

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN  
TECNOLOGÍAS DE LA INFORMACIÓN



"DESARROLLO DE APLICACIÓN WEB  
PARA GESTIÓN DE AVERÍAS"

TRABAJO FIN DE GRADO

Junio -2021

AUTOR: Carmelo Peñalver Sánchez

DIRECTOR: Eloy Alarcón Ruiz

## **AGRADECIMIENTOS**

Agradezco a mis padres y mis amigos por todo el apoyo y la dedicación que han puesto en mi durante los 5 años de carrera.

También le doy mi agradecimiento a todos los profesores que me han impartido clase a lo largo de mi carrera universitaria, gracias a ellos estoy donde estoy ahora.



# Índice de contenido

1. INTRODUCCIÓN.....	7
1.1. Motivación del desarrollo .....	7
1.2. Objetivos del trabajo.....	8
1.2.1. Objetivos principales.....	8
1.2.2. Objetivos personales .....	8
1.3. Límites del proyecto.....	9
2. ESTADO DEL ARTE .....	9
2.1. Situación Inicial.....	9
2.2. Herramientas actuales .....	10
2.2.1. HOMEPPY: Presupuesto gratis, reformas y reparaciones .....	10
2.2.2. Wolly.....	18
2.3. Propuestas de mejora.....	21
3. HIPÓTESIS DE TRABAJO.....	22
3.1. Tecnologías y hardware utilizado .....	22
3.1.1. Software, lenguajes de programación .....	22
3.1.2. Hardware .....	30
4. DESARROLLO.....	31
4.1. Desarrollo de la herramienta. Requerimientos .....	31
4.1.1. Planteamiento del problema.....	31
4.1.2. Esquema organizativo.....	31
4.1.3. Análisis de requerimientos .....	35
4.2. Desarrollo de la herramienta. Fase de diseño.....	38
4.2.1. Ciclo de ingeniería del software .....	38
4.2.1.1. Descripción de actores. ....	38
4.2.1.2. Subsistemas y diagramas de casos de uso .....	39
4.2.1.3. Otros diagramas .....	56
4.3. Desarrollo de la herramienta. Implementación.....	59
4.3.1. Descripción de la base de datos.....	59
4.3.2. Implementación .....	61
5. CONCLUSIONES Y PROPUESTAS.....	63
5.1. Conclusiones sobre el proyecto .....	63
5.2. Propuestas de futuro .....	63
6. BIBLIOGRAFÍA.....	64

7. ANEXO .....	65
7.1. Punto de vista del usuario “cliente”.....	72
7.2. Punto de vista del usuario “profesional”.....	86

## Índice de ilustraciones

Ilustración 1. Página principal de HOMEPPY.....	11
Ilustración 2. Ejemplo de la lista de intervenciones .....	12
Ilustración 3. Pantalla del configurador para el servicio de manitas. ....	13
Ilustración 4. Lista de productos de aires acondicionados de HOMEPPY .....	13
Ilustración 5. Pantalla principal de la aplicación móvil.....	14
Ilustración 6. Página del Tawk To al presionar el botón de chat.....	15
Ilustración 7. Pantalla de subareas del área principal 'Electricidad' .....	16
Ilustración 8. Lista de servicios disponibles del área 'Electricidad' .....	17
Ilustración 9. Listado de servicios proporcionados por Wolly.....	18
Ilustración 10. Listado con servicios más específicos relacionados con Montaje e instalación. ....	19
Ilustración 11. Formulario para detallar más el servicio del cliente. ....	19
Ilustración 12. Formulario para añadir la ubicación a la petición. ....	20
Ilustración 13. Formulario para seleccionar el tipo de urgencia del servicio. ....	20
Ilustración 14. Logo de React Native.....	22
Ilustración 15. Logo de NodeJS. ....	23
Ilustración 16. Logo de npm.....	24
Ilustración 17. Logo de Expo.....	25
Ilustración 18. Logo de Firebase.....	26
Ilustración 19. Consulta con una cadena de filtros. ....	27
Ilustración 20. Componente React Native Gifted Chat. ....	28
Ilustración 21. Componente React Native Rating .....	29
Ilustración 22. Diagrama del proceso de prototipado efectivo.....	31
Ilustración 23. Primeros diseños de la aplicación.....	33
Ilustración 24. Diseño mencionando el elemento onboarding de la aplicación.....	33
Ilustración 25. Diseño de la homepage para el cliente y el profesional, diseño del perfil y mención al elemento drawer.....	34
Ilustración 26. Diseño de la parte inferior del perfil del cliente y del profesional.....	34
Ilustración 27. Subsistemas de AveriasApp.....	39

<b>Ilustración 28. Casos de uso del usuario no registrado en el subsistema de gestión de usuarios. ....</b>	<b>39</b>
<b>Ilustración 29. Casos de uso del usuario en el subsistema de gestión de usuarios.....</b>	<b>40</b>
<b>Ilustración 30. Casos de uso del administrador en el subsistema de gestión de usuarios. ..</b>	<b>40</b>
<b>Ilustración 31. Casos de uso del usuario cliente en el subsistema de gestión de tratos. ....</b>	<b>45</b>
<b>Ilustración 32. Casos de uso del administrador en el subsistema de gestión de tratos. ....</b>	<b>46</b>
<b>Ilustración 33. Casos de uso del usuario cliente en el subsistema de gestión de averías. ....</b>	<b>53</b>
<b>Ilustración 34. Casos de uso del usuario profesional en el subsistema de gestión de averías. ....</b>	<b>53</b>
<b>Ilustración 35. Diagrama de actividad relacionado con iniciar sesión. ....</b>	<b>56</b>
<b>Ilustración 36. Diagrama de actividad relacionado con la búsqueda de profesionales.....</b>	<b>57</b>
<b>Ilustración 37. Diagrama de clases relacionado con los tratos. ....</b>	<b>58</b>
<b>Ilustración 38. Diagrama de distribución/despliegue.....</b>	<b>58</b>
<b>Ilustración 39. Diagrama de carril.....</b>	<b>59</b>
<b>Ilustración 40. Diagrama Gantt. ....</b>	<b>59</b>
<b>Ilustración 41. Primera página del elemento Onboarding. ....</b>	<b>65</b>
<b>Ilustración 42. Segunda página del elemento Onboarding.....</b>	<b>66</b>
<b>Ilustración 43. Última página del elemento Onboarding.....</b>	<b>67</b>
<b>Ilustración 44. Formulario de inicio de sesión. ....</b>	<b>68</b>
<b>Ilustración 45. Formulario de registro del usuario. ....</b>	<b>69</b>
<b>Ilustración 46. Formulario de registro del usuario. ....</b>	<b>70</b>
<b>Ilustración 47. Menú de categorías. ....</b>	<b>71</b>
<b>Ilustración 48. Opciones disponibles del menú desplegable. ....</b>	<b>72</b>
<b>Ilustración 49. Pantalla principal de la aplicación. ....</b>	<b>73</b>
<b>Ilustración 50. Pantalla principal de la sección Perfil.....</b>	<b>74</b>
<b>Ilustración 51. Datos personales del usuario.....</b>	<b>75</b>
<b>Ilustración 52. Formulario para editar los datos personales.....</b>	<b>76</b>
<b>Ilustración 53. Listado de profesionales en el ranking.....</b>	<b>77</b>
<b>Ilustración 54. Listado de categorías. ....</b>	<b>78</b>
<b>Ilustración 55. Listado de recomendaciones de profesionales relacionados con Mobiliario. ....</b>	<b>79</b>
<b>Ilustración 56. Perfil del profesional Carmelico Profesional.....</b>	<b>80</b>
<b>Ilustración 57. Chat entre el usuario cliente y el usuario profesional. ....</b>	<b>81</b>
<b>Ilustración 58. Advertencia de la creación del trato entre cliente y profesional. ....</b>	<b>82</b>
<b>Ilustración 59. Advertencia para editar el título de la avería.....</b>	<b>83</b>
<b>Ilustración 60. Formulario para cambiarle el título a la avería.....</b>	<b>84</b>
<b>Ilustración 61. Cambios registrados tras haber registrado el nuevo título.....</b>	<b>84</b>

<b>Ilustración 62. Advertencia sobre notificar la finalización de la avería. ....</b>	<b>85</b>
<b>Ilustración 63. Formulario para valorar al profesional encargado de la avería. ....</b>	<b>85</b>
<b>Ilustración 64. Pantalla principal del usuario “profesional”. ....</b>	<b>86</b>
<b>Ilustración 65. Perfil del profesional con el historial de averías completadas. ....</b>	<b>87</b>
<b>Ilustración 66. Perfil con el listado de valoraciones dadas por los clientes. ....</b>	<b>88</b>
<b>Ilustración 67. Información sobre una de las insignias del profesional.....</b>	<b>89</b>
<b>Ilustración 68. Listado de salas de chat del profesional con los clientes. ....</b>	<b>90</b>

## Índice de tablas

<b>Tabla 1. Worksheet 4.1. ....</b>	<b>37</b>
<b>Tabla 2. Actor Usuario no registrado.....</b>	<b>38</b>
<b>Tabla 3. Tabla Cliente (Rol de usuario) ....</b>	<b>38</b>
<b>Tabla 4. Tabla Profesional (Rol de usuario) ....</b>	<b>38</b>
<b>Tabla 5. Tabla Administrador .....</b>	<b>38</b>
<b>Tabla 6. Caso de uso 1.1.....</b>	<b>41</b>
<b>Tabla 7. Caso de uso 1.2.....</b>	<b>42</b>
<b>Tabla 8. Caso de uso 1.3.....</b>	<b>42</b>
<b>Tabla 9. Caso de uso 1.4.....</b>	<b>43</b>
<b>Tabla 10. Caso de uso 1.5.....</b>	<b>43</b>
<b>Tabla 11. Caso de uso 1.6.....</b>	<b>44</b>
<b>Tabla 12. Caso de uso 1.7.....</b>	<b>44</b>
<b>Tabla 13. Caso de uso 1.8.....</b>	<b>45</b>
<b>Tabla 14. Caso de uso 2.1.....</b>	<b>46</b>
<b>Tabla 15. Caso de uso 2.2.....</b>	<b>47</b>
<b>Tabla 16. Caso de uso 2.3.....</b>	<b>48</b>
<b>Tabla 17. Caso de uso 2.4.....</b>	<b>49</b>
<b>Tabla 18. Caso de uso 2.5.....</b>	<b>50</b>
<b>Tabla 19. Caso de uso 2.6.....</b>	<b>51</b>
<b>Tabla 20. Caso de uso 2.7.....</b>	<b>51</b>
<b>Tabla 21. Caso de uso 2.8.....</b>	<b>52</b>
<b>Tabla 22. Caso de uso 2.9.....</b>	<b>52</b>
<b>Tabla 23. Caso de uso 3.1.....</b>	<b>54</b>
<b>Tabla 24. Caso de uso 3.2.....</b>	<b>54</b>
<b>Tabla 25. Caso de uso 3.3.....</b>	<b>55</b>
<b>Tabla 26. Caso de uso 3.4.....</b>	<b>56</b>

## **1. INTRODUCCIÓN**

El tema que se va a abordar es la utilidad e importancia de las aplicaciones en la actualidad. En los tiempos que corren, la mayoría de la población ha evolucionado sus maneras de vivir: pedir comida, hacer la compra, tener citas... todo lo mencionado a través de aplicaciones móviles.

Esta aplicación está diseñada para ser usada por todo tipo de persona que requiera de los servicios de otra persona, en este caso un profesional, para que le resuelva todo tipo de problemas.

La aplicación se basa principalmente en la gestión de averías, donde un usuario puede navegar a través de ella, encontrar profesionales relacionados con una categoría en especial, comunicarse con dicho profesional a través del sistema de chat de la aplicación, cerrar tratos y valorar al profesional con el que ha tratado la avería.

### **1.1. Motivación del desarrollo**

La principal motivación por el desarrollo del trabajo de fin de grado es continuar fomentando la gran utilidad de las aplicaciones móviles, esta vez, con un gestor de averías.

Con mi aplicación, se podrá conseguir agilizar los típicos procedimientos que se tendrían que realizar en caso que un usuario corriente decida arreglar su avería por otros medios, ya sea de manera telefónica con su tienda de confianza o mediante alguna persona cercana a él.

Otro de los motivos del desarrollo de la aplicación es por el simple hecho de no estar sujeto a los horarios de alguna empresa dedicada a fontanería o electricidad, que el usuario pueda usar la aplicación sin tener ninguna aplicación.

## **1.2. Objetivos del trabajo**

### **1.2.1. Objetivos principales**

El principal objetivo de este trabajo de fin de grado es conseguir una aplicación que sea completamente útil y funcional, que sea intuitiva para un usuario con un rango de edad amplio.

Otro objetivo del trabajo es fomentar la comunicación y el uso de las aplicaciones móviles hacia las personas mayores ya que dicho rango de población tiende a ser reacia a usar aplicaciones móviles o páginas web para algo tan clásico como llamar a un fontanero a arreglar una tubería. Con mi aplicación aprenderán a comprender que hay que perderle el miedo a las tecnologías y ver que lo más importante que aportan es utilidad e incrementar la calidad de vida de las personas.

### **1.2.2. Objetivos personales**

Los objetivos personales que se han conseguido con este trabajo son: Aprender un framework desde 0, siendo este React Native, que está en auge últimamente, aprender a manejar Firebase, que es una plataforma para desarrollo de aplicaciones web y móvil perteneciente a Google, relacionando con Firebase, también es importante mencionar Cloud Firestore, la base de datos de tipo NoSQL que se ha utilizado para alojar todo tipo de datos(usuarios, mensajes de chat, categorías...etc.) que se encuentran en la app.

Otro objetivo personal digno de mencionar es el poder desarrollar la aplicación de manera híbrida una aplicación, en la universidad nos han enseñado a desarrollar aplicaciones para web o para móvil (Android) pero no se ha investigado de manera exhaustiva el desarrollo de aplicaciones híbridas, por eso quería tomar el reto de poder crear desde 0 una aplicación que funcione tanto en web como en cualquier dispositivo móvil.

Pienso que esta aplicación se puede seguir desarrollando en un futuro ya que cumple con funciones esenciales en el día a día de un numero incontable de personas, pienso que es una aplicación que puede ayudar a bastante gente que necesite arreglar algo de manera inmediata o que tenga alguna avería peculiar y encuentre en ella a un profesional capaz de poder arreglarlo y crear un trato justo tanto para el usuario como para el profesional.

### **1.3. Límites del proyecto.**

Con este proyecto no se pretende construir una aplicación que vaya a ir destinada al mundo real, por lo que en este proyecto no habrá elementos como el cifrado de datos del usuario como, por ejemplo, la contraseña. Como este proyecto va a ir destinado a un ámbito académico como es el trabajo de fin de grado, consideré el obviar dichos elementos.

Otra cosa a tener en cuenta es el tema de la monetización de la aplicación. En este caso no va a haber ningún tipo de monetización alguna, ya sea de manera: Unirse como socio premium para obtener más beneficios que otros usuarios sin pagar o tener que pagar desde el principio para usar la aplicación. Como he mencionado antes, este proyecto está ceñido a un ámbito académico por lo que no hay cabida a elementos así.

