store, donde podremos descargarla en caso de disponer de un iPhone.





Ilustración 1. SimplyGest SAT móvil

Con esta aplicación los técnicos del SAT pueden llevar un control sobre el estado de las incidencias que se les envían. El funcionamiento es muy similar al propuesto en nuestra idea. Sin embargo, en nuestro caso pretendemos añadirle mejoras como las imágenes del artículo reparado, o la descripción entre otras. Como el objetivo de nuestro proyecto es cubrir una necesidad y en iOS ya se cubre, nos centraremos en el desarrollo de una nueva aplicación sobre la plataforma Android.

## **1.2-Objetivos del trabajo.**

El fin de este proyecto es conseguir una aplicación funcional para lo que se tendrán que superar los diferentes puntos:

- Encontrar una necesidad que pueda ser satisfecha mediante una aplicación para un smartphone.
- Controlar los plazos de entrega mediante una planificación correcta.
- Realizar un análisis de las opciones disponibles en el mercado para observar que funciones cubren y cuales pueden añadirse o ser mejoradas.
- Realizar un pequeño estudio sobre los usuarios potenciales y sus necesidades para mejorar el diseño.
- Realizar el diseño de los requisitos más importantes para nuestra aplicación, centrándonos en los puntos imprescindibles para que sea realmente útil y posibilitando una mejora posterior.
- Lograr una aplicación que pueda resultar interesante para los usuarios y que sea totalmente funcional.
- Mejorar los conocimientos adquiridos a lo largo del máster en la plataforma elegida, en este caso Android.

Por último, se adjunta una lista con los objetivos que cubrirá nuestra aplicación:

- Conexión a base de datos.
- Login de usuario en la aplicación.
- Añadir artículo a reparar.
- Añadir imagen del artículo.
- Añadir información y estado de la avería.
- Añadir información de la reparación.
- Datos del propietario del artículo a reparar.
- Editar campos anteriores.

### 1.3-Enfoque y método elegido.

Como se observó en el estudio de mercado, no hay aplicaciones similares en Android por lo que se realizará una aplicación nueva partiendo desde cero. Para desarrollarla utilizaremos la API 14 de Android que garantiza compatibilidad con el 97.4% de los dispositivos Android del mercado, tal y como extraemos del IDE Android Studio y de la [documentación oficial de Android referente al uso relativo de plataformas de este sistema.](#)

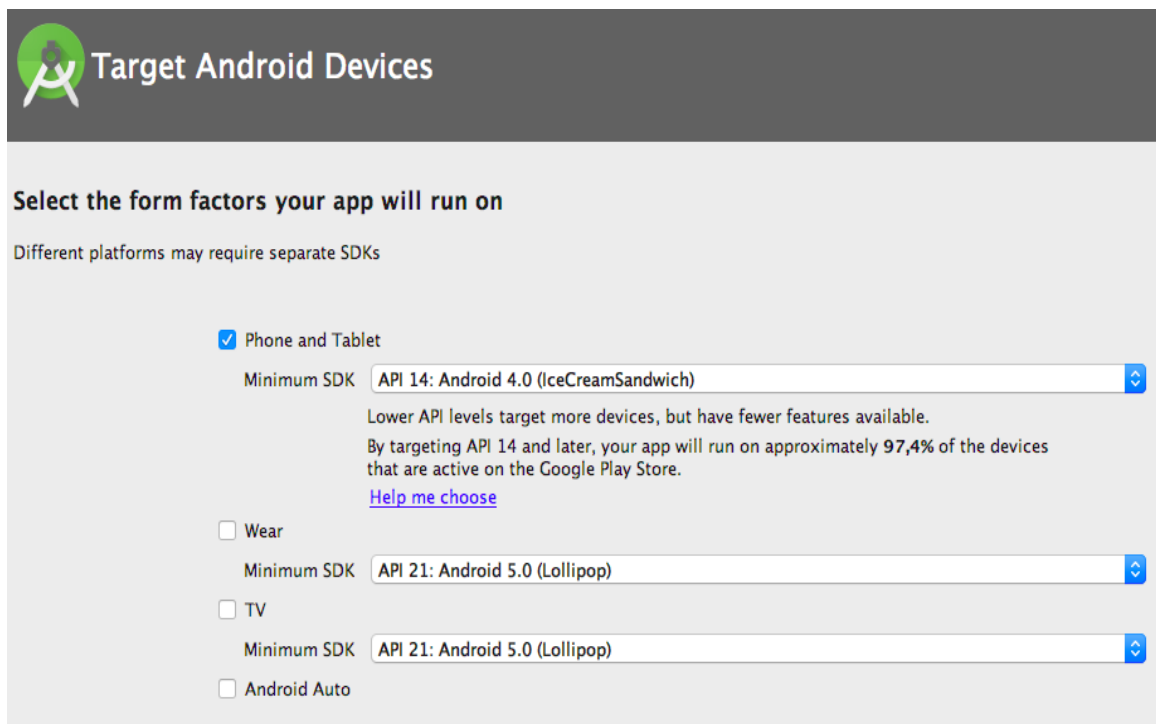


Ilustración 2. Información IDE Android Studio

Al ser un desarrollo nuevo y debido al alto grado de cambios y mejoras que puede tener la aplicación se recurrirá a una metodología ágil. Por último, se establecerán las siguientes fases para su desarrollo. Primero se realizara el diseño de la interfaz de usuario de la aplicación y de las pantallas de navegación. Después nos centraremos en el uso de la base de datos donde se almacenarán los datos con todas las opciones propuestas. Finalmente, se realizarán las pruebas y las mejoras necesarias para hacer la aplicación útil e intuitiva.

## 1.4-Planificación del trabajo.

Para el desarrollo de este proyecto se van a utilizar los siguientes recursos:

- Android Studio 2.3, que es el IDE oficial para el desarrollo de aplicaciones Android.
- PC de sobremesa con Windows 10 donde se realizará el desarrollo de la aplicación con el IDE anterior.
- Smartphone Xiaomi Redmi Note 2, para realizar las pruebas fuera del emulador ofrecido por Android Studio.
- GIMP 2.8.18 para crear imágenes personalizadas de la aplicación.
- Backendless para guardar y gestionar nuestra base de datos online.
- Balsamiq mockups 3 para generar el prototipo de alta fidelidad.
- Word 2013 para la edición de documentos.

A continuación se detallan las tareas a realizar a partir de la PEC 2 con la estimación de las horas dedicadas a ellas:

- Perfil de usuario y requisitos -15 horas.
- Prototipo baja fidelidad - 12 horas.
- Evaluar prototipo básico- 3 horas
- Prototipo alta fidelidad- 6horas.
- Diseñar la arquitectura de la aplicación - 27 horas.
- Configurar Backendless- 3horas
- Desarrollo de login - 9 horas
- Desarrollo de añadir pieza - 15 horas
- Desarrollo de editar pieza - 9horas
- Desarrollo de estado pieza -9horas
- Desarrollo de imagen pieza - 12 horas
- Desarrollo datos cliente - 9 horas
- Navegación - 6 horas
- Mejoras - 21 horas
- Pruebas producto final - 12 horas.
- Documentación - 45 horas
- Entregas - 30 horas

Todas estas tareas se llevarán a cabo mediante la siguiente distribución horaria:

- 3 horas de lunes a domingos, desde el inicio de la PEC 2 hasta la entrega final del proyecto.
- Domingo 19, Lunes 20 se descansará por motivo de las fiestas patronales de Castellón.

Podemos observar la planificación de las tareas en el siguiente diagrama de Gantt:

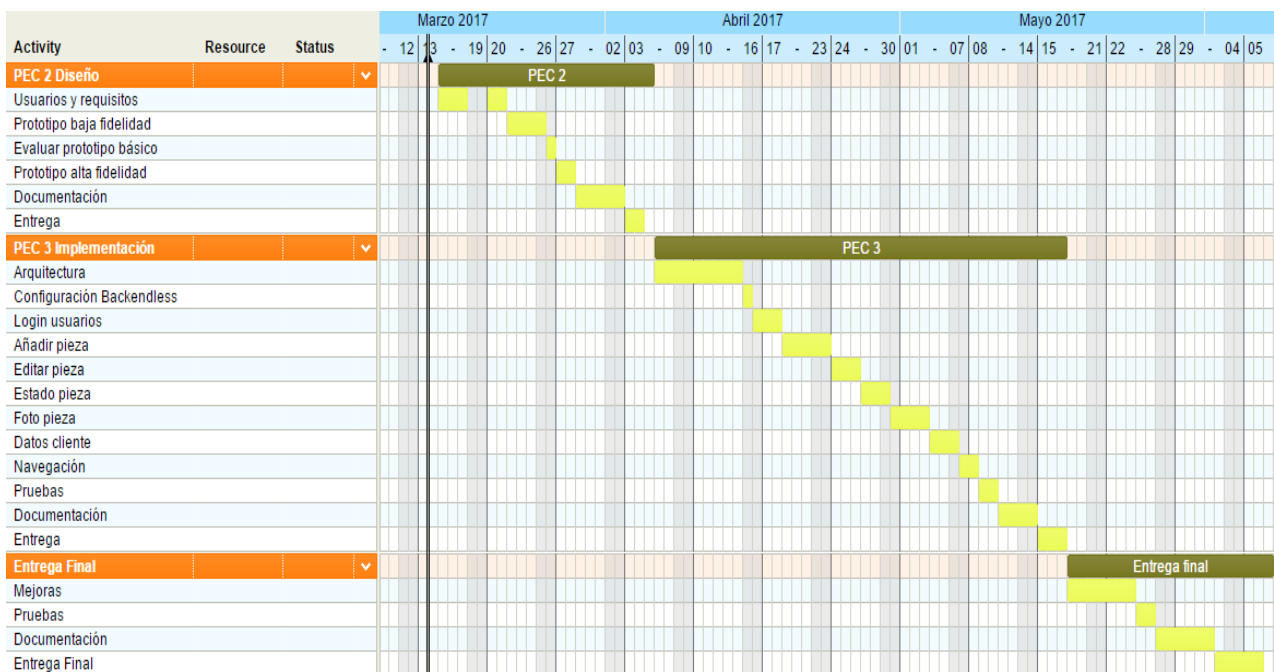


Ilustración 3. Diagrama de Gantt

## **1.5-Breve resumen del trabajo obtenido.**

Como se observa en la planificación del proyecto, el trabajo se estructurará en cuatro entregables, los cuales pasan a describirse a continuación:

1. La primera entrega constará de una descripción de nuestra idea. En ella se describirán los objetivos, el enfoque, el método elegido y la planificación de las tareas.
2. Esta entrega se centrará en el diseño de la aplicación. En ella se tratará el estudio de los usuarios y los requisitos deseados. El diseño de las pantallas de nuestra aplicación y la navegación. Además, se creará el prototipo de esta.
3. Partiendo del diseño de la aplicación se abordará su desarrollo. Nos centraremos en la creación y las pruebas de esta. Con esto obtendremos el código fuente de nuestra aplicación.
4. En la última entrega se aplicarán algunas mejoras al producto en caso de ser necesario. Nos encargaremos de generar el ejecutable con extensión .apk (Android Application Package) para la instalación en terminales Android. También se documentará todo lo realizado.

## **1.6-Breve descripción de otros capítulos de la memoria.**

En los próximos capítulos de la memoria se describirán los siguientes puntos:

- El análisis de la aplicación usando un diseño centrado en el usuario.
- Descripción de los puntos más importantes de la implementación.
- El resultado de las pruebas realizadas sobre la aplicación.
- Las conclusiones obtenidas con la elaboración de este proyecto.

## 2-Usuarios y contexto de uso

En este capítulo se analizarán quienes son nuestros usuarios potenciales así como sus necesidades, objetivos y el contexto de uso de la aplicación. De esta manera se detectarán las funcionalidades que deberá incluir la aplicación para cubrir estos puntos.

### 2.1-Investigación

Para recopilar información se realizarán varias entrevistas a un grupo de usuarios potenciales. En este caso la aplicación se centra en un segmento de usuarios específicos como son los trabajadores de un taller. En las entrevistas se recopilarán datos personales. Se han considerado relevantes los siguientes datos:

- Edad
- Nivel de estudios
- Puesto en la empresa
- Experiencia en el puesto

También se plantearán cuestiones sobre la utilidad de su sistema actual, su uso del smartphone y su punto de vista sobre nuestra idea. A continuación se muestran las cuestiones:

- ¿Dispone de smartphone?
- ¿Qué sistema operativo utiliza su smartphone?
- ¿Está familiarizado con el uso de aplicaciones móviles?
- ¿Qué nivel de conocimientos informáticos tiene?
- ¿Qué opina de su sistema de gestión del taller?
- ¿Le parece interesante el uso de una aplicación móvil para realizar la gestión de las averías?
- ¿Le parecen suficientes las opciones disponibles (alta artículo, buscador, edición de ficha)? En caso negativo, indicar que añadiría.

Una vez planteadas nuestras cuestiones se realizará la entrevista a un grupo de 6 usuarios potenciales, 4 técnicos que se encargan de las reparaciones del taller y 2 recepcionistas encargadas de tramitar la entrada de artículos averiados. Como resultado de estas entrevistas se obtienen los siguientes datos:

## Usuario 1

### Datos personales.

**Edad:** 31

**Nivel de estudios:** Ingeniería en electrónica.

**Puesto en la empresa:** Técnico de reparaciones.

**Experiencia en el puesto:** 4 años en el SAT.

### Preguntas.

**Dispone de smartphone:** Si

**¿Qué sistema operativo utiliza su smartphone?** iOS

**¿Está familiarizado con el uso de aplicaciones móviles?** Si

**¿Qué nivel de conocimientos informáticos tiene?** Medio

**¿Qué opina de su sistema de gestión del taller?** Bastante útil pero me limita a depender de un PC.

**¿Le parece interesante el uso de una aplicación móvil para realizar la gestión de las averías?** Muy interesante.

**¿Le parecen suficientes las opciones disponibles en su aplicación actual?** En caso negativo, indicar que añadiría. Si

## Usuario 2

### Datos personales.

**Edad:** 55

**Nivel de estudios:** Formación profesional superior en electrónica.

**Puesto en la empresa:** Técnico de reparaciones.

**Experiencia en el puesto:** 35 años.

### Preguntas.

**Dispone de smartphone:** Si.

**¿Qué sistema operativo utiliza su smartphone?** Android.

**¿Está familiarizado con el uso de aplicaciones móviles?** Si.

**¿Qué nivel de conocimientos informáticos tiene?** Medio.

**¿Qué opina de su sistema de gestión del taller?** Muy útil, no lo cambiaría.

**¿Le parece interesante el uso de una aplicación móvil para realizar la gestión de las averías?** Si. Podría dar de alta artículo cuando me desplazo a algún domicilio.

**¿Le parecen suficientes las opciones disponibles en su aplicación actual?** En caso negativo, indicar que añadiría. Si



### Usuario 3

#### **Datos personales.**

**Edad:** 21.

**Nivel de estudios:** Formación profesional superior en electrónica.

**Puesto en la empresa:** Técnico de reparaciones.

**Experiencia en el puesto:** Estancia en prácticas.

#### **Preguntas.**

**Dispone de smartphone:** Si.

**¿Qué sistema operativo utiliza su smartphone?** Android.

**¿Está familiarizado con el uso de aplicaciones móviles?** Si

**¿Qué nivel de conocimientos informáticos tiene?** Medio.

**¿Qué opina de su sistema de gestión del taller?** Muy bueno, me ayuda con el control de las tareas pendientes.

**¿Le parece interesante el uso de una aplicación móvil para realizar la gestión de las averías?** Si, en el taller solo se disponen de dos equipos de sobremesa y pueden estar ocupados.

**¿Le parecen suficientes las opciones disponibles en su aplicación actual?** En caso negativo, indicar que añadiría. Si

### Usuario 4

#### **Datos personales.**

**Edad:** 30

**Nivel de estudios:** Ingeniería electrónica.

**Puesto en la empresa:** Técnico de reparaciones.

**Experiencia en el puesto:** 2 años.

#### **Preguntas.**

**Dispone de smartphone:** Si.

**¿Qué sistema operativo utiliza su smartphone?** Android.

**¿Está familiarizado con el uso de aplicaciones móviles?** Si.

**¿Qué nivel de conocimientos informáticos tiene?** Altos.

**¿Qué opina de su sistema de gestión del taller?** Muy útil.

**¿Le parece interesante el uso de una aplicación móvil para realizar la gestión de las averías?** Mucho, es una forma de no depender de un PC para actualizar las fichas de los artículos.

**¿Le parecen suficientes las opciones disponibles en su aplicación actual?** En caso negativo, indicar que añadiría. Si

## Usuario 5

### **Datos personales.**

**Edad:** 50.

**Nivel de estudios:** Bachillerato.

**Puesto en la empresa:** Recepcionista.

**Experiencia en el puesto:** 30 años.

### **Preguntas.**

**Dispone de smartphone:** Si

**¿Qué sistema operativo utiliza su smartphone?** Android.

**¿Está familiarizado con el uso de aplicaciones móviles?** Si.

**¿Qué nivel de conocimientos informáticos tiene?** Bajo.

**¿Qué opina de su sistema de gestión del taller?** Correcto, me ayuda con la gestión.

**¿Le parece interesante el uso de una aplicación móvil para realizar la gestión de las averías?** Siempre que no sea demasiado complicado.

**¿Le parecen suficientes las opciones disponibles en su aplicación actual?** En caso negativo, indicar que añadiría. Si

## Usuario 6

### **Datos personales.**

**Edad:** 30

**Nivel de estudios:** ESO.

**Puesto en la empresa:** Recepcionista.

**Experiencia en el puesto:** 1 año.

### **Preguntas.**

**Dispone de smartphone:** Si.

**¿Qué sistema operativo utiliza su smartphone?** Android.

**¿Está familiarizado con el uso de aplicaciones móviles?** Si.

**¿Qué nivel de conocimientos informáticos tiene?** Bajo.

**¿Qué opina de su sistema de gestión del taller?** Útil para mis tareas diarias.

**¿Le parece interesante el uso de una aplicación móvil para realizar la gestión de las averías?** Si.

**¿Le parecen suficientes las opciones disponibles en su aplicación actual?** En caso negativo, indicar que añadiría. Si

## 2.2-Perfiles de usuario

Como podemos observar en base a los resultados de nuestras entrevistas, encontramos dos perfiles de usuarios que pueden utilizar nuestra aplicación:

Recepcionista: Este perfil tiene una formación en los estudios menor, además suelen tener un nivel más bajo en informática. Aun así, al disponer de un smartphone con sistema operativo Android y estar familiarizado con estas aplicaciones, el salto a esta tecnología no les supondrá mucho problema. Este grupo se encargará sobre todo en dar de alta un artículo en la recepción del taller.

Técnico: Dentro de este grupo se observa que la formación académica es mayor y más específica. Este grupo tiene un nivel más alto en informática y también dispone de un smartphone por lo que tampoco tendrá problemas con el uso de la aplicación. Principalmente se encargarán de actualizar el estado del artículo cuando terminen de realizar las tareas pertinentes sobre él. También pueden desplazarse al domicilio de un cliente y tramitar el alta de artículos.

## 2.3-Contexto de uso

En este caso, para el uso de la aplicación será necesaria la utilización de un smartphone con un sistema operativo Android y conexión a internet. A nivel de competencias, para utilizarla bastará con tener conocimientos mínimos sobre el uso de aplicaciones en Android.

El uso habitual de la aplicación será el de consulta o modificación de estado de algún artículo durante la jornada laboral del personal del taller. Así pues, podemos diferenciar entre tres contextos de uso:

1. Recepcionista del taller hace uso de la aplicación durante su jornada para dar de alta un artículo, en caso excepcional puede desear modificar algún campo del artículo ya dado de alta. Este proceso lo puede realizar en la recepción del taller.
2. Técnico del taller hace uso de la aplicación durante su jornada para actualizar algunos de los campos relacionados con el artículo sobre el que está trabajando. Este proceso lo puede realizar en cualquier punto del taller.

3. Técnico del taller se desplaza al domicilio de un cliente para realizar una reparación de un artículo o la recogida de este. En este caso el proceso se llevará a cabo fuera del taller.

## 2.4-Análisis de tareas

Siguiendo con nuestro análisis veremos cuáles serán las tareas que podrán realizar nuestros usuarios:

- Registrar usuario.  
Como primer paso nuestros usuarios deberán registrarse en la aplicación para poder usarla y acceder a la base de datos donde se almacenarán todos los artículos.
- Autenticarse como personal del taller.  
Un usuario podrá introducir su usuario y contraseña con los que identificarse como miembro del taller para realizar las búsquedas y modificaciones oportunas de la ficha de un artículo.
- Añadir un nuevo artículo.  
Al dar de alta un nuevo artículo se solicitarán datos como una descripción del problema, una fotografía, los datos del propietario y la fecha de entrada. Todos estos datos se almacenarán en la base de datos.
- Consultar el estado de un artículo.  
Al usuario se le presentará un buscador donde localizará y visualizará los artículos para consultar el estado de las reparaciones.
- Actualizar el estado de un artículo.  
El usuario podrá actualizar los datos del artículo, estado de la reparación, imagen, descripción, notas del técnico sobre la reparación.
- Terminar sesión.  
En este caso el usuario podrá cerrar la sesión al terminar su jornada laboral para que en caso de ser un terminal propiedad de la empresa el resto de usuarios no puedan acceder con sus credenciales.

## 3-Diseño conceptual

### 3.1-Escenarios de uso

En este punto veremos los diferentes escenarios de uso que se pueden dar:

Los dos primeros escenarios serán los relacionados con los técnicos.