## **2. ESTADO DEL ARTE**

### **2.1. Situación Inicial**

Actualmente la gestión de averías se ha llevado siempre de manera telefónica, hablando con alguna tienda en particular o con algún experto que conozca el usuario. Existen algunas aplicaciones móviles sobre gestión de averías, pero dichas aplicaciones están sujetas a una empresa en particular, ¿Qué conlleva eso? Que el presupuesto puede que esté fijado por lo que no hay margen de negociación entre usuario y profesional, tener que seguir el horario que imponga la empresa a los profesionales por lo que puede que no se resuelva una urgencia el mismo día de la creación del trato.

El mundo de las averías necesita rapidez, más comunicación digital, más inmersión de profesionales que pueden que estén en paro y quieran darse a conocer mediante las redes.

Como se ha comentado con anterioridad, ya existen aplicaciones referentes a la gestión de averías que cumplen una serie de funciones por lo que este trabajo está destinado a mejorar algunos aspectos de dichas aplicaciones e incluso implementar nuevas ideas que pueden llegar a ser útiles en un futuro.

## **2.2. Herramientas actuales**

A continuación, se mostrará una lista de descripciones de las herramientas actuales que tratan sobre el mismo tema que he decidido desarrollar.

### **2.2.1. HOMEPPY: Presupuesto gratis, reformas y reparaciones**

HOMEPPY es una aplicación web en la que su principal función es el mantenimiento del hogar a través del arreglo de averías de mayor importancia como fontanería, electricidad o temas de cerrajería.

Por lo que se ve en su página web, se puede comprobar que sus precios son cerrados, es decir, no hay cabida a negociación. Esto es comprensible ya que los profesionales que van a arreglar o instalar algún tipo de elemento son trabajadores de la empresa de HOMEPPY.

La principal diferencia entre HOMEPPY y mi aplicación es que HOMEPPY se basa principalmente en instalaciones de todo tipo mientras que dejan el apartado de averías para 4 categorías (mencionadas anteriormente), comparándolo con mi aplicación, esta se basa principalmente en la gestión de averías de un número extenso de categorías importantes donde un usuario normal podría tener una avería o incidencia.

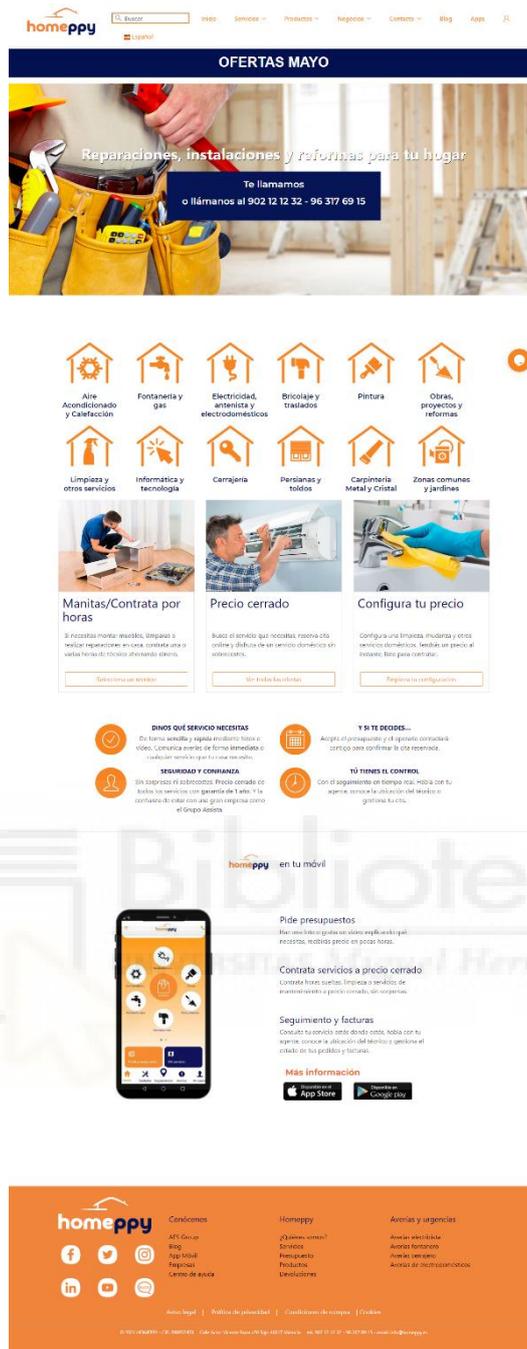


Ilustración 1. Página principal de HOMEPPY

Esto es lo que se encuentra un usuario al entrar en la página web de HOMEPPY, como se puede comprobar por la imagen, HOMEPPY muestra algunas áreas donde realizan instalaciones y algunas reparaciones.

También se puede observar los tres puntos que destacan tras la lista de áreas:

- Manitas/Contrata por horas: Dan una lista de intervenciones de técnicos especializados en algunas áreas con un precio totalmente cerrado, sin cabida de negociación. Otra cosa a tener en cuenta es que solo incluye 1 hora de mano de obra, dando a suponer que cobrarán más si están más con el trabajo a realizar.

### Contrata un técnico por horas



**Intervención de un fontanero**

Incluye desplazamiento y una hora de mano de obra de un técnico fontanero en tu domicilio o negocio.

**57,23 €**  
IVA incluido



**Intervención técnico calefacción**

Incluye desplazamiento y una hora de mano de obra de un técnico de calefacción en tu domicilio o negocio.

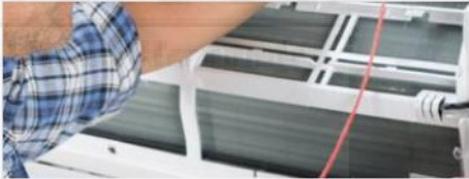
**63,89 €**  
IVA incluido



**Intervención técnico de gas**

Incluye desplazamiento y una hora de mano de obra de un técnico de gas en tu domicilio o negocio.

**63,89 €**  
IVA incluido



**Intervención técnico aire acondicionado**

Presupuesto intervención técnico aire acondicionado. Contrata un técnico para pequeños ajustes, revisiones o asesoramiento. Técnicos

**63,89 €**  
IVA incluido

*Ilustración 2. Ejemplo de la lista de intervenciones*

- Precio cerrado: HOMEPPY da una lista de todas las áreas que cubren además de todo lo que realizan en dichas áreas, dando por cada trabajo un precio cerrado.
- Configura tu precio: El usuario elige una configuración del trabajo a pedir y HOMEPPY da un precio aproximado según el trabajo pedido y las horas de trabajo. Se puede observar que HOMEPPY deja claro varias cosas: Hay precio mínimo para el pedido, el servicio es únicamente ofrecido a capitales de provincia

o lugares con una población superior a 100.000 habitantes, cobrando un extra si el domicilio del usuario no se encuentra en la lista que ofrece.

Configurador para:  
Servicio de manitas

### Elementos decorativos

<p><b>Colgar cuadros:</b> <span style="float: right;">2</span></p> <p><small>Incluye colgar cuadros con clavo, tornillo, alcañata o enganche tradicional. No está incluido colgar ...</small></p>	<p><b>Colgar barra de cortina:</b> <span style="float: right;">0</span></p> <p><small>Incluye instalar una barra de cortina de hasta 150 x 210 cm de material no pesado como acero, antracita, bronce ...</small></p>
<p><b>Fijar estor:</b> <span style="float: right;">0</span></p> <p><small>Incluye la fijación de un estor de hasta 80 x 150cm. Se deberán proporcionar los accesorios de ...</small></p>	<p><b>Fijar veneciana:</b> <span style="float: right;">0</span></p> <p><small>Incluye instalar una cortina veneciana de hasta 150 x 250 cm, con tirador de nylon. Se deberán aportar los ...</small></p>
<p><b>Fijar estantería a la pared:</b> <span style="float: right;">0</span></p> <p><small>Incluye fijar una estantería de hasta 154x90x35 cm. Se deberán aportar las guías, bisagras y bases regulables ...</small></p>	<p><b>Fijar espejo:</b> <span style="float: right;">0</span></p> <p><small>Incluye fijar un espejo decorativo de hasta 160x80 cm. Se deberá aportar los accesorios necesarios para una ...</small></p>
<p><b>Fijar tendedero:</b> <span style="float: right;">0</span></p> <p><small>Incluye fijar un tendedero de acordeón en pared con medidas no superiores a los 4 metros. Los ...</small></p>	<p><b>Colocar vinilo pequeño:</b> <span style="float: right;">0</span></p> <p><small>Incluye la colocación de un revestimiento vinílico de pared autoadhesivo con unas medidas no ...</small></p>
<p><b>Colocar vinilo mediano:</b> <span style="float: right;">0</span></p> <p><small>Incluye la colocación de un revestimiento vinílico de pared autoadhesivo con unas medidas no ...</small></p>	<p><b>Colocar vinilo grande:</b> <span style="float: right;">0</span></p> <p><small>Incluye la colocación de un revestimiento vinílico de pared autoadhesivo con unas medidas no ...</small></p>
<p><b>Colocar cenefa adhesiva:</b> <span style="float: right;">0</span></p> <p><small>Incluye la colocación de una cenefa adhesiva de medidas aproximadas 500 x 13,25 cm en una sola estancia.</small></p>	

#### Resumen del servicio

<b>Colgar cuadros</b>	2 cuadro (s)
<b>Tiempo contratado:</b>	1 hora.
<b>Total (IVA incluido):</b>	<b>Inferior al pedido mínimo.</b>

Necesitas alcanzar un mínimo de 58€ para hacer tu pedido

Continuar

Se realizarán trabajos hasta 3 metros de altura.

Servicio ofrecido únicamente en capitales de provincia y poblaciones de más de 100.000 habitantes con desplazamiento hasta 30km desde las mismas en las siguientes autonomías:  
Andalucía, Aragón, Islas Baleares, Cataluña, Comunidad de Madrid, Comunidad Valenciana, Región de Murcia, \*Salamanca y \*Valladolid.

Fuera de estas localizaciones se aplicarán costes adicionales por desplazamientos a razón de 0,32€ el km.

Ilustración 3. Pantalla del configurador para el servicio de manitas.

Navegando más en la página web de HOMEPPY se puede encontrar que también se dedican a la venta de productos como aire acondicionado, placas de cocción...etc.

Filtros

Marcas +

Precio +

Subcategoría -

Split pared

Aire Acondicionado

<p style="font-size: 0.8em; margin-top: 5px;">HogarClick HCW09A</p> <p style="font-size: 0.7em; margin-top: 5px;">✔ Instalación disponible</p> <p style="font-size: 0.8em; margin-top: 5px;">342,01€ IVA incluido</p>	<p style="font-size: 0.8em; margin-top: 5px;">HogarClick HCW12A</p> <p style="font-size: 0.7em; margin-top: 5px;">✔ Instalación disponible</p> <p style="font-size: 0.8em; margin-top: 5px;">359,01€ IVA incluido</p>	<p style="font-size: 0.8em; margin-top: 5px;">HogarClick HCW12A+ INSTALACIÓN BÁSICA</p> <p style="font-size: 0.7em; margin-top: 5px;">✔ Instalación incluida</p> <p style="font-size: 0.8em; margin-top: 5px;">669,00€ IVA incluido</p>
---	---	---

Ilustración 4. Lista de productos de aires acondicionados de HOMEPPY

Si nos descargamos la aplicación móvil nos encontramos con algo más directo en la pantalla principal, brindando al usuario ya las distintas áreas en las que la empresa trabaja.



*Ilustración 5. Pantalla principal de la aplicación móvil.*

Como se puede observar en la pantalla principal, concretamente en la parte derecha del header, se encuentra un botón para un chat, este botón te lleva a una página externa llamada 'Tawk To'. 'Tawk To' es un software de live chat, básicamente es un servicio de chat para que el usuario pueda hablar con los profesionales de la empresa sujeta a HOMEPPY.



\* Nombre

\* Correo electrónico

\* Mensaje

Enviar

We're ⚡ by tawk.to

Ilustración 6. Página del Tawk To al presionar el botón de chat.

Si nos adentramos a alguna de las áreas que se muestran en la pantalla principal se puede observar que te brindan subáreas relacionadas con el área principal seleccionada y tras haber entrado en alguna subárea la aplicación nos muestra todos los servicios que ofrecen al usuario.



Ilustración 7. Pantalla de subareas del área principal 'Electricidad'.



Ilustración 8. Lista de servicios disponibles del área 'Electricidad'

En definitiva, estamos ante una aplicación similar a la página web, hay menos información hacia el usuario y va directo al uso del servicio que ofrecen, esto es comprensible debido a que puede que el usuario no quiera tener tanto texto delante de él y poder pedir un servicio (ya sea avería o instalación) lo más rápido posible.

### 2.2.2. Wolly

Wolly es un servicio de reparación e instalación en el que un usuario encontrará una lista de profesionales listos para ayudar en lo que necesite dicho usuario. El procedimiento para que un usuario encuentre un profesional es el siguiente:

- a) El usuario tiene que escoger entre todos los servicios que proporciona Wolly.

## ¿Qué servicio necesitas?

Q Busca el servicio

 <b>Manitas</b> Montaje e instalación Reparaciones en casa	 <b>Cristalero</b> Instalar cristales Cambiar cristales Otros
 <b>Pintor</b> Pintar exterior vivienda o local Pintar interior vivienda o local Pintar habitación Quitar o poner gotelé Quitar papel o empapelar Otro	 <b>Suelo</b> Instalación de suelo Reparaciones de suelo Lijar y barnizar suelo Otros
 <b>Fontanero</b> Montaje o instalación Desatascar tuberías	 <b>Piscina</b> Limpieza de piscina Cambiar depuradora Reparar depuradora

Ilustración 9. Listado de servicios proporcionados por Wolly.

- b) Una vez escogido el servicio, se le proporciona al usuario más opciones específicas relacionadas con el servicio a elegir.

## ¿En qué estancia necesitas el servicio?

Selecciona una de las opciones para aportar más información al profesional.

- Dormitorio o salón
- Baño
- Cocina
- Jardín

*Ilustración 10. Listado con servicios más específicos relacionados con Montaje e instalación.*

- c) Tras haber escogido varios detalles relacionados con el servicio elegido el usuario tendrá que escribir detalles significativos para encontrar profesionales acordes a lo que pide el usuario.

### Danos más detalles

Da al profesional detalles significativos que creas que puedan ayudarle a entender mejor lo que necesitas.

Escribe algunos detalles sobre tu solicitud

Faltan 40 caracteres

*Ilustración 11. Formulario para detallar más el servicio del cliente.*



Finalmente, tras haber seguido todos los pasos se mandará la petición al servidor y el usuario recibirá varias ofertas de profesionales acordes a lo que haya pedido, con una serie de presupuestos que se podrán negociar entre cliente y profesional.

Si el acuerdo entre usuario y profesional se finaliza y el servicio se completa, el usuario pagará al profesional a través de la herramienta que proporciona Wolly llamada WollyPay, además del pago el usuario podrá opinar sobre el trabajo del profesional.