Escenario 1	
<b>Perfil</b>	Técnico.
<b>Personaje</b>	Juan Carlos Andreu de 55 años es técnico en electrónica y es un trabajador del SAT.
<b>Descripción</b>	Juan Carlos al comenzar con la reparación de un artículo se dispone a buscar su ficha para ver el estado actual y realizar las modificaciones pertinentes sobre sus datos.
<b>Contexto</b>	Puesto trabajo.
<b>Tareas</b>	Autenticarse, buscar artículo en base de datos, modificar datos.
<b>Objetivo</b>	Gestión ficha del artículo reparado.
<b>Necesidad de información</b>	Código seguimiento artículo. Descripción.
<b>Funcionalidades</b>	Login. Buscar artículo. Editar artículo.
<b>Desarrollo tareas</b>	El usuario se identifica, busca el artículo y modifica el estado en caso de ser necesario.

Tabla 1. Escenario de uso 1

Escenario 2	
<b>Perfil</b>	Técnico.
<b>Personaje</b>	Alexis Pallarés de 31 años es ingeniero en electrónica y es un trabajador del SAT.
<b>Descripción</b>	Alexis se desplaza al domicilio de un cliente para tramitar una avería. Si no es posible su reparación recogerá el artículo y lo llevara al taller.
<b>Contexto</b>	Domicilio cliente.
<b>Tareas</b>	Autenticarse, dar de alta el artículo en la base de datos.
<b>Objetivo</b>	Alta de un artículo para su reparación en taller.
<b>Necesidad de</b>	Datos de cliente, imagen, descripción del problema,

<b>información</b>	fecha del problema.
<b>Funcionalidades</b>	Login. Dar de alta el artículo.
<b>Desarrollo tareas</b>	El usuario se identifica y da de alta el artículo.

Tabla 2. Escenario de uso 2

Por otro lado tenemos el escenario relacionado con recepción.

<b>Escenario 3</b>	
<b>Perfil</b>	Recepcionista.
<b>Personaje</b>	María José de 50 años tiene estudios de bachillerato y trabaja como recepcionista del SAT.
<b>Descripción</b>	Al estar en su puesto recibe un cliente con un artículo averiado. Ella se encargará de darlo de alta asignándole un número de seguimiento, añadiendo los datos del cliente, tomando una fotografía, la fecha de entrada y por último la descripción del problema.
<b>Contexto</b>	Puesto de trabajo.
<b>Tareas</b>	Autenticarse, dar de alta el artículo.
<b>Objetivo</b>	Dar de alta artículo.
<b>Necesidad de información</b>	Datos de cliente, imagen, descripción del problema, fecha del problema.
<b>Funcionalidades</b>	Login. Dar de alta artículo.
<b>Desarrollo tareas</b>	El usuario se identifica y da de alta el artículo.

Tabla 3. Escenario de uso 3

### 3.2-Diagrama de flujo

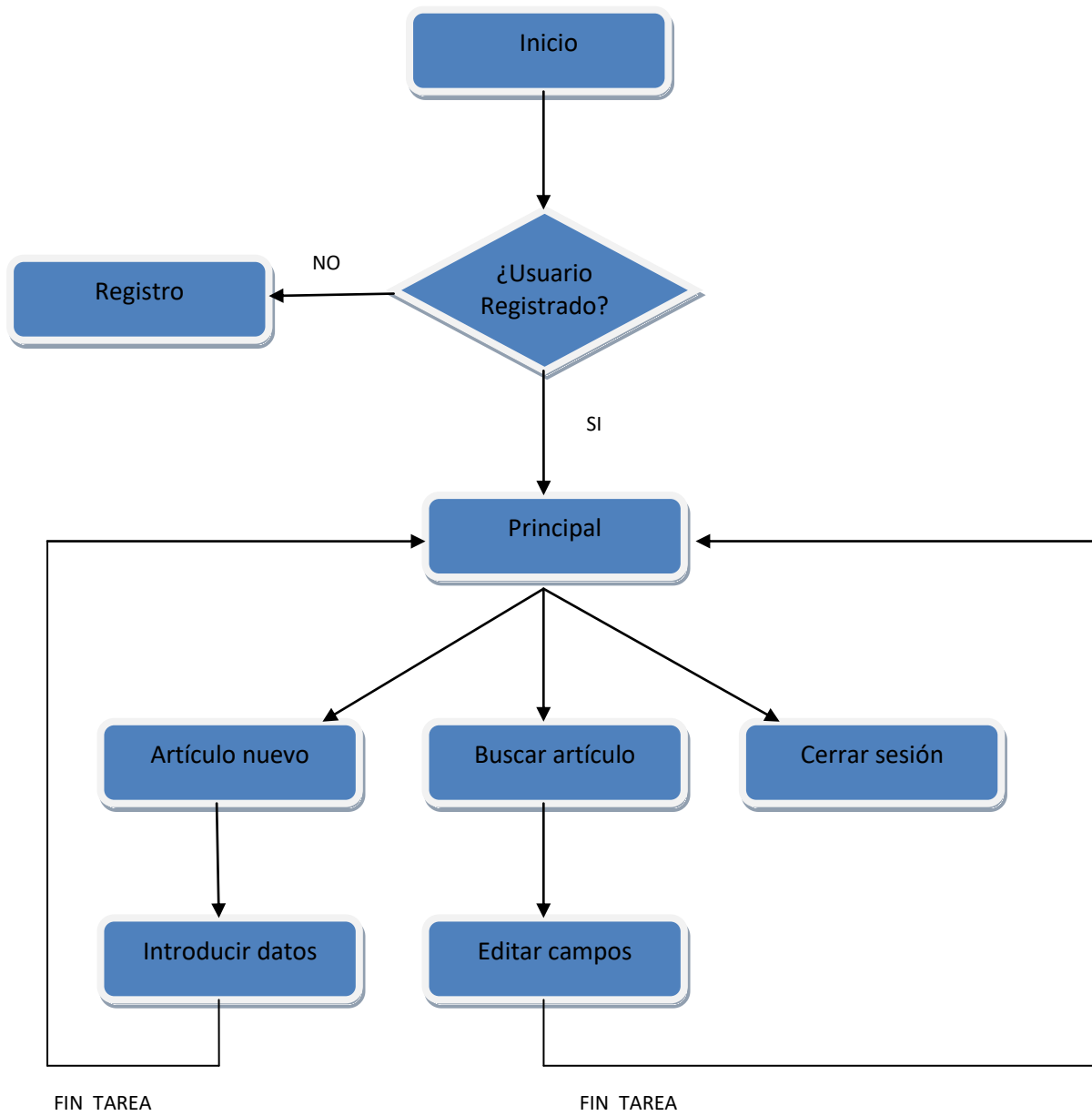


Ilustración 4. Diagrama de flujo

## 4-Prototipado

En este punto se mostrará una primera aproximación del aspecto visual de nuestra aplicación. Primero se optará por la presentación de los Sketches a mano alzada y finalmente se recurrirá a una versión que se ajusta más a la realidad, un prototipo de alta fidelidad.

### 4.1-Sketches

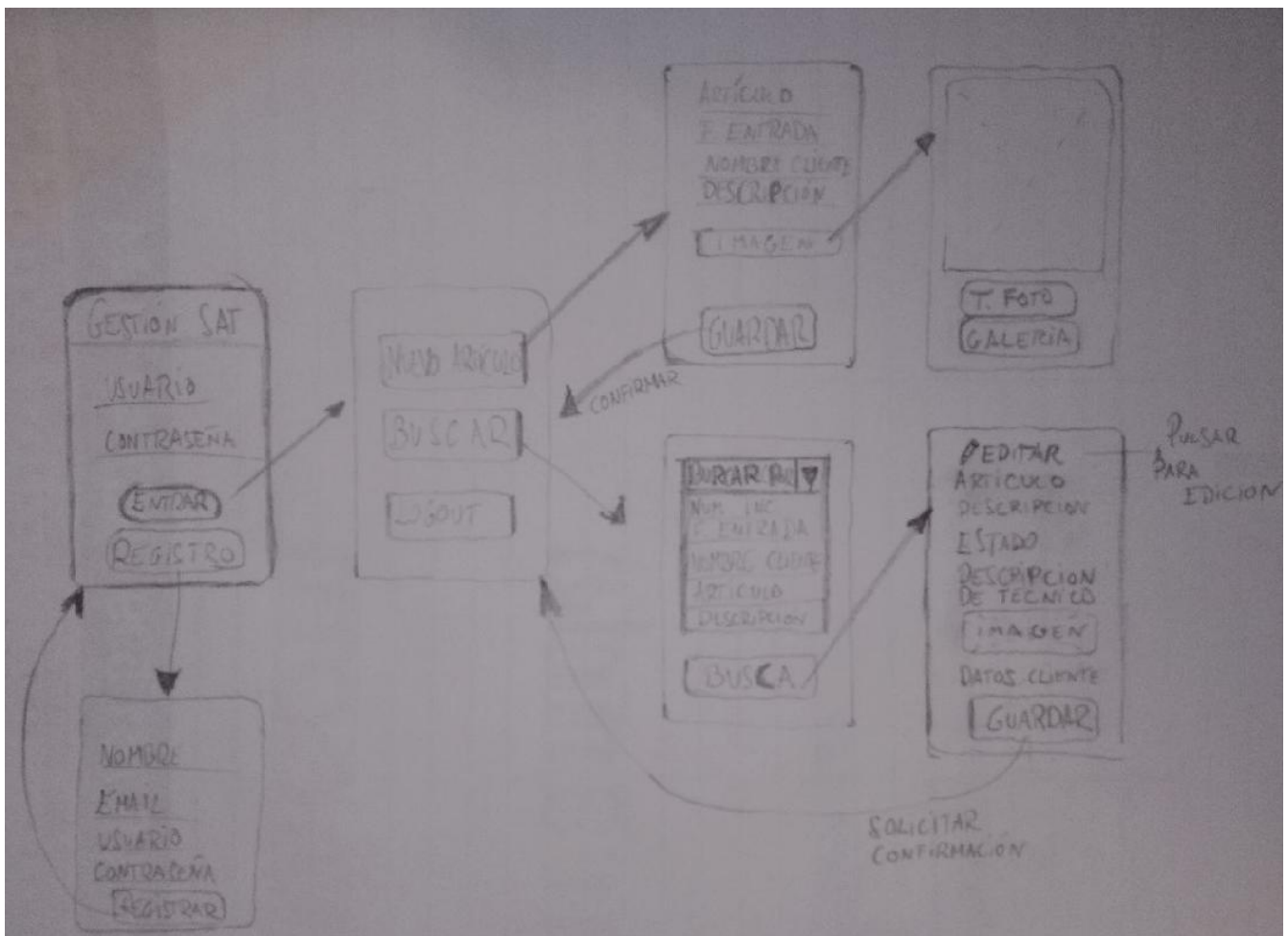


Ilustración 5.Sketches



## 4.2-Prototipo alta fidelidad

A continuación se presentará el prototipo horizontal en alta fidelidad, analizándolo por pantallas.

### 4.2.1 Login

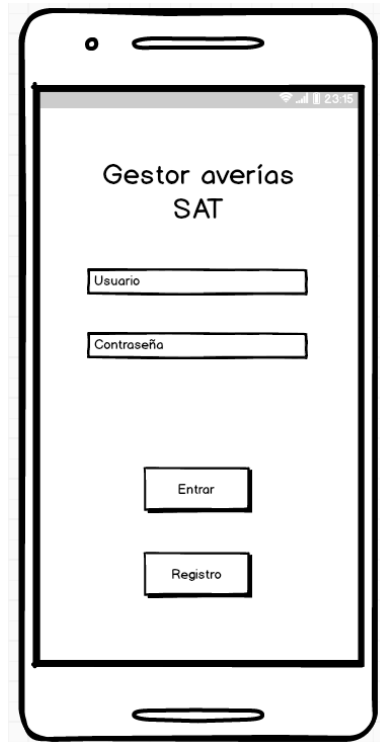


Ilustración 6. Prototipo login

Esta será la pantalla que aparecerá al abrir la aplicación, para acceder a ella el usuario necesitará identificarse utilizando su usuario y contraseña, pulsando después entrar para así acceder al sistema. Si no dispone de usuario pulsara en registro para proceder con su registro.

## 4.2.2 Registro

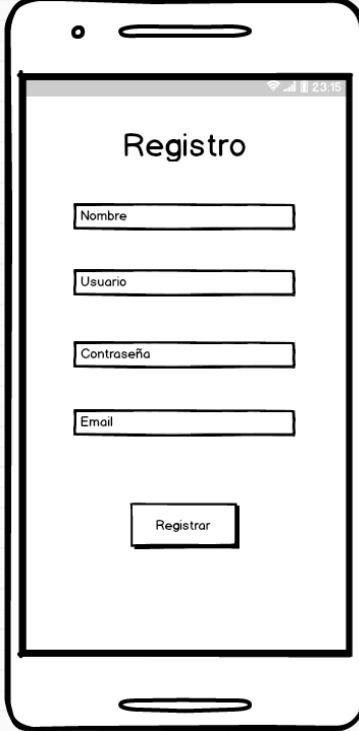


Ilustración de un prototipo de pantalla de registro en un teléfono móvil. La pantalla muestra el título "Registro" en el centro. Debajo del título hay cuatro campos de entrada de texto, uno por línea, etiquetados como "Nombre", "Usuario", "Contraseña" y "Email". Debajo de los campos hay un botón rectangular etiquetado como "Registrar". La pantalla también muestra un estado de la barra superior con el tiempo "23:15" y un estado de la barra inferior con un botón de inicio.

Ilustración 7. Prototipo registro

En la pantalla de registro el usuario podrá darse de alta en el sistema. Primero tendrá que rellenar los campos nombre, usuario, contraseña y email. Por último pulsará finalizar para que se registre en la base de datos.

## 4.2.3 Menú

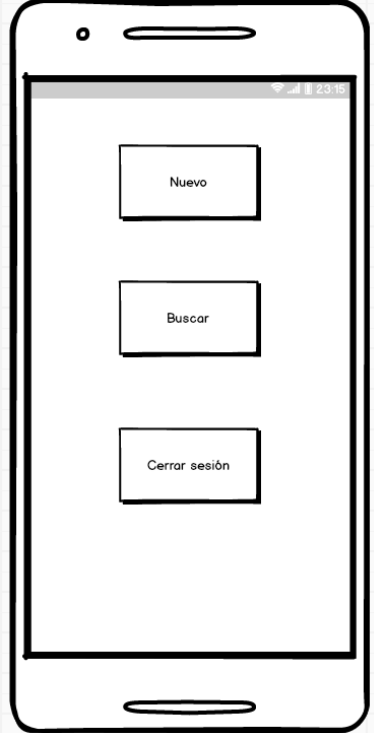


Ilustración de un prototipo de pantalla de menú en un teléfono móvil. La pantalla muestra tres botones rectangulares apilados verticalmente en el centro. Los botones están etiquetados como "Nuevo", "Buscar" y "Cerrar sesión". La pantalla también muestra un estado de la barra superior con el tiempo "23:15" y un estado de la barra inferior con un botón de inicio.

Ilustración 8. Prototipo menú

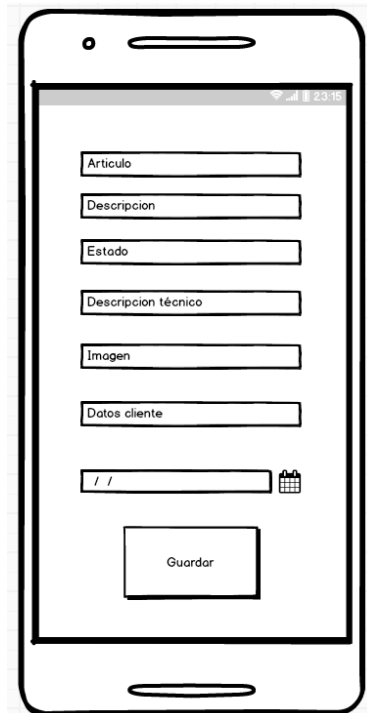
En la pantalla menú encontraremos las opciones disponibles en la aplicación.

-Nuevo. Los pulsaremos para dar de alta un artículo nuevo.

-Buscar. Al pulsarlo se accederá al buscador de artículos.

-Cerrar sesión. Se utilizará en caso de querer terminar la sesión y salir de la aplicación.

#### 4.2.4 Alta artículo

El prototipo muestra una interfaz de usuario para la creación de un artículo. El formulario está organizado verticalmente y contiene los siguientes elementos:

- Campo de texto etiquetado como "Artículo".
- Campo de texto etiquetado como "Descripción".
- Campo de texto etiquetado como "Estado".
- Campo de texto etiquetado como "Descripción técnico".
- Campo de texto etiquetado como "Imagen".
- Campo de texto etiquetado como "Datos cliente".
- Campo de texto con el formato " / / " y un icono de calendario a la derecha.
- Botón rectangular etiquetado como "Guardar".

Ilustración 9. Prototipo alta artículo

En la pantalla de alta de artículos se podrá registrar la entrada de estos en el taller. Se necesitará completar todos los campos expuestos como son artículo, descripción, estado de la reparación, tomar una fotografía del artículo, los datos del cliente y la fecha de entrada. La descripción del técnico no será obligatoria ya que todos los usuarios de la aplicación no son técnicos.

#### 4.2.5 Añadir imagen

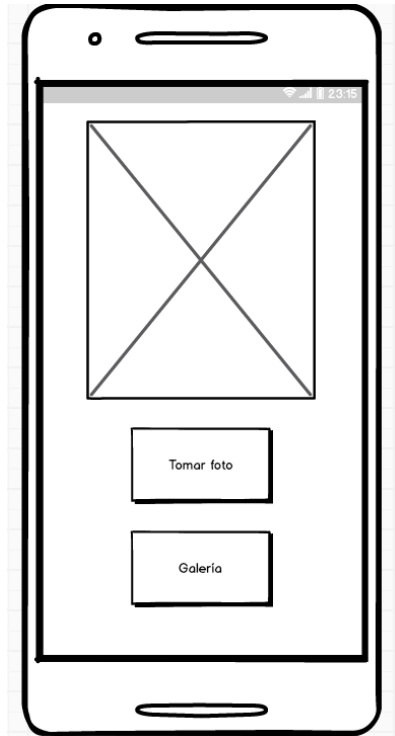


Ilustración 10. Prototipo añadir imagen

La pantalla de añadir imagen servirá para adjuntar una imagen a la ficha de nuestro artículo, desde ella podremos acceder a la cámara o a la galería para asignarle una imagen a nuestro artículo.

#### 4.2.6 Buscar artículo

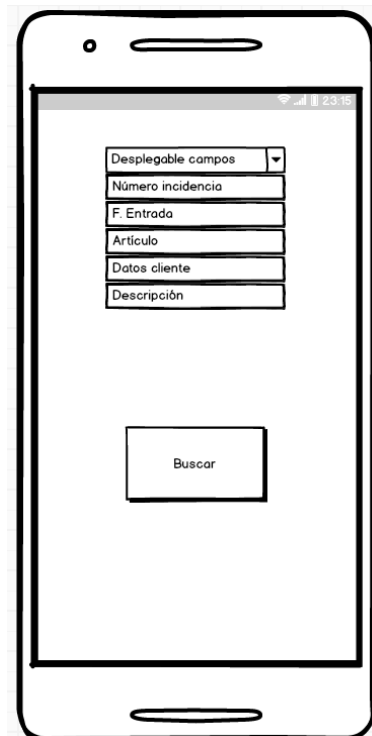


Ilustración 11. Prototipo buscar artículo

La pantalla de buscar artículo servirá para encontrar los artículos almacenados en nuestra base de datos. En ella elegiremos el tipo de atributo que queremos encontrar y teclearemos el campo a buscar para finalmente pulsar buscar y obtener los resultados disponibles con la opción deseada.

#### 4.2.7 Editar artículo encontrado

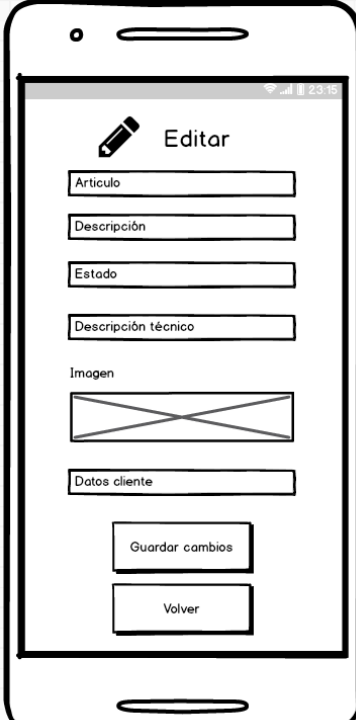


Ilustración 12. Prototipo de la pantalla de edición de un artículo en un dispositivo móvil. La interfaz muestra un título 'Editar' con un ícono de lápiz. Debajo se encuentran campos de texto para 'Artículo', 'Descripción', 'Estado', 'Descripción técnico' y 'Datos cliente'. Hay un campo 'Imagen' con un ícono de una imagen tachada. Al final de la pantalla hay dos botones: 'Guardar cambios' y 'Volver'.

Ilustración 12. Prototipo editar artículo

Desde esta pantalla podremos editar los artículos encontrados. Para ello pulsaremos encima de editar, lo que nos desbloqueará los atributos mostrados y nos permitirá modificarlos. Una vez finalizada la edición pulsaremos en guardar cambios para que estos sean permanentes y se almacenen en el sistema. Dispondremos también de un botón volver que utilizaremos en caso de necesitar acceder de nuevo a la pantalla buscar.

## 5-Evaluación

Una vez realizado diseño se procederá a realizar la evaluación que permitirá recibir las opiniones de los usuarios sobre la usabilidad de la aplicación.