### **2.3. Propuestas de mejora**

La principal mejora que se propone para diferenciarnos de otras aplicaciones web es realizar la aplicación de manera híbrida, con esta decisión quería innovarme y adecuar mi aplicación a los tiempos que corren hoy en día con la invención y la utilización de nuevos frameworks.

Otra propuesta de mejora ante otras aplicaciones es la expansión de tipos de averías que un usuario normal se podría encontrar. Como he comentado antes, la aplicación HOMEPPY se encarga de 4 tipos de averías en general (fontanería, electricidad, cerrajería y electrodomésticos). El principal problema de esto es que puede darse el caso de que algún usuario tenga cierto tipo de avería que no dé cabida a lo que proponen algunas aplicaciones, por eso he implementado una lista de categorías más amplias para que no sucedan cosas como las que he mencionado antes.

Como se ha comentado en uno de los aspectos del análisis de HOMEPPY, su chat no es de ellos si no que usan un software externo llamado Tawk.to, en mi aplicación he creado un sistema de chat propio donde tanto las salas entre usuario y profesional como los mensajes de los usuarios se guardarán en la base de datos Firebase (concretamente Cloud Firestore). Creo que tener un sistema de chat propio es más beneficioso en un futuro ya que se podría dar el caso de alguna trifulca entre usuario y profesional y se necesiten pruebas de que se haya producido eso, el administrador de la aplicación mirará en la base de datos para comprobar el registro de mensajes entre ese usuario y profesional.

### 3. HIPÓTESIS DE TRABAJO

En este apartado se explicará todo lo relacionado con el software y el hardware usado en el trabajo, incluyendo también todo tipo de lenguajes de programación, librerías, launchers, bases de datos, software de apoyo...etc. A continuación, se especificará para que se ha utilizado cada recurso.

#### 3.1. Tecnologías y hardware utilizado

##### 3.1.1. Software, lenguajes de programación

Englobaremos en este apartado todo tipo de programas utilizados, así como lenguajes de programación usados, bases de datos y lanzadores.

##### 3.1.1.1. React Native



*Ilustración 14. Logo de React Native.*

React Native [1] es una tecnología creada por Facebook y lanzada al mundo el 26 de marzo de 2015 y cuya función principal es el desarrollo de aplicaciones destinadas a Android, iOS, aplicaciones web o las aplicaciones de televisión pertenecientes a los sistemas operativos mencionados anteriormente

Como se ha mencionado antes, React Native es capaz de desarrollar aplicaciones nativas tanto para Android como para iOS utilizando principalmente el lenguaje de programación JavaScript. Podríamos decir que React Native actúa de puente entre los módulos nativos de Android e iOS y lo que escribamos con JavaScript ya que lo que hace RN con JavaScript es transformar los elementos creados con JavaScript para renderizarlos con código nativo de Android e iOS, consiguiendo así abarcar una variedad de plataformas usando solo React Native.

Con React Native no habrá ningún problema con la experiencia del usuario ya que el framework provee una serie de componentes como View, Text o Image que se renderizan directamente en los bloques de construcción de la interfaz de usuario.

### 3.1.1.2. NodeJS



*Ilustración 15. Logo de NodeJS.*

NodeJS [2] es un entorno de ejecución asíncrono de código escrito en JavaScript para la capa del servidor. La principal característica que tiene NodeJS es que se está usando JavaScript en la capa del servidor cuando JS siempre trabaja desde la capa del cliente, el desarrollador y creador de NodeJS, Ryan Dahl quiso cambiar esto y aprovechar JS para que sea útil en la creación de programas y aplicaciones altamente escalables.

Para seguir explicando NodeJS hay que hablar brevemente de V8. V8 es el entorno de ejecución de JavaScript para Google. Ryan Dahl supo ver las ventajas que poseía el motor de Google como, por ejemplo, su gran mecánica de optimización de código y también la capacidad de poder ser ejecutado de manera independiente de cualquier navegador. También tuvo en cuenta la escalabilidad que podría tener JavaScript si se usaba fuera de la capa del cliente por lo que Ryan usó V8 para la creación de NodeJS.

Además de V8, NodeJS usa una variedad de librerías para que funcione de la manera en la que funciona, como, por ejemplo:

- Libuv: Esta librería es usada para abstraer las operaciones de entrada y salida no bloqueantes a una interfaz consistente a través de todas las plataformas posibles.
- Lhttp.
- C-ares: Utilizada para algunas peticiones DNS asíncronas.

- OpenSSL: Utilizada para la implementación de funciones de criptografía que son necesarias en la navegación de la web en estos tiempos.
- Zlib: Utilizada para la creación de interfaces de compresión y descompresión síncronas, asíncronas.

NodeJS también una serie de herramientas para su correcto funcionamiento, estos son:

- Gyp: gyp es una herramienta basada en Python de automatización para la generación de archivos.
- Gtest: Esta herramienta es utilizada para el testeo de código nativo, también permite el testeo de código en C/C++.

### 3.1.1.3. Npm



*Ilustración 16. Logo de npm.*

Conectando con el capítulo de NodeJS, toca hablar de una de las herramientas más importantes que posee, siendo esta npm. Npm [3] es el registro de software más grande del mundo donde desarrolladores de código abierto de todo tipo usan esta herramienta para compartir y usar paquetes. Esta herramienta también es utilizada por varias organizaciones para gestionar el desarrollo privado de sus entornos.

Npm está compuesto por tres componentes importantes:

- La página web: En la página web el usuario podrá descubrir todo tipo de paquetes, podrá establecer perfiles y podrá gestionar otros aspectos de la experiencia de usuario de npm.
- La interfaz de comando de línea o CLI: Es el componente más importante de npm ya que es donde prácticamente todos los usuarios interactúan con npm a través de la terminal que proporciona.

- El registro: Es una base de datos enorme de software donde hay una cantidad inmensa de paquetes para todos los usuarios.

Algunos usos que se le pueden dar a npm son: Añadir paquetes a las aplicaciones para adaptarlas o incorporarlas para sumar más funcionalidad a dicha aplicación, poder ejecutar paquetes sin tener que descargarlos usando npx, compartir código con cualquier usuario de npm desde cualquier sitio y, por último, gestionar múltiples gestiones del código y también de las dependencias de dicho código.

#### 3.1.1.4. Expo



Expo [4] es un framework de código abierto y también una plataforma para la creación de aplicaciones nativas para Android, iOS y web con JS y React. Expo se descompone en una serie de herramientas y servicios basadas en React Native y en plataformas nativas que ayudará al usuario a desarrollar, construir, desplegar y poder rotar rápidamente en Android, iOS y aplicaciones web partiendo de la misma base de código basada en JavaScript y TypeScript.

Otra característica de Expo es que el usuario no tendrá ningún problema si la aplicación enviada a internet tiene algún tipo de fallo ya que con comandos de expo podrá desplegar un arreglo rápido cuyos usuarios podrán ver al instante, sin tener que descargarse otra vez la aplicación.

Con el comando “expo start” en el CLI de NodeJS abriremos las herramientas de desarrollo de la aplicación y desde esas herramientas el usuario podrá visualizar la aplicación de React Native como aplicación web y también como aplicación para Android

e iOS, descargándose la aplicación de Expo para móvil y leyendo el código QR que proporciona la página de las herramientas de desarrollo.

### 3.1.1.5. Firebase y Cloud Firestore



*Ilustración 18. Logo de Firebase.*

Firestore [5] es una plataforma dedicada al desarrollo de aplicaciones web y móvil creada en 2011 y adquirida por la compañía Google en el año 2014. Firestore está situada en la nube, dándole al usuario ventajas como la máxima optimización a la hora de la creación y sincronización de proyectos.

Las ventajas principales de Firestore son la rápida sincronización de los datos sujetos a un proyecto sin llegar a tener que escribir algún tipo de lógica perteneciente a otro tipo de sincronización compleja. Otra ventaja principal es que el usuario puede crear proyectos sin la necesidad de usar ningún tipo de servidor ya que las herramientas que se encargan de ese tipo de funcionalidad se incluyen en los SDK de Firestore. Como Firestore es una herramienta de Google usa su infraestructura por lo que obtiene una escalabilidad automática para todo tipo de aplicación.

Gracias a Firestore tenemos herramientas como Cloud Firestore. Cloud Firestore [6] es una base de datos de tipo NoSQL que está alojada en la nube y a la que pueden acceder aplicaciones de todo tipo gracias a los SDK.

¿Qué significa que sea de tipo NoSQL? Cloud Firestore trabaja con una serie de elementos llamados colecciones, que sería el equivalente a las tablas en un sistema de base de datos de tipo SQL.

Estas colecciones están compuestas por documentos que están identificados por un ID generado por Firebase, aunque le da la oportunidad al usuario de declarar un ID personalizado.

Es de tipo NoSQL porque no hay ningún tipo de relación entre tablas ya sea 1:1, N:1 o N:N, para relacionar documentos de colecciones entre si habrá que crear una colección aparte incluyendo los ID de los documentos a relacionar.

Las principales ventajas que se obtienen a la hora de usar Cloud Firestore son:

- **Flexibilidad:** El modelo de Cloud Firestore permite estructuras de datos flexibles. El usuario puede almacenar los datos en documentos y dichos documentos pueden obtener objetos anidados como subcolecciones, compuestas por otros documentos.
- **Consultas expresivas:** Con Cloud Firestore el usuario podrá recuperar datos de un documento en específico o de toda una colección con los parámetros que le pase en la consulta. Dichas consultas también podrán contener filtros individuales o encadenar varios filtros.

```
const getTrabajosTerminados = async() => {
  firebase.firestore().collection('tratos')
    .where('tratoTerminado', '=', 'si')
    .where('tratoCliente', '=', uid)
    .get()
    .then(querySnapshot => {
      const itemsTerminado = [];
      querySnapshot.forEach((doc) => {
        itemsTerminado.push(doc.data());
      })
      setTratosTerminados(itemsTerminado)
    })
}
```

Ilustración 19. Consulta con una cadena de filtros.

- **Actualizaciones en tiempo real:** Cloud Firestore usa la sincronización de datos para la actualización de datos de cualquier dispositivo.

- Asistencia sin conexión: Firestore almacena en caché los datos que usa la aplicación de forma activa por lo que dicha aplicación puede hacer consultas de escritura, lectura y consultar datos, aunque el dispositivo no tenga conexión. Una vez que el dispositivo vuelva a tener conexión Firestore sincroniza todos los cambios realizados anteriormente.

### 3.1.1.6. Componentes usados de React Native

El directorio de React Native provee al usuario una variedad de componentes que le puede servir para enriquecer la aplicación...etc. En este trabajo de fin de grado he usado una serie de componentes que me han ayudado al desarrollo de la aplicación, los componentes usados más importantes son:

#### 3.1.1.6.1. React Native Gifted Chat



*Ilustración 20. Componente React Native Gifted Chat.*

Este componente [7] es principalmente un sistema de chat entre usuarios. Explicando brevemente como funciona se guardan los mensajes del chat en un array de objetos compuestos por el texto del mensaje, el id del usuario, cuando se crea el mensaje y el usuario con quien habla.

Para instalar este componente el usuario puede usar npm o yarn, en mi caso he usado npm con el comando “npm install react-native-gifted-chat –save”. En la documentación que ofrece el creador del componente en GitHub se puede encontrar todo tipo de información relacionada con los props (funciones) del componente.

### 3.1.1.6.2. React Native Ratings



*Ilustración 21. Componente React Native Rating*

Este componente [8] tiene la principal función de dar una valoración sobre 5, también incluye por defecto una UI compuesta por estrellas, aunque se puede customizar mediante un prop del componente. Otros props o funciones destacables de este componente son la capacidad de poder dar valores en decimales, siendo el mínimo 2 decimales y el máximo 20. El valor de la valoración se recogerá en un prop llamado onFinishRating, mediante ese prop el usuario podrá transferir dicho valor a una variable para un posterior manejo.

### 3.1.1.6.3. React Native Picker

React Native Picker [9] es un componente para implementar un menú desplegable en la aplicación, este componente es compatible con Android, iOS, Windows y macOS. En la documentación de este componente se puede encontrar lo que se puede hacer con él, hay que tener en cuenta de que algunas funciones que brinda el componente solo son compatibles para Android como por ejemplo las funciones blur o focus que cierran y abren el componente de manera automática.

### 3.1.2. Hardware

Para el desarrollo de este trabajo de fin de grado se ha usado un ordenador de sobremesa, teclado, monitor, ratón y un móvil. A continuación, se ilustrarán las especificaciones de dichos componentes.

- Ordenador de sobremesa.
  - Marca: MSI.
  - Modelo: MSI Infinite 8RB-618XES.
  - Procesador: Intel® Core™ i5-8400 (Desde 2.8GHz hasta 4.0GHz).
  - Memoria RAM: 1 stick de 8GB de tipo DDR4 con una velocidad de 2666MHz.
  - Disco duro de 1 TB más un SSD de 256 GB.
  - Gráfica: NVIDIA® GeForce® GTX 1050Ti 4GB ITX GDDR5.
  - Sistema operativo: Windows 10 Home
- Teclado.
  - Marca: NEWSKILL.
  - Modelo: Newskill Serike TKL.
- Monitor.
  - Marca: ASUS.
  - Modelo: ASUS VG249Q.
- Ratón.
  - Marca: Hiditec.
  - Modelo: Hiditec GX10.
- Móvil.
  - Marca: Xiaomi.
  - Modelo: Xiaomi Redmi Note 9S.
  - Procesador: Snapdragon 720G.
  - Memoria RAM: 6 GB.
  - Memoria interna: 64 GB.
  - Sistema Operativo: Android 10 con capa de personalización MIUI 12.
  - Batería: 5020 mAh.
  - Cámara frontal: 16 MP.
  - Cámara trasera: 48 MP.

## 4. DESARROLLO

### 4.1. Desarrollo de la herramienta. Requerimientos

#### 4.1.1. Planteamiento del problema

Para empezar a desarrollar una aplicación hay que tener un método a seguir, hay varios métodos como los que hay en el campo de ingeniería de software (Modelo en V, modelo en cascada...etc.)[10] pero también hay otros métodos a utilizar, como es el que se ha utilizado en este proyecto, llamado “proceso de prototipado efectivo” [11].

Dicho método tiene 4 fases, se citan a continuación:

- Planificación.
- Especificación.
- Diseño.
- Resultados.

Este método me ha resultado idóneo y práctico para seguir ya que sus 4 fases siguen una lógica sencilla a la hora de tener que desarrollar un proyecto de desarrollo software.

#### 4.1.2. Esquema organizativo

El diagrama estructural que sigue el “Proceso de prototipado efectivo” se organiza en 4 fases principales que se subdividen en 11 subfases como se observa en la siguiente figura:.



Ilustración 22. Diagrama del proceso de prototipado efectivo.

En la primera fase del método nos encontramos con la fase de planificación. Esta fase se centra en la verificación de requerimientos (mencionados en el punto 4.1.3), la creación de un flujo de tareas, es decir, como va a navegar el usuario de una pantalla a otra, conocer que pasa cuando le da a un botón.