Para ello, se realizarán los test de usuario que son un método de evaluación donde nos centraremos en el usuario final y en definir si se cumple con el objetivo o es necesario aplicar alguna mejora a nuestro diseño. A los usuarios se les proporcionará el prototipo horizontal de la aplicación y se les pedirá que realicen las tareas detectadas en el análisis de tareas:

1. Registro de usuario y login.
2. Alta de nuevo artículo.
3. Búsqueda de un artículo.
4. Modificación de los atributos de un artículo.
5. Cerrar sesión.

Una vez realizado el testeo con las tareas propuestas se les formularán una serie de preguntas para recoger todos los datos posibles sobre el uso y sus impresiones. Estas preguntas serán las siguientes:

1. ¿Pudo realizar las tareas de forma correcta? En caso negativo, indicar que actividad.
2. ¿Le resultó fácil realizar estas tareas?
3. ¿Le resultó intuitiva la navegación?
4. ¿Añadiría algún elemento a la navegación?
5. ¿Añadiría alguna función?
6. ¿La aplicación cumple sus expectativas?
7. ¿Mejoraría algún elemento visual de la aplicación? ¿Cuál?

## 6-Definición casos de uso

### 6.1-Diagrama UML

A continuación se presenta el diagrama de flujo de la aplicación:

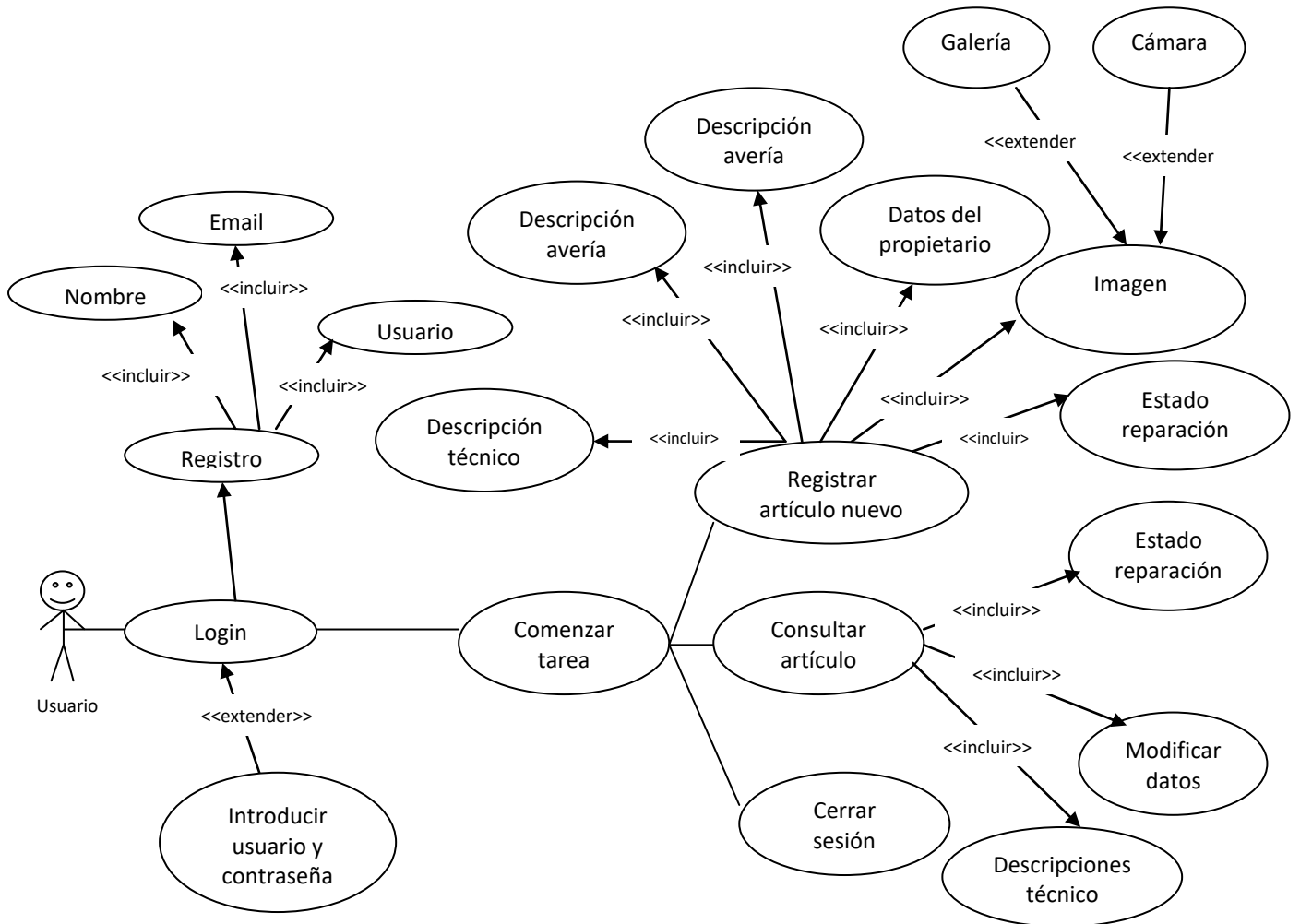


Ilustración 13. Diagrama casos de uso

## 6.2-Detalle casos de uso

Caso de uso 1	
Nombre	Registro usuario.
Actor	Técnico, recepcionista.
Precondiciones	Empresa tiene que estar dada de alta en Backendless y disponer de su base de datos.
Flujo	<ol style="list-style-type: none"> <li>1. Abrir aplicación</li> <li>2. En pantalla de inicio pulsar registro.</li> <li>3. Rellenar los datos de registro de usuario (Nombre, email, usuario y contraseña)</li> <li>4. Pulsar el botón confirmar.</li> </ol>
Postcondiciones	Se registra usuario en la aplicación.

Tabla 4. Caso de uso 1

Caso de uso 2	
Nombre	Entrada artículo nuevo.
Actor	Recepcionista, técnico.
Precondiciones	Estar registrado en el sistema.
Flujo	<ol style="list-style-type: none"> <li>1. Abrir aplicación</li> <li>2. En pantalla inicio introducir usuario y contraseña.</li> <li>3. En pantalla de opciones elegir nuevo artículo.</li> <li>4. Rellenar los datos del artículo (datos de cliente, fecha de entrada, descripción de la avería, imagen del artículo)</li> <li>5. Pulsar sobre el botón guardar.</li> </ol>
Postcondiciones	Hay un artículo nuevo en el sistema.

Tabla 5. Caso de uso 2

Caso de uso 3	
Nombre	Consulta estado artículo.
Actor	Recepcionista, técnico.
Precondiciones	Que el artículo este creado en el sistema.
Flujo	<ol style="list-style-type: none"> <li>1. Abrir aplicación</li> <li>2. En pantalla inicio introducir usuario y contraseña.</li> <li>3. En pantalla de opciones elegir buscar artículo.</li> <li>4. Rellenar el campo que se desea buscar y pulsar en buscar.</li> </ol>
Postcondiciones	Se muestra ficha del artículo deseado.

Tabla 6. Caso de uso 3



<b>Caso de uso 4</b>	
<b>Nombre</b>	<b>Modificar datos artículo.</b>
<b>Actor</b>	<b>Técnico.</b>
<b>Precondiciones</b>	<b>Que el artículo este creado en el sistema.</b>
<b>Flujo</b>	<ol style="list-style-type: none"> <li><b>1. Abrir aplicación</b></li> <li><b>2. En pantalla inicio introducir usuario y contraseña.</b></li> <li><b>3. En pantalla de opciones elegir buscar artículo.</b></li> <li><b>4. Rellenar el campo que se desea buscar y pulsar en buscar.</b></li> <li><b>5. Modificar los atributos.</b></li> <li><b>6. Pulsar botón guardar.</b></li> </ol>
<b>Postcondiciones</b>	<b>Los atributos de la pieza se modifican.</b>

Tabla 7. Caso de uso 4

<b>Caso de uso 5</b>	
<b>Nombre</b>	<b>Cerrar sesión.</b>
<b>Actor</b>	<b>Recepcionista, técnico.</b>
<b>Precondiciones</b>	<b>Acceder al sistema con usuario registrado.</b>
<b>Flujo</b>	<ol style="list-style-type: none"> <li><b>1. Abrir aplicación</b></li> <li><b>2. En pantalla inicio introducir usuario y contraseña.</b></li> <li><b>3. Realizar las tareas deseadas.</b></li> <li><b>3. Volver a pantalla de opciones y pulsar cerrar sesión.</b></li> </ol>
<b>Postcondiciones</b>	<b>Se requiere login para volver a usar la aplicación</b>

Tabla 8. Caso de uso 5

## 7-Diseño de la arquitectura

Para el diseño de la arquitectura se utilizará el patrón [modelo-vista-controlador \(MVC\)](#), que separa los datos y la lógica del negocio de la aplicación de la interfaz de usuario. Siguiendo este patrón los componentes se podrían definir de la siguiente forma:

-Modelo. Es la representación de la información con la que el sistema opera.

-Vista. Presenta el modelo en un formato adecuado para interactuar por tanto requiere de dicho modelo la información que debe representar como salida.

-Controlador. Responde a eventos e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información.

Que adaptándolo a nuestro diseño para Android, nos dará el siguiente resultado.

-Modelo. En modelo se usarán las clases que extienden de la API de Backendless. La API será la encargada de crear automáticamente las tablas necesarias en el servidor para la persistencia de los datos.

-Vista. En vista definiremos cada una de las pantallas de la aplicación.

-Controlador. Dentro del controlador se implementarán las clases java encargadas de controlar los eventos y acciones del usuario.