La verificación de requerimientos se realiza mediante unos worksheets que se completan tras varias reuniones. En estos worksheets se enumerarán los requerimientos recogidos en las primeras reuniones y se validarán o no dependiendo de su importancia, de si son prescindibles o no para la aplicación, si aportan funcionalidad o usabilidad ante el usuario...etc. Como ya se mencionará en el apartado 4.1.3, ha habido varios requerimientos que se han rechazado debido a que no cumplían algunas de las pautas mencionadas a la hora de verificar un requerimiento.

En la segunda fase de este método está la fase de especificación, donde se discutirá las características del prototipo, el método a seguir para el prototipo y con qué herramienta se creará dicho prototipo.

Las características a destacar son: Audiencia, estado, velocidad, longevidad, expresión, estilo, medio y fidelidad. Mediante estas características se podrá averiguar para quien va a ser el producto, el estilo del producto, si va a ser un producto a largo plazo o para una función en concreto...etc. En cuanto al método a seguir para el prototipo y la herramienta para crearlo yo me he decantado con el método del prototipo codificado y la herramienta VSCode.

Siguiendo la fase de especificación, tenemos la fase de diseño, aquí es donde se diseña la interfaz del usuario según el criterio que se ha especificado en la fase 1 de requerimientos.

A continuación, voy a insertar algunos bocetos que hice a lo largo del desarrollo de la aplicación para crear el diseño final de ella.

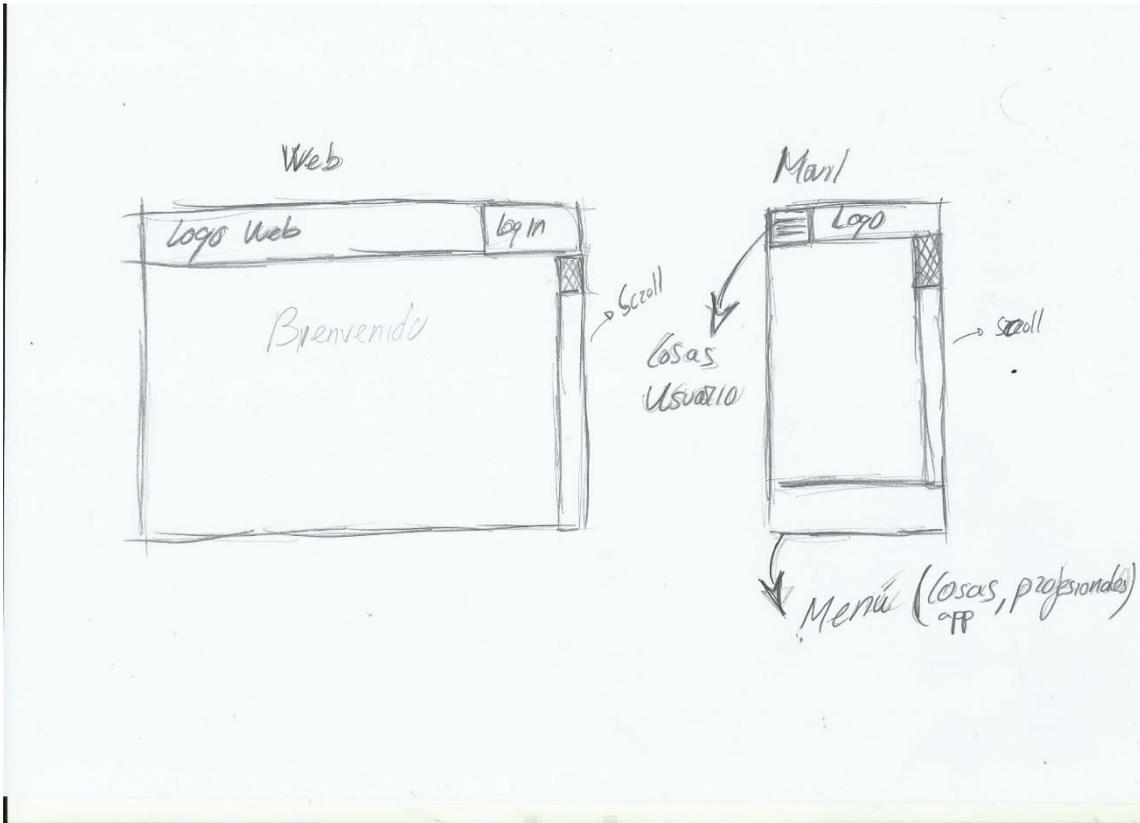


Ilustración 23. Primeros diseños de la aplicación.

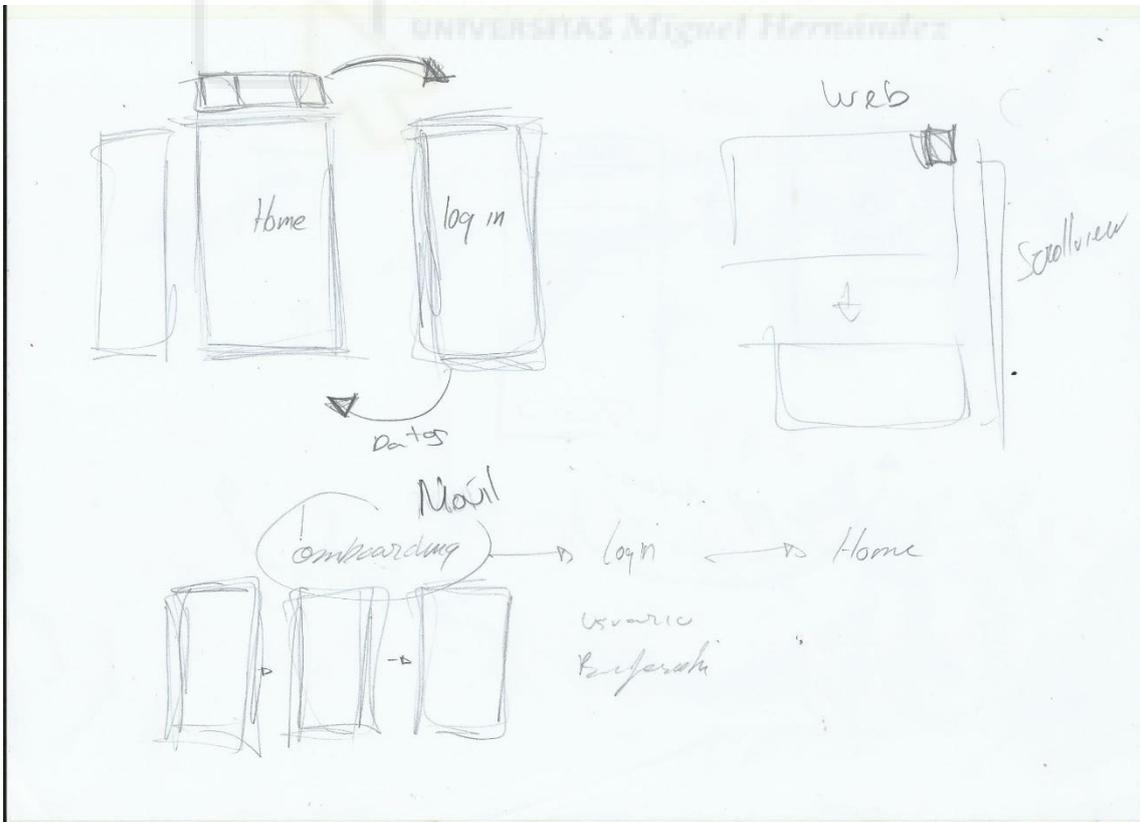


Ilustración 24. Diseño mencionando el elemento onboarding de la aplicación.

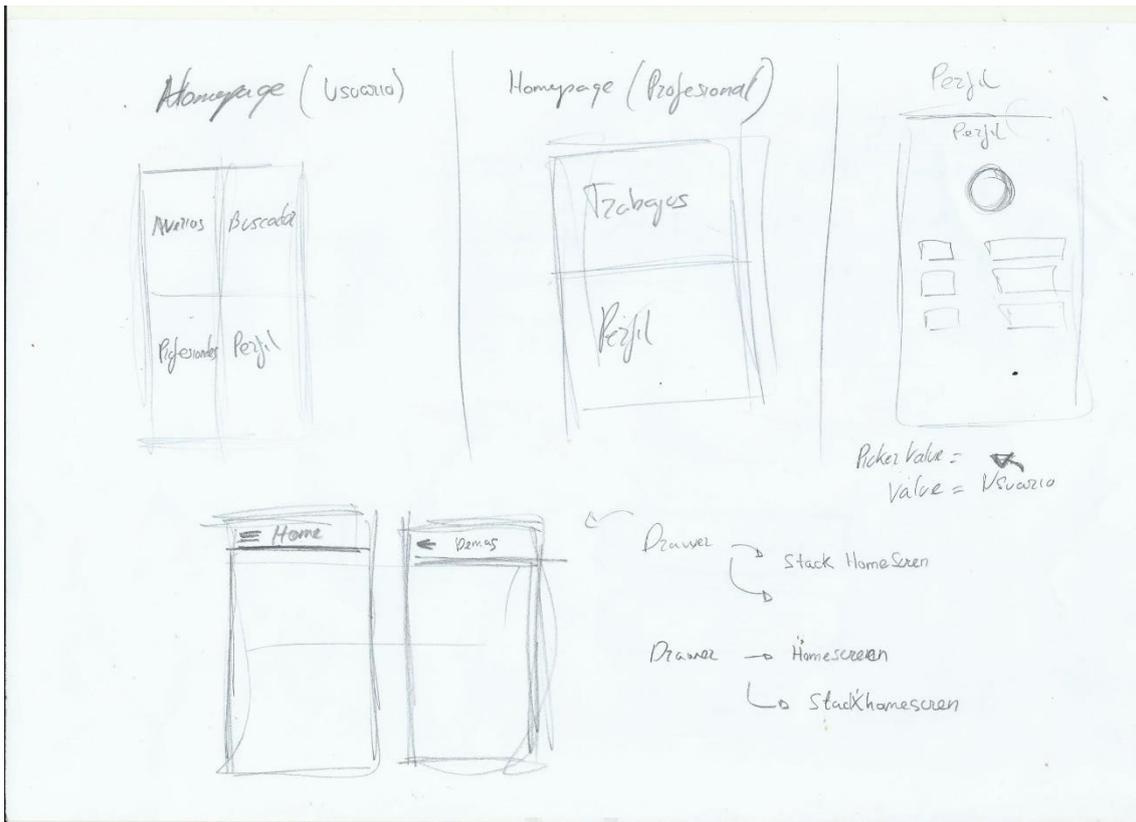


Ilustración 25. Diseño de la homepage para el cliente y el profesional, diseño del perfil y mención al elemento drawer.

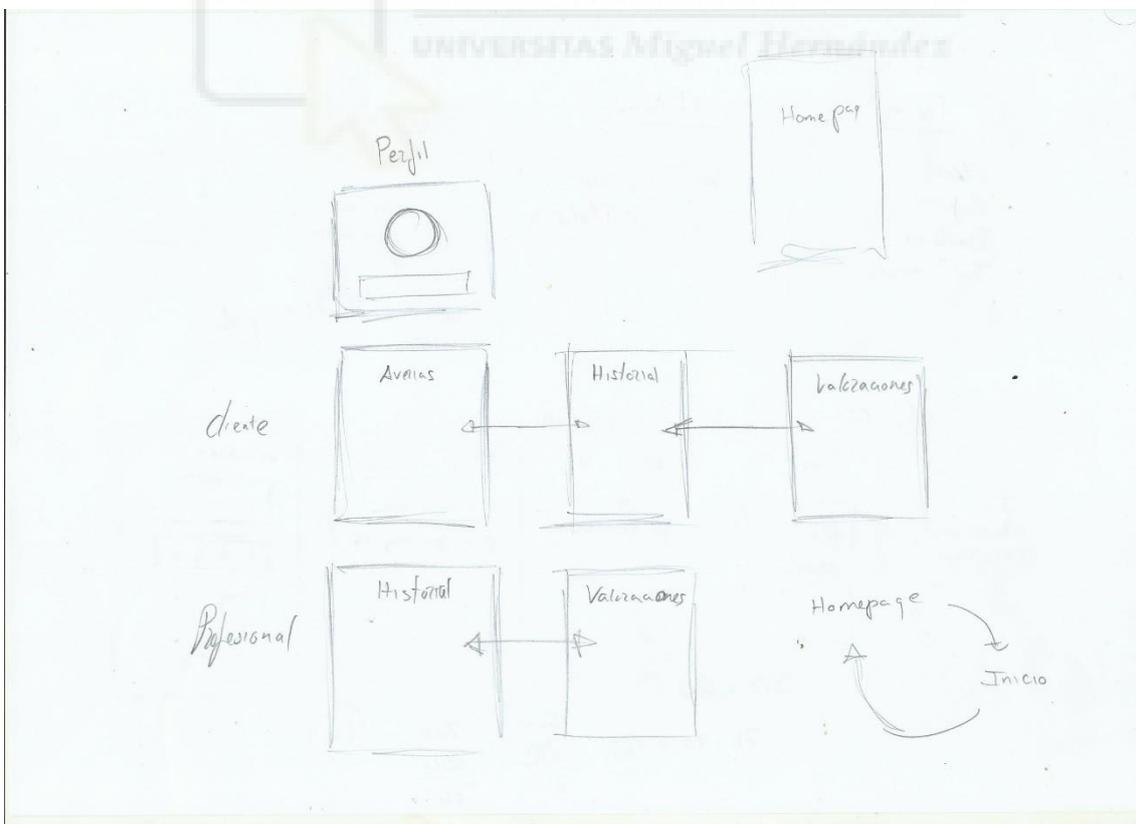


Ilustración 26. Diseño de la parte inferior del perfil del cliente y del profesional.

Por último, tenemos la fase de resultados, en esta fase se valida el diseño y se implementa en algo real, siendo esto la aplicación final del trabajo de fin de grado.

### **4.1.3. Análisis de requerimientos**

En este apartado se explicará el análisis y la selección de los requerimientos mencionados en varias reuniones que se han realizado a lo largo del desarrollo de la aplicación.

Tras las primeras reuniones, se han ido estableciendo los siguientes requerimientos:

1. Interfaz amigable para el usuario.
2. Elemento drawer que permitirá al usuario navegar por toda la página sin la necesidad de ir a una sección en concreto.
3. Que la interfaz sea usable e intuitiva.
4. Todos los tipos de usuario podrán actualizar los datos de su cuenta sin ningún tipo de límite.
5. Mostrar mapas de ubicaciones.
6. Se mostrará un ranking de profesionales, habrá un filtro para mostrarlos según la valoración y el número de averías completadas.
7. Habrá una lista de categorías con su descripción y una lista de subcategorías donde el usuario podrá encontrar la categoría adecuada.
8. Mostrar una lista de profesionales recomendados tras haber elegido una categoría.
9. Conocer al profesional mediante su perfil en el caso de que el usuario lo elija de la lista mencionada en el punto anterior.
10. Mostrar en el perfil del profesional las insignias ganadas desde su registro en la aplicación.
11. Abrir un chat entre el usuario y el profesional presionando el botón de chat del perfil del profesional.
12. Sistema de chat entre cliente y profesional.
13. Cerrar tratos desde la pantalla de chat mediante un botón.
14. Poder editar el título del trato desde la pantalla de perfil del usuario cliente.
15. Cerrar el trato presionando varios segundos la tarjeta del trato a cerrar.
16. Valorar al profesional con una opinión y una valoración numérica del 0 al 5.
17. Métodos de pago virtuales.
18. El administrador podrá examinar y modificar datos de usuario.
19. El administrador podrá desactivar o borrar cuentas de usuarios.

20. El administrador podrá modificar datos de tratos.
21. El administrador podrá cancelar tratos entre clientes y profesionales.
22. Solo el usuario cliente será capaz de navegar por la lista de categorías y la lista de profesionales de dichas categorías.
23. Solo el usuario cliente será capaz de navegar por el ranking de profesionales.
24. La valoración entre usuario y profesional solo se podrá realizar tras terminar el trato y solo podrá valorarle el usuario cliente que haya creado el trato.

Tras reuniones posteriores, se han ido modificando varios requerimientos que no se han pensado como necesarios para el funcionamiento y el desarrollo de la aplicación. Algunos de estos requerimientos son como, por ejemplo, mostrar el mapa de ubicaciones, este requerimiento no se ha considerado del todo importante ya que el usuario puede buscar la ubicación de otro usuario mediante una simple búsqueda en un motor de búsqueda como puede ser Google.

Otro de los requerimientos tumbados en dichas reuniones es el tema de los métodos de pago virtuales, he creído conveniente rechazar este requisito debido a que se puede dar el caso de que un cliente no esté familiarizado con dichos métodos de pago virtuales y puede traer a discusiones entre cliente y profesional.

A continuación, voy a ilustrar un worksheet que creo conveniente a la hora de explicar el análisis y la selección de requisitos.

Nombre de la Tarea	Descripción	Requerimientos	Base Fundamental	Pasos Previos	Pasos Posteriores
Iniciar sesión / Registro	Iniciar sesión en la aplicación o registrarse si no lo está	No está sujeto a ningún requisito	Antes de navegar por la aplicación hay que registrarse en ella o si ya se está registrado, iniciar sesión	Ninguno	Acceder pantalla principal
Acceder pantalla principal	Acceder a la pantalla principal de la aplicación	3	Es el núcleo de la aplicación, a partir de aquí el usuario puede navegar por toda la aplicación	Iniciar sesión / Registro	Consultar perfil, consultar ranking, consultar lista de categorías, consultar chats
Consultar perfil	Acceder a la sección de la aplicación relacionada con nuestro perfil	No está sujeto a ningún requisito	El usuario podrá consultar todos los datos que introdujo al registrarse en la aplicación	Acceder pantalla principal	Editar datos
Editar datos	Acceder a la sección de nuestro perfil y cambiar algún dato	4	El usuario podrá cambiar cualquier dato que haya insertado a la hora de registrarse	Consultar perfil	Ninguno
Consultar ranking	Acceder a la sección de la aplicación relacionada con el ranking de profesionales	6	El usuario podrá consultar el ranking de profesionales de la aplicación, también podrá aplicar filtros a dicho ranking	Acceder pantalla principal	Ninguno
Consultar lista de categorías	Acceder a la sección de la aplicación relacionada con la lista de categorías	7	El usuario podrá consultar la lista de categorías que le brinda la aplicación para poder encontrar la solución a su avería	Acceder pantalla principal	Consultar profesionales recomendados
Consultar profesionales recomendados	Acceder a la lista de profesionales recomendados de una categoría en específico	8	El usuario podrá encontrar una lista de profesionales recomendados por la aplicación, ordenados con la valoración más alta	Consultar lista de categorías	Consultar perfil del profesional
Consultar perfil del profesional	Acceder al perfil del profesional	9 y 10	El usuario podrá consultar el perfil del profesional, donde encontrará sus valoraciones recibidas y un botón para chatear con él	Consultar profesionales recomendados	Chatear con el profesional
Chatear con el profesional	Hablar con el profesional mediante el chat de la aplicación	12	Se podrá mantener una conversación entre el cliente y el profesional para acercar posturas y poder cerrar tratos.	Consultar perfil del profesional	Crear un trato con el profesional
Crear un trato con el profesional	Crear un trato tras haber hablado con el profesional	13	Tras haber hablado con el profesional, el usuario decide crear un trato con él.	Chatear con el profesional	Cerrar trato
Cerrar trato	Cerrar trato tras haber arreglado la avería	15	Tras haber arreglado la avería, el cliente cierra el trato con el profesional y lo valora con su opinión y una nota	Crear un trato con el profesional	Ninguno

Tabla 1. Worksheet 4.1.

## 4.2. Desarrollo de la herramienta. Fase de diseño

### 4.2.1. Ciclo de ingeniería del software

#### 4.2.1.1. Descripción de actores.

Para empezar este apartado relacionado con el ciclo de ingeniería del software, lo necesario es describir que tipos de usuarios hay en la aplicación, estos son:

<b>Actor</b>	<b>Usuario no registrado</b>
<b>Descripción</b>	Un usuario no registrado es cualquier tipo de persona que entre a la aplicación y no esté registrada, no puede hacer ningún tipo de actividad como navegar por la pantalla principal de la aplicación o hacer tratos con profesionales.

*Tabla 2. Actor Usuario no registrado.*

<b>Actor</b>	<b>Cliente (Rol de usuario)</b>
<b>Descripción</b>	Un usuario de tipo cliente es el que se haya registrado con la meta de buscar a un profesional en concreto para el arreglo de alguna avería que tenga. Este usuario podrá chatear con otros usuarios de tipo profesional y cerrará tratos. También podrá valorar al usuario profesional dependiendo del trabajo que haya realizado dicho usuario profesional.

*Tabla 3. Tabla Cliente (Rol de usuario)*

<b>Actor</b>	<b>Profesional (Rol de usuario)</b>
<b>Descripción</b>	Un usuario de tipo profesional es el que se haya registrado con la meta de darse a conocer por sus cualidades a la hora de arreglar averías de ciertas categorías. El usuario profesional podrá chatear con el usuario cliente para cerrar posibles tratos.

*Tabla 4. Tabla Profesional (Rol de usuario)*

<b>Actor</b>	<b>Administrador</b>
<b>Descripción</b>	Un administrador es el tipo de usuario que tiene control absoluto sobre la aplicación. Desde el panel de control de Firebase podrá manejar todo tipo de material relacionado con la aplicación como, por ejemplo: Control de usuarios, logs de las salas de chat, comprobar los tratos hechos y por terminar...etc.

*Tabla 5. Tabla Administrador*

#### 4.2.1.2. Subsistemas y diagramas de casos de uso

Tras haber explicado los actores de la aplicación se ilustrarán los subsistemas que existen en ella, con sus correspondientes casos de uso, estos casos de uso siguen el modelo UML. [12]

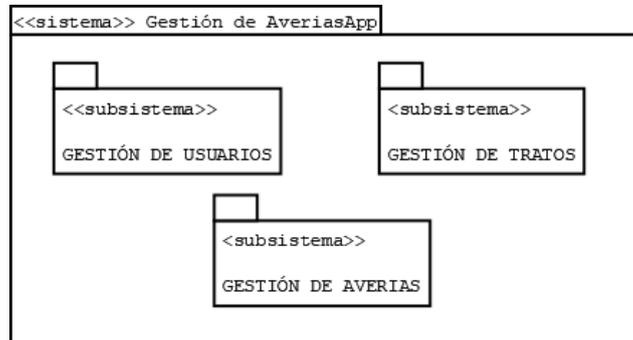


Ilustración 27. Subsistemas de AveríasApp.

A continuación, se ilustrarán los casos de uso relacionados con cada subsistema, empezando con el subsistema de gestión de usuarios.

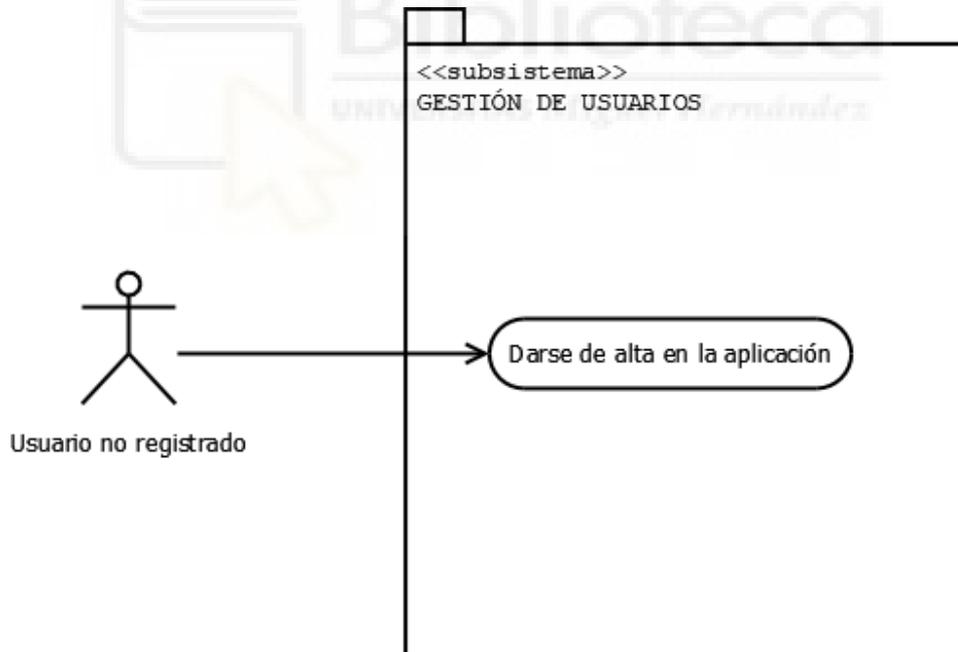


Ilustración 28. Casos de uso del usuario no registrado en el subsistema de gestión de usuarios.

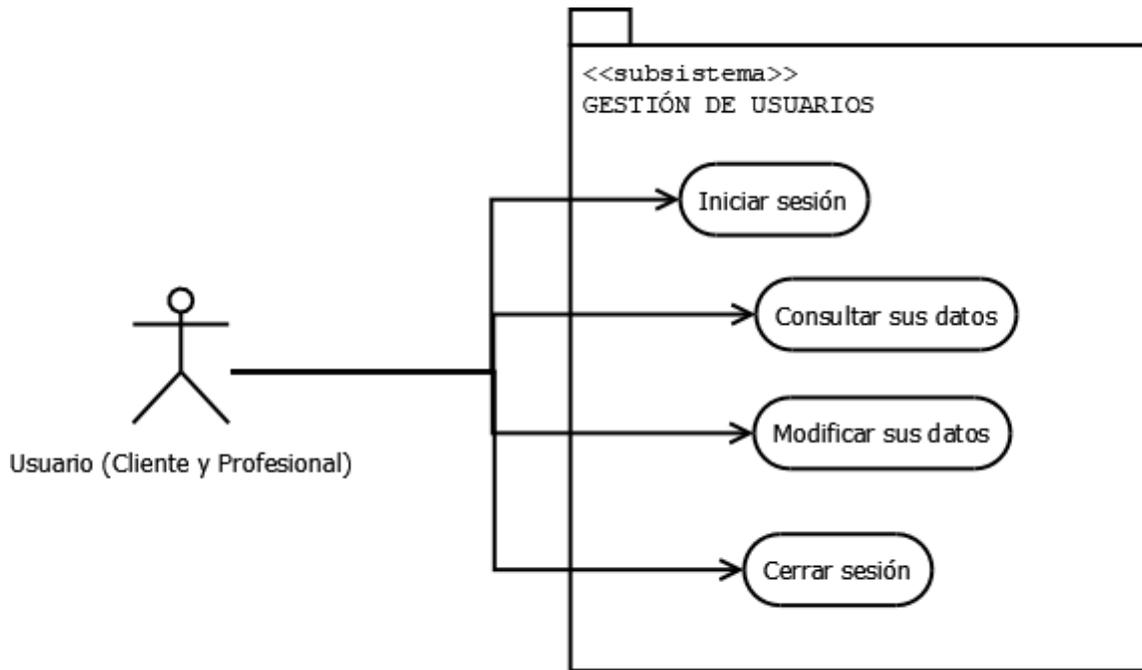


Ilustración 29. Casos de uso del usuario en el subsistema de gestión de usuarios.

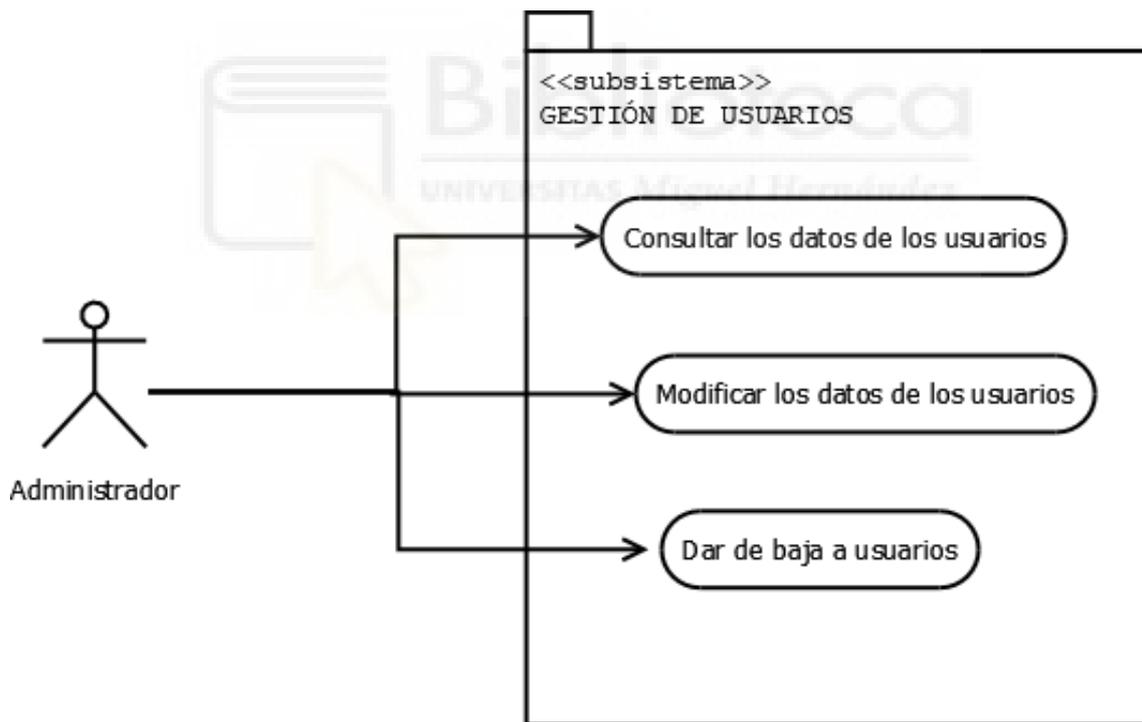


Ilustración 30. Casos de uso del administrador en el subsistema de gestión de usuarios.