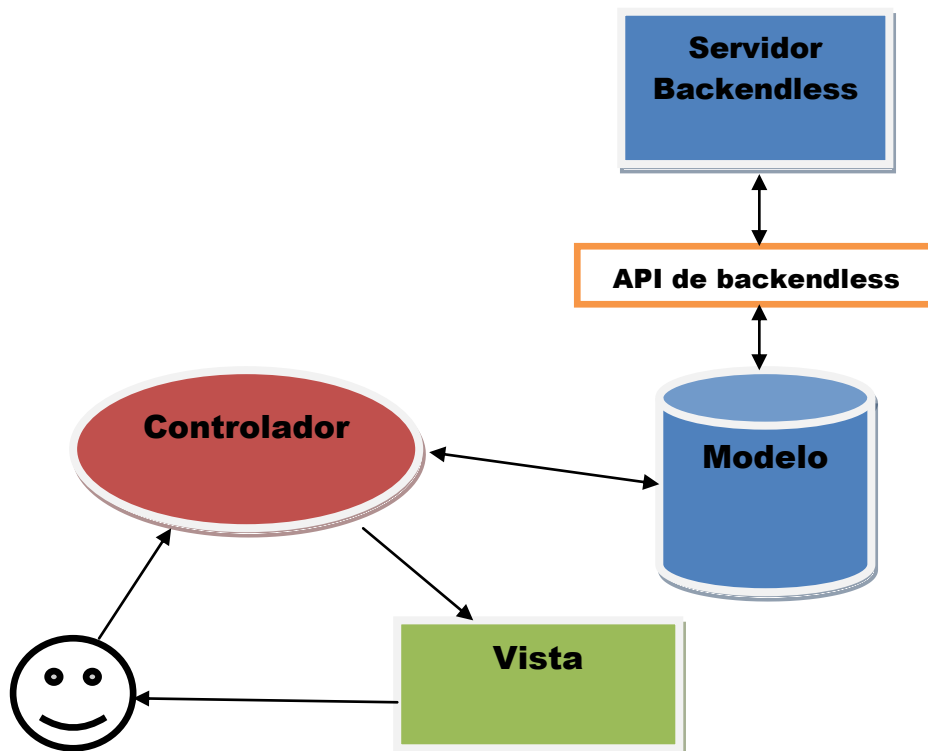


Ilustración 14. Diagrama explicativo del sistema

Definiremos también las entidades presentes en nuestro sistema, sus relaciones y atributos:

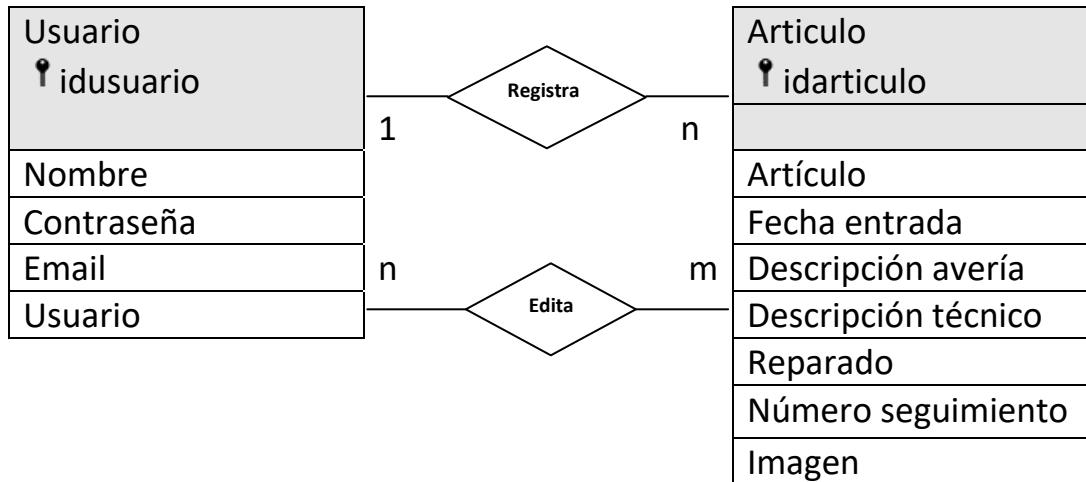


Ilustración 15. Entidades y atributos

Por último se mostrarán las clases que presentará nuestro sistema.

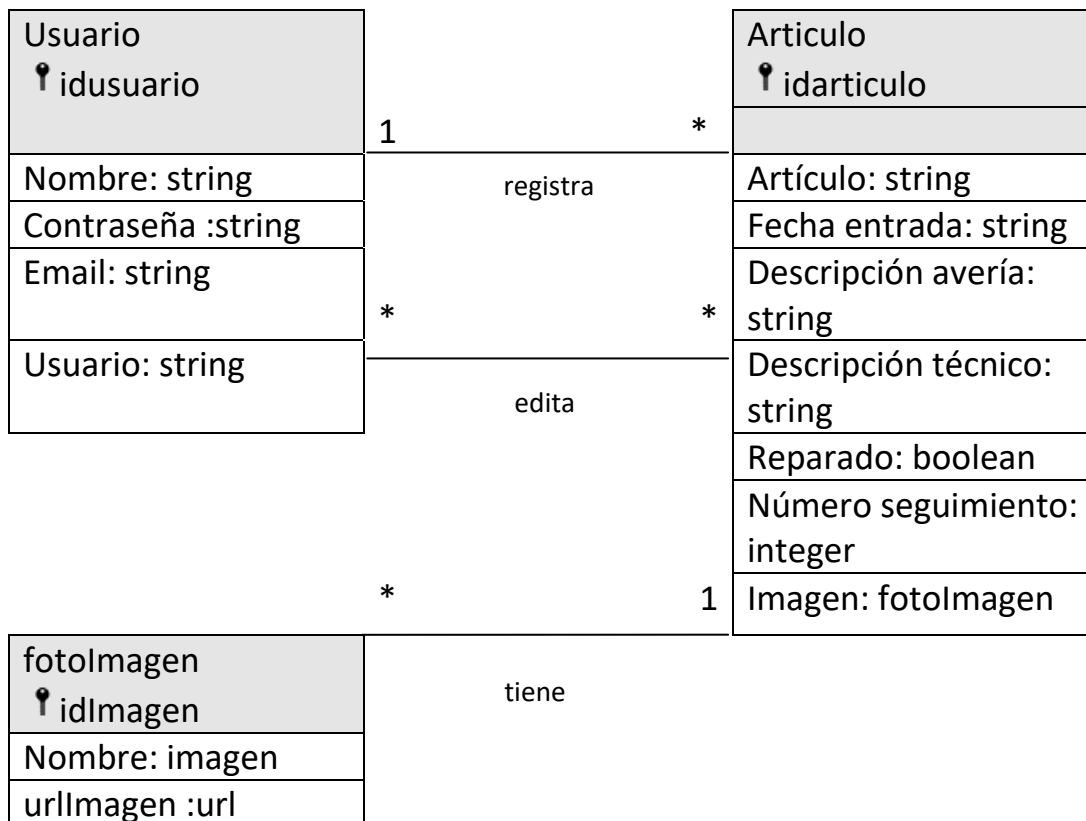


Ilustración 16. Diagrama UML entidades y clases

## 8-Implementación

La implementación de la aplicación se ha llevado a cabo desde el IDE AndroidStudio. En él se ha desarrollado el código fuente de las funcionalidades deseadas así como la integración de nuestra App con Backendless. Durante este proceso se han modificado algunas de las pantallas que se habían descrito en nuestro prototipo, buscando facilitar la navegación y el desarrollo. En capítulos posteriores se describirán estos cambios.

### 8.1-Servidor backend

Para el correcto funcionamiento de nuestra aplicación se utilizará como servidor backend el que nos proporciona la plataforma Backendless. El primer paso para empezar a trabajar será registrarnos en su página. Este usuario será el administrador de la base de datos de la aplicación. En este caso se ha utilizado el usuario “Raulrs1985@hotmail.com” con contraseña “proyecto”.

Para continuar bastará con seguir los pasos que se indican en la [guía rápida de inicio](#). Una vez realizados estos pasos ya tendremos la última versión de Backendless disponible en nuestro proyecto.

Dentro de la página de gestión de Backendless veremos que en el apartado “Data” de Backendless podremos ir consultando los artículos y los usuarios que se irán almacenando. También podremos consultar las imágenes guardadas desde el apartado “Files” y la carpeta “fotosSat”.

### 8.2-Estructura de la aplicación

A continuación, se mostrará la estructura de nuestra aplicación y se comentarán los archivos más importantes que la componen.

#### 8.2.1 Activities de carpeta com.edu.uoc.gestionsat

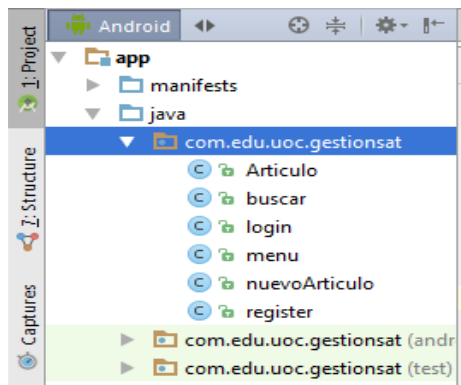


Ilustración 17. Estructura carpeta java

Estas son las *activities* que lo componen:

- Artículo. Aquí se define nuestra clase artículo y todos sus atributos.
- Buscar. En la *activity* buscar se realizará todo el proceso de búsqueda. En caso de ser una búsqueda exitosa también se podrá editar el artículo deseado.
- Login. La *activity* login es la encargada de gestionar la pantalla inicial de nuestra aplicación y es donde se realizará el proceso de login o en caso de querer registrarse se nos enviará a la *activity* register.
- Menu. En esta *activity* se gestionará el comportamiento del menú principal de la aplicación y se redireccionará a los diferentes apartados de la aplicación según el botón pulsado.
- NuevoArticulo. Dentro de esta *activity* se llevará a cabo el alta del artículo nuevo. Será necesario rellenar los campos que se solicitan e incluir una foto de este desde la cámara o desde la galería.
- Register. Desde esta *activity* se gestionará el alta del usuario en el servidor Backendless. Será necesario completar todos los campos solicitados para que se realice con éxito.

### 8.2.2 Carpeta res

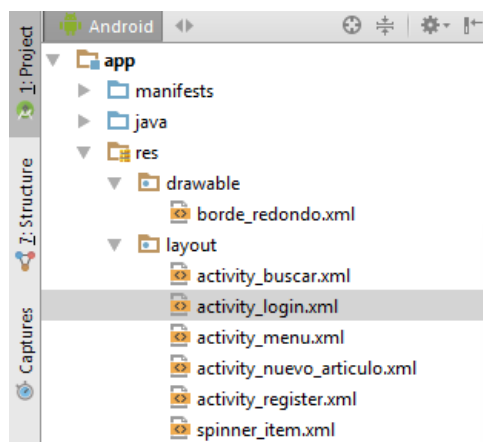


Ilustración 18. Estructura de res

En la imagen podemos observar la estructura de los archivos de *layouts* que se utilizarán en la aplicación. Cada *layout* hace referencia a la *activity* con su mismo nombre y se encargará de configurar el aspecto visual de nuestra aplicación mediante los ficheros “.xml”. También se pueden observar los ficheros “borde\_redondo”, que se ha utilizado para personalizar los botones de la aplicación y “spinner\_item” el cual se encargará de cambiar el tamaño y color de los elementos del *spinner* que se muestra en la pantalla de buscar. Desde aquí también podemos acceder a la carpeta “ic\_launcher” para modificar el icono de la aplicación. En este caso se ha mantenido el que genera por defecto AndroidStudio.

### 8.3-Modificaciones en el diseño de la aplicación

Como se ha comentado durante el desarrollo de la aplicación se han modificado algunas de las pantallas presentadas en el prototipo. A continuación, se mostrarán y se justificarán estos cambios.

#### 8.3.1-Resultado pantalla alta de nuevo artículo



Ilustración 19. Prototipo y resultado de pantalla artículo nuevo

En este caso, en el prototipo se pulsaba en imagen para desplegar un menú con las opciones de cámara y galería. Al realizar el diseño ha resultado más útil, visual y fácil de implementar colocar dos botones en la

propia pantalla para acceder directamente a la opción elegida. Además, utilizando este método eliminamos pantallas innecesarias. Para el estado en que se encuentra el artículo se ha optado por un switch que cambiará el valor de “¿Reparado?”. Finalmente, para que el diseño de la pantalla no se vea alterado y quede mejor organizada la información mostrada, se ha desactivado la rotación de pantalla.

### 8.3.2-Resultado pantalla búsqueda y edición de artículo

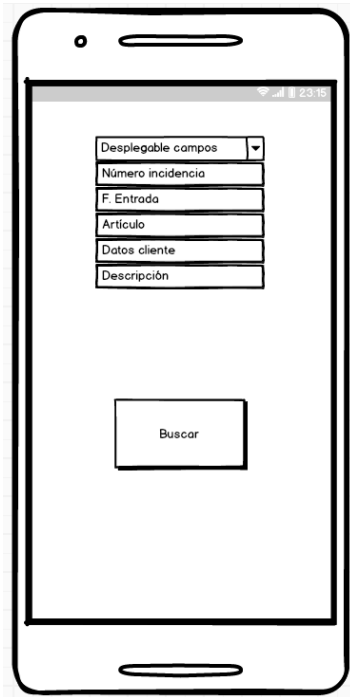


Ilustración 20. Prototipo de buscar

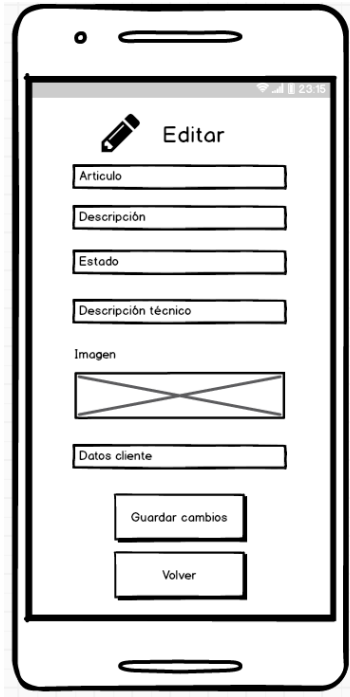


Ilustración 21. Prototipo de editar

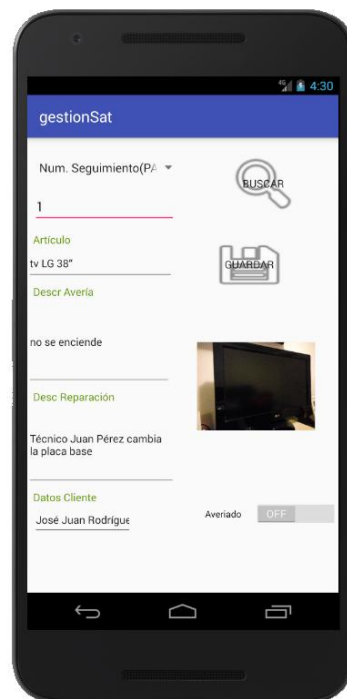
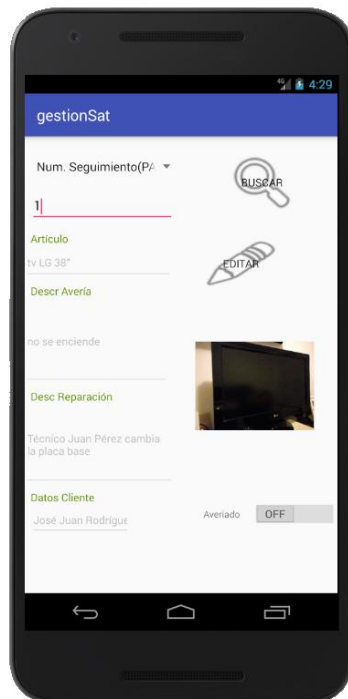


Ilustración 22. Resultado buscar y editar con un resultado encontrado

Como observamos en las imágenes al implementar la pantalla de búsqueda se ha optado por fusionar esta pantalla con la de edición mostrada en el prototipo. De esta forma se está aprovechando todo el espacio que ofrece la pantalla del terminal. En el caso anterior la pantalla de búsqueda quedaba demasiado vacía y se añadía otra pantalla (la de editar) que realmente no era necesaria. Otra de las ventajas presentes es que resulta más fácil y rápido de implementar ya que no se tienen que estar enviando datos entre *activities* y se maneja toda la información y los métodos desde la misma *activity*. También se ha modificado la pantalla para que en el caso de varias coincidencias con la búsqueda muestre una lista con scroll con todos los resultados. El resultado lo podemos ver en la siguiente imagen.

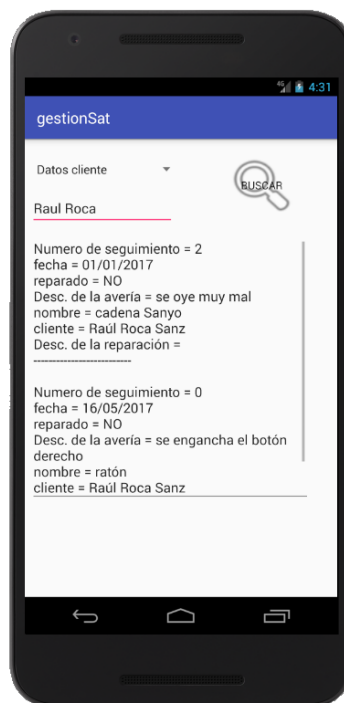


Ilustración 23. Resultado buscar con varios resultados

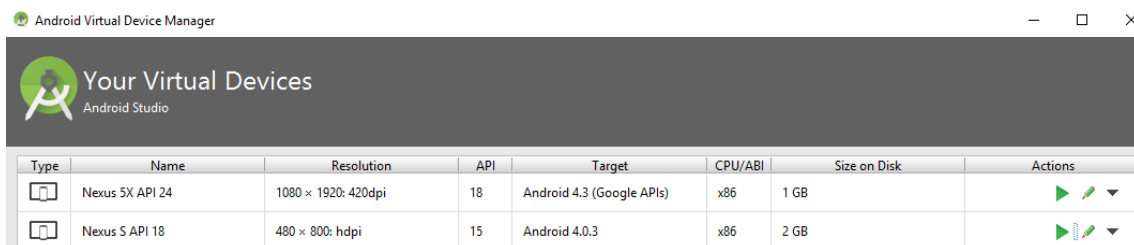
En este caso, no estará disponible la edición en esta lista. Solo se mostrará la opción de editar en el caso de que se encuentre un único artículo en la búsqueda como en el caso de la ilustración 22. En estas pantallas también se ha optado por desactivar la rotación de pantalla. De esta forma, garantizamos que la información se presentará siempre de forma vertical lo que mejorará la visualización.



## 9-Pruebas

### 9.1-Entorno de pruebas

Para realizar las pruebas será necesario configurar el entorno de pruebas. En este caso se ha optado tanto por un terminal físico como por el emulador que ofrece Android Studio 2.3.1, de esta forma podremos probar la aplicación en más dispositivos pudiendo encontrar más problemas o fallos de diseño. Nuestro terminal físico es un smartphone Xiaomi Redmi Note II, con Android 5.0.1, lo que equivale a la API 21. En el caso del emulador de Android se han aplicado diferentes configuraciones en el tamaño de pantalla y en la API sobre la que se ejecutará. En la siguiente figura se muestran las opciones utilizadas:









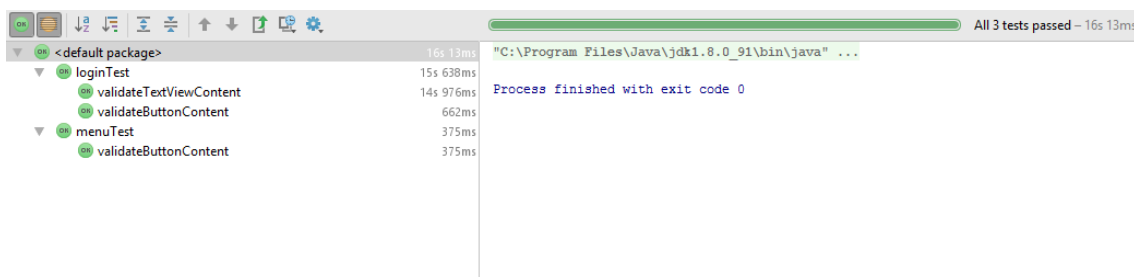
Type	Name	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Nexus 5X API 24	1080 x 1920: 420dpi	18	Android 4.3 (Google APIs)	x86	1 GB	 
	Nexus S API 18	480 x 800: hdpi	15	Android 4.0.3	x86	2 GB	 

Ilustración 24. Versiones emuladas

### 9.2-Pruebas realizadas

En este caso y al estar utilizando las librerías de Backendless para realizar tareas como el login, el almacenaje o la búsqueda no se ha considerado necesaria la realización de pruebas unitarias para estos métodos. Al haber sido ya testeados por la comunidad de usuarios de Backendless se ha supuesto que el funcionamiento no tiene fallos. Por otra parte, sí que se han realizado pruebas unitarias de los elementos presentes en las pantallas login y menú para verificar la existencia y el contenido de los “Textview” y “Button”. Estos son los resultados obtenidos:



Test Name	Duration
<default package>	16s 13ms
loginTest	15s 638ms
loginTest.validateTextViewContent	14s 976ms
loginTest.validateButtonContent	662ms
menuTest	375ms
menuTest.validateButtonContent	375ms

Ilustración 25. Pruebas unitarias

Además, también se han realizado las pruebas de integración de la aplicación para garantizar el funcionamiento adecuado según los casos de uso descritos. En la siguiente tabla se muestran los diferentes casos de prueba llevadas a cabo con el usuario “raulrs1985@gmail.com” y el password “4”.