Tras haber ilustrado los casos de uso del subsistema de gestión de usuarios, continua la descripción de los casos de uso con sus apartados correspondientes.

<b>C.U.1.1</b>	Darse de alta en la aplicación.
<b>Actores</b>	Usuario no registrado.
<b>Descripción</b>	Este tipo de usuario necesita usar los servicios de la aplicación por lo que decide darse de alta para disfrutar todo lo que ofrece la aplicación.
<b>Dependencias</b>	No tiene dependencias.
<b>Precondición</b>	PreC1: Que la aplicación funcione. PreC2: Que el usuario no registrado tenga los datos necesarios para registrarse.
<b>Secuencia Normal</b>	P1: El usuario no registrado entra en la aplicación. P2: Navega por el elemento onboarding de la aplicación. P3: Pulsa el botón para registrarse en la aplicación P4: Introduce todos los datos que la aplicación le pregunta. P5: Tras registrarse entra en la aplicación siendo un usuario registrado.
<b>Postcondición</b>	PostC1: Pasa a ser un usuario registrado
<b>Excepciones</b>	Primera excepción: Si el usuario en el cuarto paso introduce algún dato incorrecto o se salta algún dato obligatorio, a la hora de darle al botón de registrarse la aplicación le avisará de la información que falta para completar el registro.

*Tabla 6. Caso de uso 1.1*

<b>C.U.1.2</b>	Iniciar sesión.
<b>Actores</b>	Usuario cliente y profesional.
<b>Descripción</b>	Este tipo de usuario necesita usar los servicios de la aplicación por lo que inicia sesión, en el caso del usuario cliente para encontrar un profesional acorde a su problema y en el caso del usuario profesional para hablar con los usuarios clientes y cerrar tratos.
<b>Dependencias</b>	No tiene dependencias.
<b>Precondición</b>	PreC1: Que la aplicación funcione. PreC2: Que el usuario se haya registrado en la aplicación.
<b>Secuencia Normal</b>	P1: El usuario registrado entra en la aplicación. P2: Navega por el elemento onboarding de la aplicación. P3: Introduce los datos necesarios para iniciar sesión.

	P4: Entra en la página principal de la aplicación .
<b>Postcondición</b>	PostC1: Consigue iniciar sesión y entra en la aplicación.
<b>.Excepciones</b>	Primera excepción: Si el usuario no introduce los datos necesarios o introduce algún dato incorrecto la aplicación le avisará para que revise sus datos y vuelva a introducirlos de manera correcta.

*Tabla 7. Caso de uso 1.2.*

<b>C.U.1.3</b>	Consultar sus datos.
<b>Actores</b>	Usuario cliente y profesional.
<b>Descripción</b>	Este tipo de usuario decide consultar sus datos en la sección de su perfil para comprobar que todo está correcto.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	PreC1: Que la aplicación funcione. PreC2: Que el usuario se haya registrado en la aplicación. PreC3: Que el usuario haya iniciado sesión en la aplicación.
<b>Secuencia Normal</b>	P1: El usuario registrado inicia sesión en la aplicación P2: Entra en la sección de Perfil de la pantalla principal de la aplicación. P3: Consulta los datos que se muestran en su perfil tras hacer un swipe en la parte superior de su perfil.
<b>Postcondición</b>	No hay postcondición.
<b>Excepciones</b>	No hay excepciones.

*Tabla 8. Caso de uso 1.3.*

<b>C.U.1.4</b>	Modificar sus datos.
<b>Actores</b>	Usuario cliente y profesional.
<b>Descripción</b>	Este tipo de usuario se ha dado cuenta de que tiene algún dato erróneo o decide actualizar sus datos por lo que planea editar sus datos personales.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	PreC1: Que la aplicación funcione. PreC2: Que el usuario se ha registrado. PreC3: Tener algún dato erróneo o querer actualizar otro dato.
<b>Secuencia Normal</b>	P1: El usuario registrado inicia sesión en la aplicación.

	<p>P2: Entra en la sección de Perfil de la pantalla principal de la aplicación.</p> <p>P3: Pulsa el botón para editar sus datos.</p> <p>P4: Actualiza los datos que crea necesarios.</p> <p>P5: La aplicación le confirma que los datos han sido modificados.</p>
<b>Postcondición</b>	PostC1: Los datos que haya modificado se han registrado correctamente.
<b>Excepciones</b>	Primera excepción: Si el usuario en el cuarto paso introduce algún dato con un formato incorrecto o se salta algún procedimiento como, por ejemplo, a la hora de cambiar de contraseña la aplicación le avisará de la información que debe revisar lo que esté modificando.

Tabla 9. Caso de uso 1.4.

<b>C.U.1.5</b>	Cerrar sesión.
<b>Actores</b>	Usuario cliente y profesional.
<b>Descripción</b>	Este tipo de usuario ya no necesita realizar ninguna actividad en la aplicación por lo que decide cerrar sesión.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	PreC1: Que la aplicación funcione. PreC2: Que el usuario tenga una cuenta en la aplicación.
<b>Secuencia Normal</b>	<p>P1: El usuario abre el menú lateral de la aplicación</p> <p>P2: Pulsa el botón para cerrar sesión.</p> <p>P3: La aplicación le pregunta si quiere cerrar sesión.</p> <p>P4: Si el usuario acepta cerrar sesión, se cierra sesión y lo redirige al onboarding de la aplicación.</p>
<b>Postcondición</b>	PostC1: El usuario ha cerrado sesión
<b>Excepciones</b>	No hay excepciones.

Tabla 10. Caso de uso 1.5.

<b>C.U.1.6</b>	Consultar los datos de los usuarios.
<b>Actores</b>	Administrador
<b>Descripción</b>	Este tipo de usuario necesita consultar los datos de los usuarios para comprobar que todo está correcto.
<b>Dependencias</b>	No tiene ninguna dependencia.

<b>Precondición</b>	No hay precondiciones.
<b>Secuencia Normal</b>	P1: El administrador entra en el panel de control de Firebase. P2: Entra en el apartado Cloud Firestore. P3: Entra en la colección “users.” P4: Consulta todos los documentos relacionados con la colección “users”.
<b>Postcondición</b>	No hay postcondiciones.
<b>Excepciones</b>	No hay excepciones.

Tabla 11. Caso de uso 1.6.

<b>C.U.1.7</b>	Modificar datos de los usuarios.
<b>Actores</b>	Administrador.
<b>Descripción</b>	Este tipo de usuario recibe alguna petición de un usuario para la modificación de un dato en específico. También puede darse el caso de que el administrador vea conveniente la modificación de datos por algún motivo.
<b>Dependencias</b>	No hay dependencias.
<b>Precondición</b>	No hay precondiciones.
<b>Secuencia Normal</b>	P1: El administrador entra en el panel de control de Firebase. P2: Entra en el apartado Cloud Firestore. P3: Entra en la colección “users.” P4: Entra en el documento del usuario a modificar datos. P5: Registra los datos nuevos y confirma los cambios.
<b>Postcondición</b>	PostC1: El usuario tendrá cambios en sus datos.
<b>Excepciones</b>	No hay excepciones.

Tabla 12. Caso de uso 1.7.

<b>C.U.1.8</b>	Dar de baja a los usuarios
<b>Actores</b>	Administrador.
<b>Descripción</b>	El administrador, por petición de otros clientes o por decisión propia, decide dar de baja a usuarios.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	No hay precondiciones.
<b>Secuencia Normal</b>	P1: El administrador entra en el panel de control de Firebase. P2: Entra en el apartado Cloud Firestore.

	<p>P3: Entra en la colección “users.”</p> <p>P4: Selecciona el documento del usuario a dar de baja.</p> <p>P5: Confirma el borrado del documento.</p>
<b>Postcondición</b>	PostC1: El usuario cuyo documento ha sido borrado se ha dado de baja.
<b>Excepciones</b>	No hay excepciones.

Tabla 13. Caso de uso 1.8.

A continuación, vienen las ilustraciones y los diagramas de casos de usos relacionados con el subsistema de gestión de tratos.

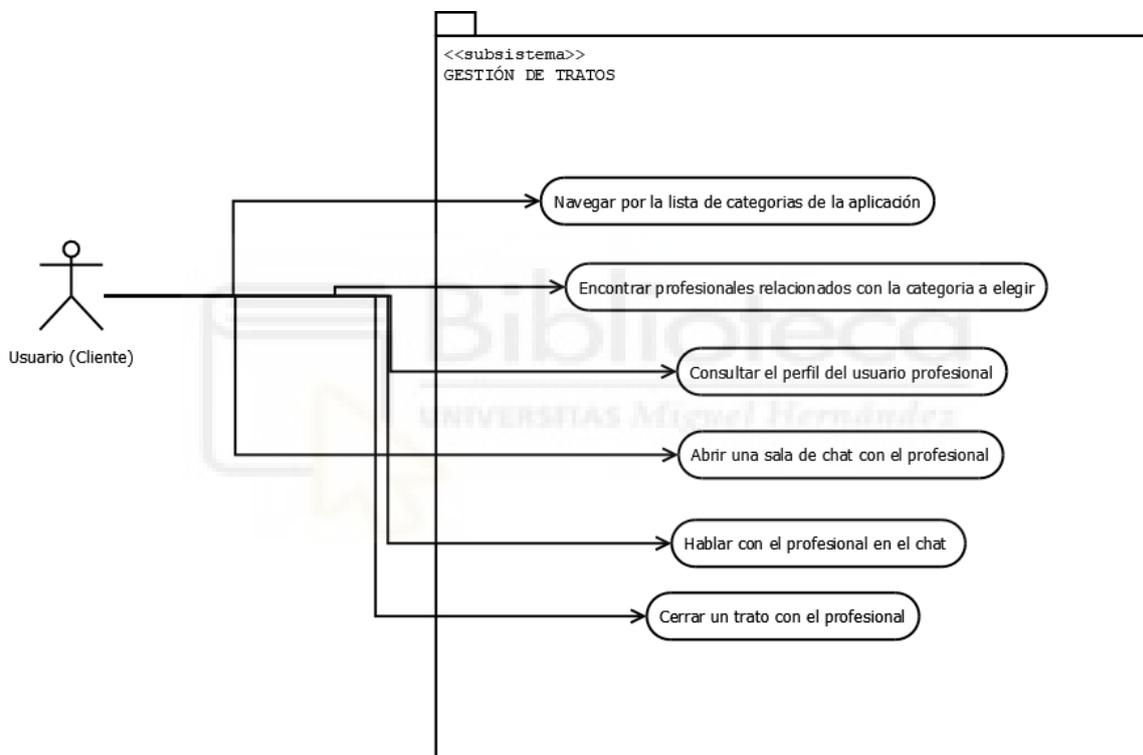


Ilustración 31. Casos de uso del usuario cliente en el subsistema de gestión de tratos.

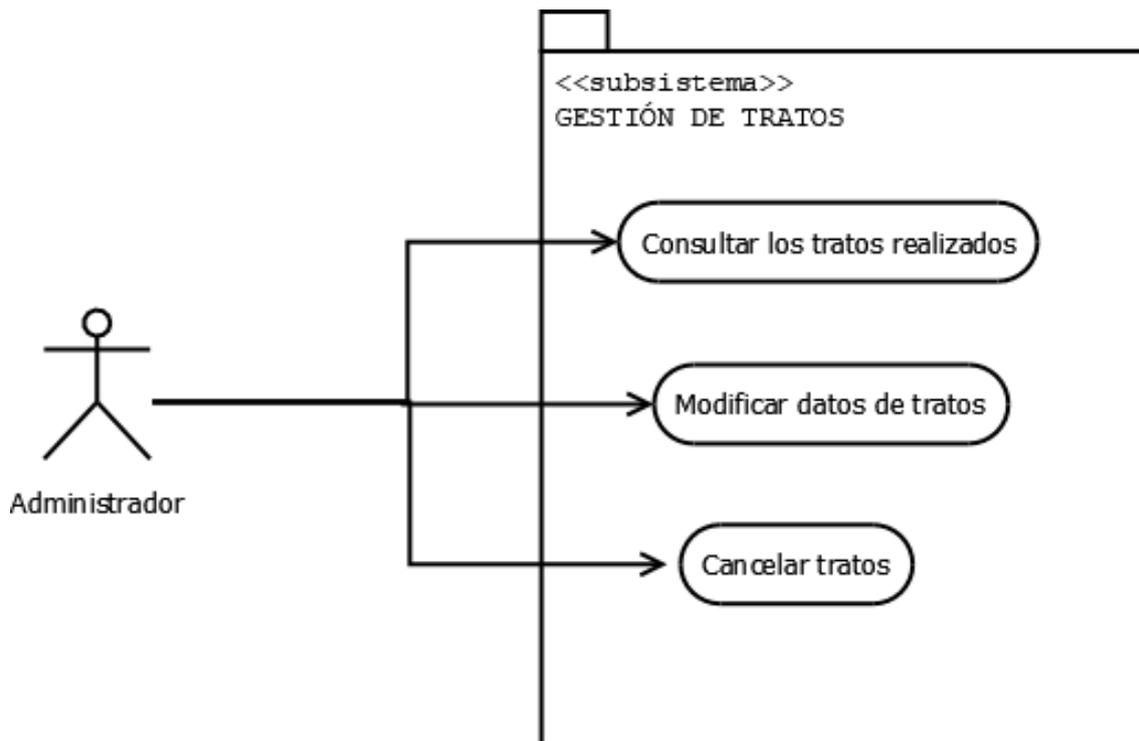


Ilustración 32. Casos de uso del administrador en el subsistema de gestión de tratos.

<b>C.U.2.1</b>	Navegar por la lista de categorías de la aplicación
<b>Actores</b>	Usuario (Cliente)
<b>Descripción</b>	El usuario cliente decide buscar un profesional relacionado con la avería que tiene que arreglar por lo que se adentra en la lista de categorías que brinda la aplicación.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	PreC1: Que la aplicación funcione. PreC2: Que el usuario tenga una cuenta en la aplicación.
<b>Secuencia Normal</b>	P1: El usuario registrado inicia sesión en la aplicación. P2: Entra en la lista de categorías en la sección izquierda superior de la pantalla principal. P3: Navega por la lista de categorías, leyendo su descripción y subcategorías para encontrar la adecuada.
<b>Postcondición</b>	PostC1: El usuario ha encontrado la categoría relacionada con la avería a arreglar.
<b>Excepciones</b>	No hay excepciones.

Tabla 14. Caso de uso 2.1.