Caso de prueba	Funcionalidad probada	Pasos seguidos
Registrar usuario	Registro de usuario	<ol style="list-style-type: none"> <li>1. En la pantalla de inicio pulsar botón “Registrar”.</li> <li>2. Rellenar campos solicitados y pulsar “Registrar”.</li> <li>3. Acceder a Backendless y verificar la creación del usuario.</li> </ol>
Login de usuario	Login de usuario	<ol style="list-style-type: none"> <li>1. En la pantalla de inicio teclear email y contraseña y pulsar “Login”.</li> <li>2. Verificar que se accede al menú de la aplicación.</li> </ol>
Dar de alta artículo nuevo.	Alta artículo nuevo	<ol style="list-style-type: none"> <li>1. En la pantalla de inicio teclear email y contraseña y pulsar “Login”.</li> <li>2. En la pantalla de menú pulsar el botón “Nuevo artículo”.</li> <li>3. Rellenar todos los campos solicitados.</li> <li>4. Pulsar botón “Tomar foto” o “Galería” para tomar una foto.</li> <li>5. Pulsar el botón “Guardar” para almacenar el artículo en Backendless.</li> <li>6. Acceder a base de datos de Backendless y verificar que el artículo se ha añadido correctamente.</li> </ol>
Buscar artículo por los diferentes campos disponibles.	Búsqueda artículo	<ol style="list-style-type: none"> <li>1. En la pantalla de inicio teclear email y contraseña y pulsar “Login”.</li> <li>2. En la pantalla de menú pulsar el botón “Buscar”.</li> <li>3. En el desplegable seleccionar los diferentes campos desplegables.</li> <li>4. Teclear el dato que se desea buscar.</li> <li>5. Pulsar botón “Buscar”.</li> </ol>
Editar artículo	Edición de artículos.	<ol style="list-style-type: none"> <li>1. En la pantalla de inicio teclear email y contraseña y pulsar “Login”.</li> <li>2. En la pantalla de menú pulsar el botón “Buscar”.</li> <li>3. En el desplegable seleccionar número de seguimiento del artículo.</li> <li>4. Teclear el número de seguimiento del artículo que se desea modificar.</li> <li>5. Pulsar el botón “Editar”-</li> <li>6. Modificar los atributos deseados.</li> <li>7. Pulsar botón “Guardar”.</li> </ol>

		8. Acceder base de datos de Backendless y verificar que se ha modificado correctamente.
Cerrar sesión	Cerrar sesión.	<ol style="list-style-type: none"> <li>1. En la pantalla de inicio teclear email y contraseña y pulsar "Login".</li> <li>2. En el menú pulsar el botón "Cerrar".</li> <li>3. Volver a acceder a la aplicación y verificar que no podemos acceder al menú sin identificarnos de nuevo.</li> </ol>

Tabla 9. Casos de prueba

Todas estas pruebas se han realizado con éxito en las diferentes configuraciones mostradas.

## 10-Conclusiones

Llegados a este apartado se recapitulará sobre la propuesta y el producto aportado. En este caso, se han llevado a cabo satisfactoriamente los objetivos propuestos en el planteamiento inicial del trabajo tal y como se aprecia en el resultado final. Recordemos que se modificó ligeramente el planteamiento de alguna de las pantallas de la aplicación. Esto se hizo para no complicar excesivamente el diseño y la implementación. Gracias a ello, se consiguió realizar el diseño sin perder demasiado tiempo en los problemas encontrados.

Por otro lado, sobre el tema de la planificación se puede afirmar que esta se ha ajustado de forma correcta a la realidad. En un principio se pensó en ajustar un poco más la dedicación horaria pero con la opción elegida se ha garantizado el correcto desarrollo de todas las tareas propuestas dentro del marco de tiempo establecido.

Otro de los puntos que me gustaría destacar es la posibilidad de mejoras que ofrece la aplicación. A continuación, se mencionan las posibles mejoras:

- Añadir la opción de capturar video en los artículos. Esto le daría un aporte multimedia para identificar mejor el artículo.
- Añadir la opción de guardar la dirección del cliente y mostrarla en un mapa para facilitar las entregas de artículos en el domicilio de los clientes.
- Mejorar el buscador de artículos para que en el caso de encontrar varios artículos se pueda acceder directamente a ellos desde la lista mostrada.

Finalmente, destacar que a título personal el desarrollo completo de una aplicación móvil funcional es una actividad bastante gratificante. Aunque el camino suele estar lleno de obstáculos y frustraciones sobre todo cuando algo no funciona como se desearía. Aun así, está ha sido una oportunidad para aprovechar, profundizar y poner en práctica los conceptos estudiados en el máster de desarrollo de aplicaciones para móviles.

## 11-Bibliografía

1-Tienda de aplicaciones online de IOS. APP Store.

<https://itunes.apple.com/es/genre/ios/id36?mt=8>

2-Aplicación SimplyGest SAT Móvil.

<https://itunes.apple.com/es/app/simplygest-sat-movil/id1050632473?mt=8>

3-Tienda de aplicaciones de Android. PlayStore.

<https://play.google.com/store?hl=es>

4-IDE AndroidStudio.

<https://developer.android.com/studio/index.html?hl=es-419>

5-Documentación Backendless.

[https://backendless.com/documentation/users/android/users\\_overview.htm](https://backendless.com/documentation/users/android/users_overview.htm)

6-Guia de inicio para el uso de Backendless en Android.

<https://backendless.com/mobile-developers/quick-start-guide-for-android/>

7-Documentación Android.

<https://developer.android.com/guide/index.html>

8-Cantidad relativa de dispositivos que usan una plataforma determinada de Android.

<https://developer.android.com/about/dashboards/index.html>

9-Definición patrón MVC.

<https://es.wikipedia.org/wiki/Modelo%2%80%93vista%2%80%93controlador>

10-Documentación UOC.

- Módulo 1. Introducción al trabajo final
- Módulo 2. El trabajo final como proyecto
- Módulo 3. La gestión del proyecto a lo largo del trabajo final

## 12-Anexos

### 12.1-Instalación

Para instalar nuestra aplicación se proporcionará el archivo con nombre “gestionSat.apk”. Al no estar disponible desde PlayStore nuestro smartphone bloqueará su instalación. Esto lo solucionaremos desde “ajustes->privacidad->orígenes desconocidos”. Al activar está casilla ya podremos instalar la aplicación en nuestro dispositivo. Es muy importante recordar que como mínimo será necesario que nuestro dispositivo disponga de una versión superior a la API 14 de Android para que la aplicación funcione correctamente.

Si deseamos administrar nuestra base de datos accederemos directamente a la página de Backendless. <https://develop.backendless.com/login> donde teclearemos el usuario y contraseña con el que nos hemos registrado. Para este proyecto se ha utilizado el email “raulrs1985@gmail.com” y el password “proyecto”

### 12.2-Manual de uso

#### 12.2.1-Login

Esta será la pantalla de bienvenida e inicio. Si el usuario ya está registrado solo tendrá que acceder con su correspondiente email y password. Si no lo está puede registrarse con el botón “REGISTRAR”.



Ilustración 26.Pantalla inicio

### 12.2.2-Registro

En caso de no haber usado nunca la aplicación, el usuario tendrá que realizar su registro desde esta pantalla. Bastará con rellenar los campos que se solicitan y pulsar en “Registrar”.

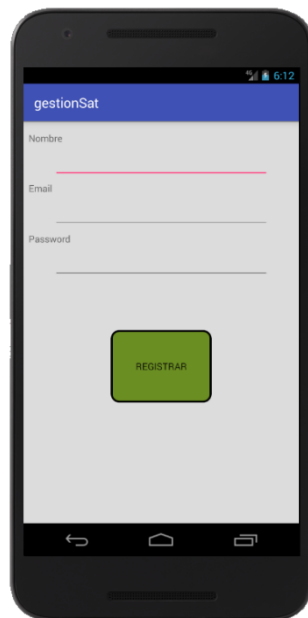


Ilustración 27. Pantalla registro

### 12.2.3-Menú

Desde esta pantalla el usuario puede decidir las acciones que desea realizar. Crear un nuevo artículo, buscarlo o cerrar la sesión.



Ilustración 28. Pantalla menú

### 12.2.4-Nuevo artículo

Para crear un nuevo artículo será necesario rellenar todos los campos que se solicitan en la pantalla a excepción de la descripción del técnico que es opcional. También será necesario añadir una imagen del artículo. Para ello podemos pulsar en “Tomar foto” si queremos hacer uso de la cámara o en “Galería” si ya disponemos de la imagen en la memoria del dispositivo. Para finalizar pulsaremos “Guardar”. Se nos pedirá la confirmación de la acción y se nos indicará cual es el número de seguimiento de ese artículo. Un mensaje nos informará si la acción se ha llevado a cabo con éxito.



Ilustración 29. Pantalla artículo nuevo

### 12.2.5-Buscar artículo

Desde esta pantalla podremos realizar la búsqueda de artículos. El menú desplegable nos dejará elegir el tipo de búsqueda que deseamos realizar. Solo tenemos que seleccionar el que más nos convenga.



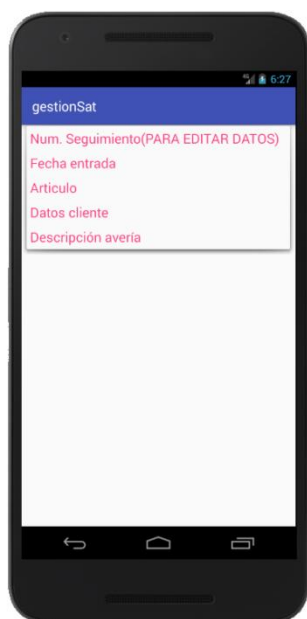


Ilustración 30. Pantalla buscar

En el caso de encontrar el artículo aparecerá el botón “editar” que se encargará de activar esa función si se pulsa.

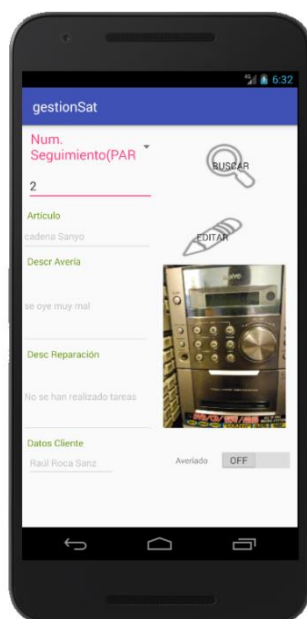


Ilustración 31. Pantalla artículo encontrado

En el caso de encontrar varias coincidencias con el campo buscado se mostrará una lista con todos los artículos que contienen ese campo.



Ilustración 32. Pantalla para varios artículos encontrados.

### 12.2.6-Editar artículo

En el caso de querer editar los campos de algún artículo pulsaremos el botón editar. Esto nos desbloqueará todos los atributos a excepción de la imagen que no es editable. Una vez realizadas las modificaciones se pueden guardar pulsando el botón “Guardar”. Antes de llevar a cabo la acción se nos pedirá confirmación. Finalmente, aparecerá un mensaje indicando si se ha llevado a cabo la acción de forma correcta.

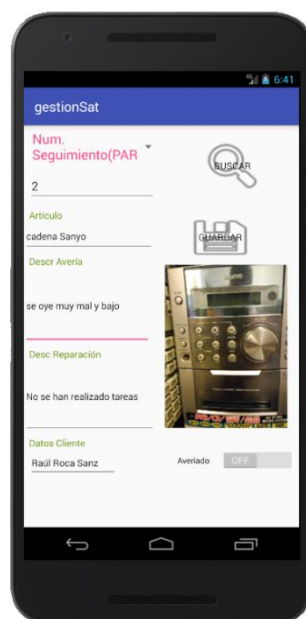


Ilustración 33. Pantalla editar artículo

### 12.2.7-Cerrar sesión

Finalmente si deseamos cerrar la sesión pulsaremos el botón del menú de "CERRAR". Un mensaje nos indicará si la acción se ha realizado correctamente.



Ilustración 34. Cerrar sesión