<b>C.U.2.2</b>	Encontrar profesionales relacionados con la categoría a elegir.
<b>Actores</b>	Usuario (Cliente).
<b>Descripción</b>	El usuario cliente decide buscar un profesional relacionado con la avería que tiene que arreglar por lo que se adentra en la lista de categorías que brinda la aplicación. Una vez encontrada la categoría, procede a elegir un profesional adecuado para arreglar la avería.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	PreC1: Que la aplicación funcione. PreC2: Que el usuario tenga una cuenta en la aplicación. PreC3: Que el usuario haya encontrado una categoría indicada para su avería. PreC4: Que haya profesionales relacionados con la categoría elegida.
<b>Secuencia Normal</b>	P1: El usuario registrado inicia sesión en la aplicación. P2: Entra en la lista de categorías en la sección izquierda superior de la pantalla principal. P3: Navega por la lista de categorías, leyendo su descripción y subcategorías para encontrar la adecuada. P4: Encontrada la categoría adecuada, el usuario pulsa en ella para ver la lista de profesionales relacionados con la categoría elegida.
<b>Postcondición</b>	PostC1: El usuario ha encontrado un profesional adecuado para arreglar la avería.
<b>Excepciones</b>	No hay excepciones.

*Tabla 15. Caso de uso 2.2.*

<b>C.U.2.3</b>	Consultar el perfil del profesional elegido.
<b>Actores</b>	Usuario (cliente)
<b>Descripción</b>	El usuario al haber encontrado ya un profesional adecuado para encargarse del trabajo decide consultar su perfil para comprobar que es un profesional bien valorado por otros usuarios.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	PreC1: Que la aplicación funcione. PreC2: Que el usuario tenga una cuenta en la aplicación.

	<p>PreC3: Que el usuario haya encontrado una categoría indicada para su avería.</p> <p>PreC4: Que haya profesionales relacionados con la categoría elegida.</p>
<p><b>Secuencia Normal</b></p>	<p>P1: El usuario registrado inicia sesión en la aplicación.</p> <p>P2: Entra en la lista de categorías en la sección izquierda superior de la pantalla principal.</p> <p>P3: Navega por la lista de categorías, leyendo su descripción y subcategorías para encontrar la adecuada.</p> <p>P4: Encontrada la categoría adecuada, el usuario pulsa en ella para ver la lista de profesionales relacionados con la categoría elegida.</p> <p>P5: Tras haber encontrado un profesional adecuado, pulsa en su card.</p> <p>P6: Consulta sus datos personales y las valoraciones recibidas por otros usuarios.</p>
<p><b>Postcondición</b></p>	<p>No hay postcondiciones.</p>
<p><b>Excepciones</b></p>	<p>No hay excepciones.</p>

Tabla 16. Caso de uso 2.3.

<p><b>C.U.2.4</b></p>	<p>Abrir una sala de chat con el profesional.</p>
<p><b>Actores</b></p>	<p>Usuario (cliente).</p>
<p><b>Descripción</b></p>	<p>El usuario, tras haber encontrado al profesional adecuado, decide abrir una sala de chat para, posteriormente, hablar con él.</p>
<p><b>Dependencias</b></p>	<p>No tiene ninguna dependencia.</p>
<p><b>Precondición</b></p>	<p>PreC1: Que la aplicación funcione.</p> <p>PreC2: Que el usuario tenga una cuenta en la aplicación.</p> <p>PreC3: Que el usuario haya encontrado una categoría indicada para su avería.</p> <p>PreC4: Que haya profesionales relacionados con la categoría elegida.</p> <p>PreC5: Haber encontrado al profesional adecuado.</p>
<p><b>Secuencia Normal</b></p>	<p>P1: El usuario registrado inicia sesión en la aplicación.</p> <p>P2: Entra en la lista de categorías en la sección izquierda superior de la pantalla principal.</p>

	<p>P3: Navega por la lista de categorías, leyendo su descripción y subcategorías para encontrar la adecuada.</p> <p>P4: Encontrada la categoría adecuada, el usuario pulsa en ella para ver la lista de profesionales relacionados con la categoría elegida.</p> <p>P5: Tras haber encontrado un profesional adecuado, pulsa en su card.</p> <p>P6: Consulta sus datos personales y las valoraciones recibidas por otros usuarios.</p> <p>P7: Convencido de ser el profesional adecuado, pulsa el botón “Empezar un chat con él”.</p>
<b>Postcondición</b>	PostC1: El usuario ha creado una sala de chat con el profesional.
<b>Excepciones</b>	No hay excepciones.

Tabla 17. Caso de uso 2.4.

<b>C.U.2.5</b>	Hablar con el profesional en el chat.
<b>Actores</b>	Usuario (Cliente)
<b>Descripción</b>	Tras haber creado la sala del chat con el profesional, el usuario decide hablar con él para llegar a un acuerdo.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	<p>PreC1: Que la aplicación funcione.</p> <p>PreC2: Que el usuario tenga una cuenta en la aplicación.</p> <p>PreC3: Que el usuario haya encontrado una categoría indicada para su avería.</p> <p>PreC4: Que haya profesionales relacionados con la categoría elegida.</p> <p>PreC5: Haber encontrado al profesional adecuado.</p> <p>PreC6: Que la sala de chat esté creada.</p>
<b>Secuencia Normal</b>	<p>P1: El usuario registrado inicia sesión en la aplicación.</p> <p>P2: Entra en la lista de categorías en la sección izquierda superior de la pantalla principal.</p> <p>P3: Navega por la lista de categorías, leyendo su descripción y subcategorías para encontrar la adecuada.</p> <p>P4: Encontrada la categoría adecuada, el usuario pulsa en ella para ver la lista de profesionales relacionados con la categoría elegida.</p>

	<p>P5: Tras haber encontrado un profesional adecuado, pulsa en su card.</p> <p>P6: Consulta sus datos personales y las valoraciones recibidas por otros usuarios.</p> <p>P7: Convencido de ser el profesional adecuado, pulsa el botón “Empezar un chat con él”.</p> <p>P8: La aplicación le redirige al chat.</p> <p>P9: El usuario hace uso del TextInput del chat para mandarle mensajes al profesional.</p>
<b>Postcondición</b>	No hay postcondiciones.
<b>Excepciones</b>	No hay excepciones.

Tabla 18. Caso de uso 2.5.

<b>C.U.2.6</b>	Cerrar trato con un profesional.
<b>Actores</b>	Usuario (Cliente)
<b>Descripción</b>	Tras haber charlado lo suficiente con el profesional, el usuario decide cerrar un trato con él.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	<p>PreC1: Que la aplicación funcione.</p> <p>PreC2: Que el usuario tenga una cuenta en la aplicación.</p> <p>PreC3: Que el usuario haya encontrado una categoría indicada para su avería.</p> <p>PreC4: Que haya profesionales relacionados con la categoría elegida.</p> <p>PreC5: Haber encontrado al profesional adecuado.</p> <p>PreC6: Que la sala de chat esté creada.</p>
<b>Secuencia Normal</b>	<p>P1: El usuario registrado inicia sesión en la aplicación.</p> <p>P2: Entra en la lista de categorías en la sección izquierda superior de la pantalla principal.</p> <p>P3: Navega por la lista de categorías, leyendo su descripción y subcategorías para encontrar la adecuada.</p> <p>P4: Encontrada la categoría adecuada, el usuario pulsa en ella para ver la lista de profesionales relacionados con la categoría elegida.</p>

	<p>P5: Tras haber encontrado un profesional adecuado, pulsa en su card.</p> <p>P6: Consulta sus datos personales y las valoraciones recibidas por otros usuarios.</p> <p>P7: Convencido de ser el profesional adecuado, pulsa el botón “Empezar un chat con él”.</p> <p>P8: La aplicación le redirige al chat.</p> <p>P9: El usuario hace uso del TextInput del chat para mandarle mensajes al profesional.</p> <p>P10: Como le ha gustado el profesional, el usuario pulsa en el botón que se encuentra en la parte inferior derecha y crea el trato entre él y el profesional.</p> <p>P11: La aplicación le redirige a la pantalla principal con un aviso mencionando que el trato se ha creado con éxito.</p>
<b>Postcondición</b>	PostC1: Se ha creado un trato entre cliente y profesional.
<b>Excepciones</b>	No hay excepciones.

Tabla 19. Caso de uso 2.6.

<b>C.U.2.7</b>	Consultar datos de tratos
<b>Actores</b>	Administrador.
<b>Descripción</b>	El administrador, por decisión propia, decide consultar los datos de los tratos existentes en la base de datos.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	No hay precondiciones.
<b>Secuencia Normal</b>	<p>P1: El administrador entra en el panel de control de Firebase.</p> <p>P2: Entra en el apartado Cloud Firestore.</p> <p>P3: Entra en la colección “tratos”.</p> <p>P4: Navega por todos los documentos relacionados con la colección para comprobar que todo está correcto.</p>
<b>Postcondición</b>	No hay postcondiciones
<b>Excepciones</b>	No hay excepciones.

Tabla 20. Caso de uso 2.7.

<b>C.U.2.8</b>	Modificar datos de tratos
<b>Actores</b>	Administrador.
<b>Descripción</b>	El administrador, por petición de otros clientes o por decisión propia, modifica algún dato que no está correcto.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	No hay precondiciones.
<b>Secuencia Normal</b>	P1: El administrador entra en el panel de control de Firebase. P2: Entra en el apartado Cloud Firestore. P3: Entra en la colección “tratos”. P4: Selecciona el documento del trato a modificar. P5: Modifica los datos que cree erróneos. P5: Confirma la modificación del documento.
<b>Postcondición</b>	PostC1: El trato ha sido modificado.
<b>Excepciones</b>	No hay excepciones.

Tabla 21. Caso de uso 2.8.

<b>C.U.2.9</b>	Cancelar tratos.
<b>Actores</b>	Administrador.
<b>Descripción</b>	El administrador, por petición de otros clientes o por decisión propia, decide cancelar un trato en específico.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	No hay precondiciones.
<b>Secuencia Normal</b>	P1: El administrador entra en el panel de control de Firebase. P2: Entra en el apartado Cloud Firestore. P3: Entra en la colección “tratos”. P4: Selecciona el documento del trato a cancelar. P5: Confirma el borrado del documento.
<b>Postcondición</b>	PostC1: El trato ha sido cancelado por el administrador.
<b>Excepciones</b>	No hay excepciones.

Tabla 22. Caso de uso 2.9.

Solo nos encontramos con un actor haciendo las funciones importantes del subsistema de gestión de tratos, siendo este el usuario cliente ya que es el único que puede crear tratos con un profesional, un usuario no registrado no puede navegar por la pantalla principal y un profesional solo puede saber que ha cerrado un trato con el cliente.

Por último, llegan las ilustraciones y los diagramas de los casos de uso relacionados con el ultimo subsistema, el de gestión de averías.

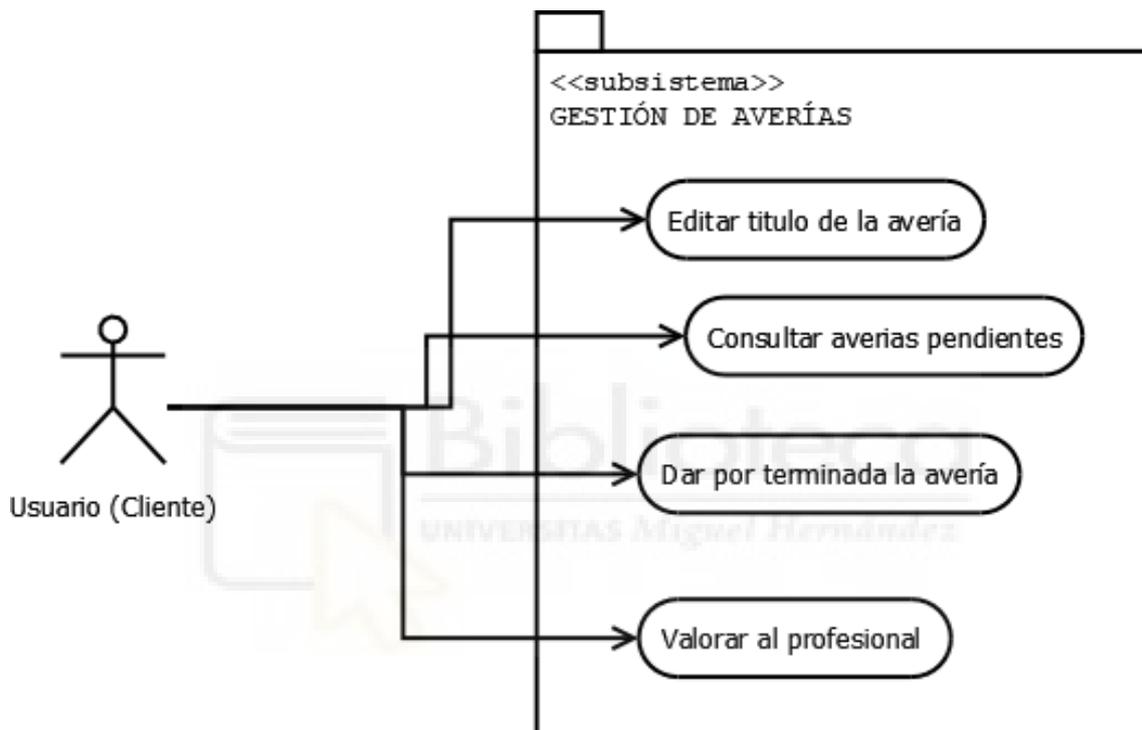


Ilustración 33. Casos de uso del usuario cliente en el subsistema de gestión de averías.

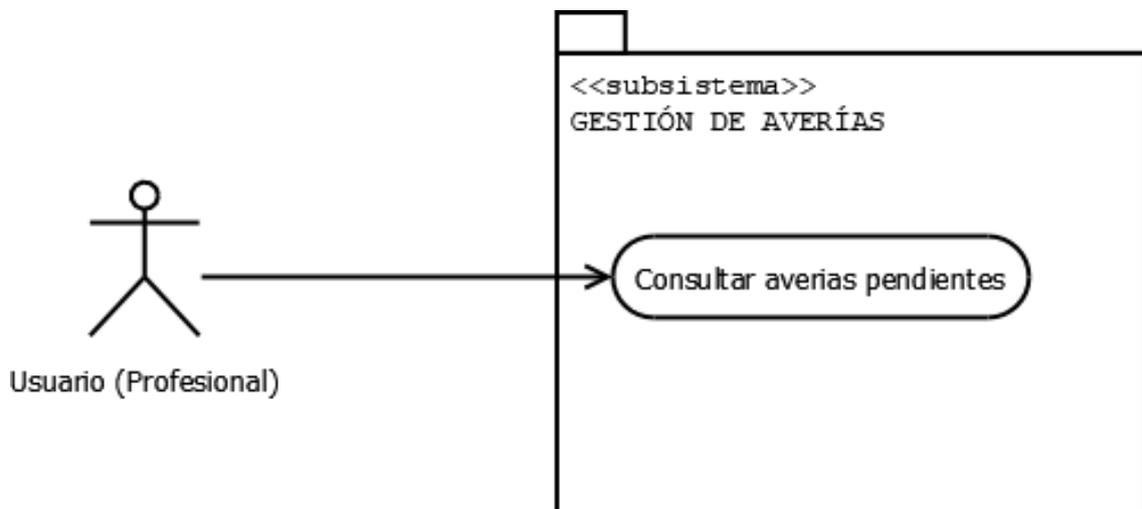


Ilustración 34. Casos de uso del usuario profesional en el subsistema de gestión de averías.

<b>C.U.3.1</b>	Editar título de la avería.
<b>Actores</b>	Usuario (Cliente)
<b>Descripción</b>	Los tratos entre cliente y profesional vienen con un título por defecto por lo que decide editar el título para darle uno más adecuado.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	PreC1: Que la aplicación funcione. PreC2: Que el usuario tenga una cuenta en la aplicación. PreC3: Que tenga un trato activo.
<b>Secuencia Normal</b>	P1: El usuario registrado inicia sesión en la aplicación. P2: Entra en su perfil. P3: Presiona levemente el trato a modificar el título. P4: Rellena el formulario donde escribirá el nuevo título del trato. P5: Confirma la modificación del título.
<b>Postcondición</b>	PostC1: El trato tiene el título modificado por el usuario.
<b>Excepciones</b>	No hay excepciones.

Tabla 23. Caso de uso 3.1.

<b>C.U.3.2</b>	Consultar averías pendientes.
<b>Actores</b>	Usuario (Cliente)
<b>Descripción</b>	El usuario navega por su perfil y comprueba la lista de tratos pendientes que tiene con profesionales.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	PreC1: Que la aplicación funcione. PreC2: Que el usuario tenga una cuenta en la aplicación. PreC3: Que tenga uno o más tratos activos.
<b>Secuencia Normal</b>	P1: El usuario registrado inicia sesión en la aplicación. P2: Entra en su perfil. P3: Lee la lista de tratos pendientes con los profesionales.
<b>Postcondición</b>	No hay postcondiciones
<b>Excepciones</b>	No hay excepciones.

Tabla 24. Caso de uso 3.2.

<b>C.U.3.3</b>	Dar por terminada la avería
<b>Actores</b>	Usuario (Cliente)
<b>Descripción</b>	El usuario, tras comprobar que el profesional ha realizado el trabajo, registra en la aplicación que la avería se ha completado.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	PreC1: Que la aplicación funcione. PreC2: Que el usuario tenga una cuenta en la aplicación. PreC3: Que tenga tratos activos. PreC4: Que el profesional haya completado el trabajo.
<b>Secuencia Normal</b>	P1: El usuario registrado inicia sesión en la aplicación. P2: Entra en su perfil. P3: Presiona por unos segundos el trato a terminar P4: La aplicación le pregunta al usuario si da por terminada esta avería. P5: El usuario confirma la pregunta.
<b>Postcondición</b>	PostC1: El trato entre cliente y profesional se ha terminado.
<b>Excepciones</b>	No hay excepciones.

Tabla 25. Caso de uso 3.3.

<b>C.U.3.4</b>	Valorar al profesional.
<b>Actores</b>	Usuario (Cliente)
<b>Descripción</b>	Una vez terminada la avería, el usuario procede a valorar el trabajo del profesional.
<b>Dependencias</b>	No tiene ninguna dependencia.
<b>Precondición</b>	PreC1: Que la aplicación funcione. PreC2: Que el usuario tenga una cuenta en la aplicación. PreC3: Que se haya completado el trato entre cliente y profesional.
<b>Secuencia Normal</b>	P1: Tras dar por terminado el trato entre profesional y usuario, la aplicación redirige al usuario a un formulario. P2: El usuario completa el formulario, diciendo su opinión y una valoración entre 1 y 5. P3: Registra la valoración para el profesional.
<b>Postcondición</b>	PostC1: El profesional ha recibido una valoración del usuario.
<b>Excepciones</b>	No hay excepciones.

Tabla 26. Caso de uso 3.4.

Tras ilustrar y explicar los subsistemas y todos los casos de uso posibles en la aplicación, continuamos con las ilustraciones de otros tipos de diagramas como, por ejemplo, diagramas de actividad o de carril.

#### 4.2.1.3. Otros diagramas

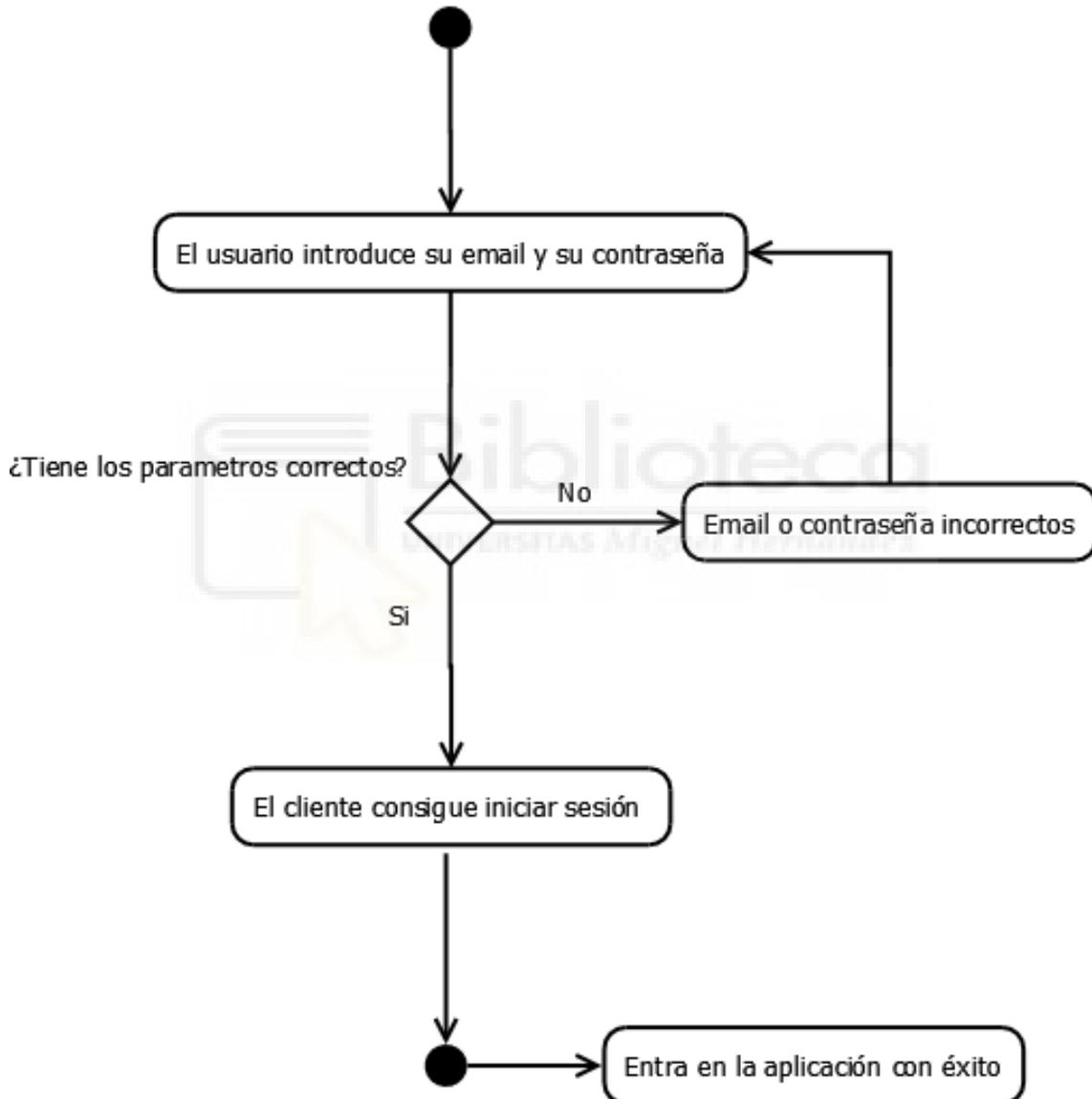


Ilustración 35. Diagrama de actividad relacionado con iniciar sesión.

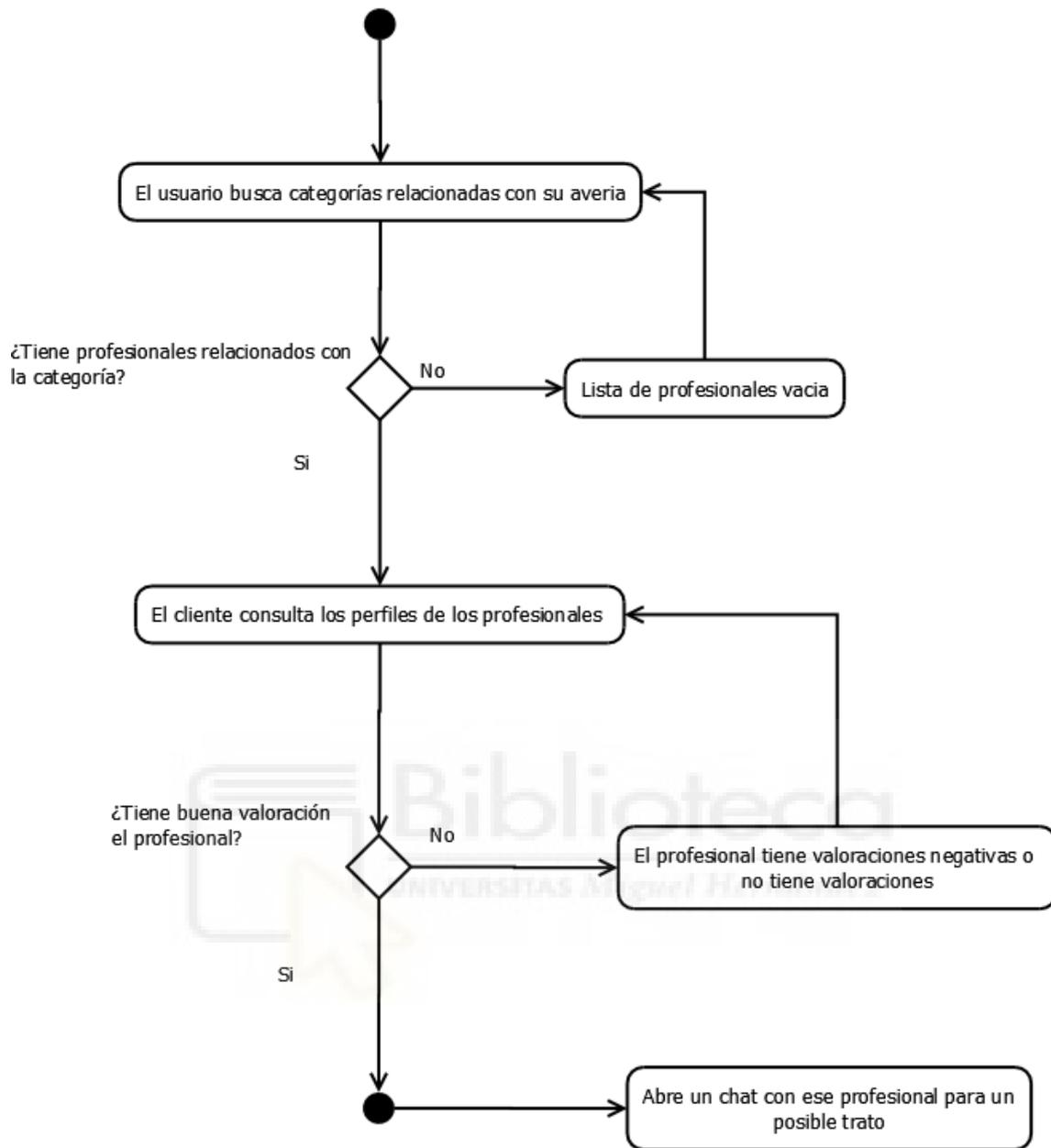


Ilustración 36. Diagrama de actividad relacionado con la búsqueda de profesionales.

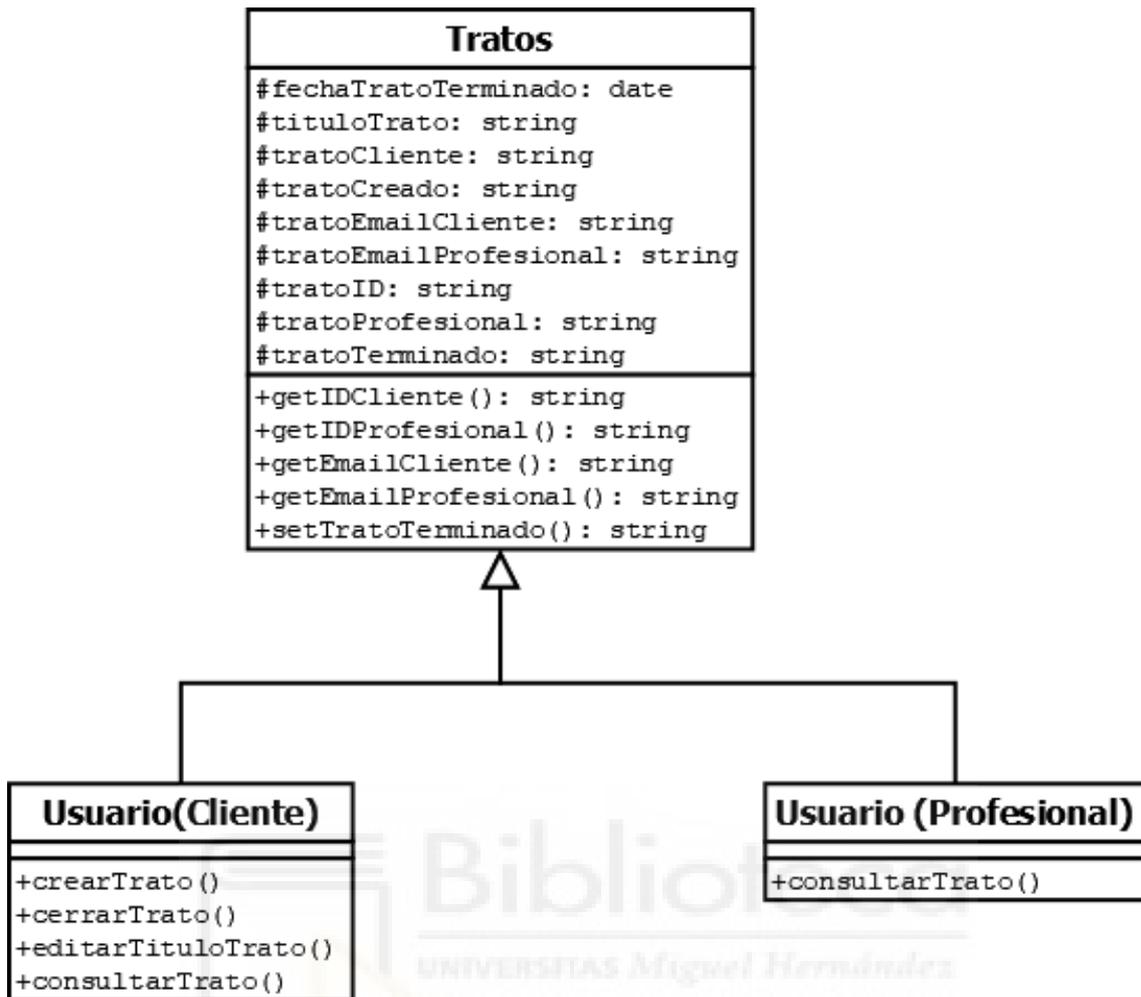


Ilustración 37. Diagrama de clases relacionado con los tratos.

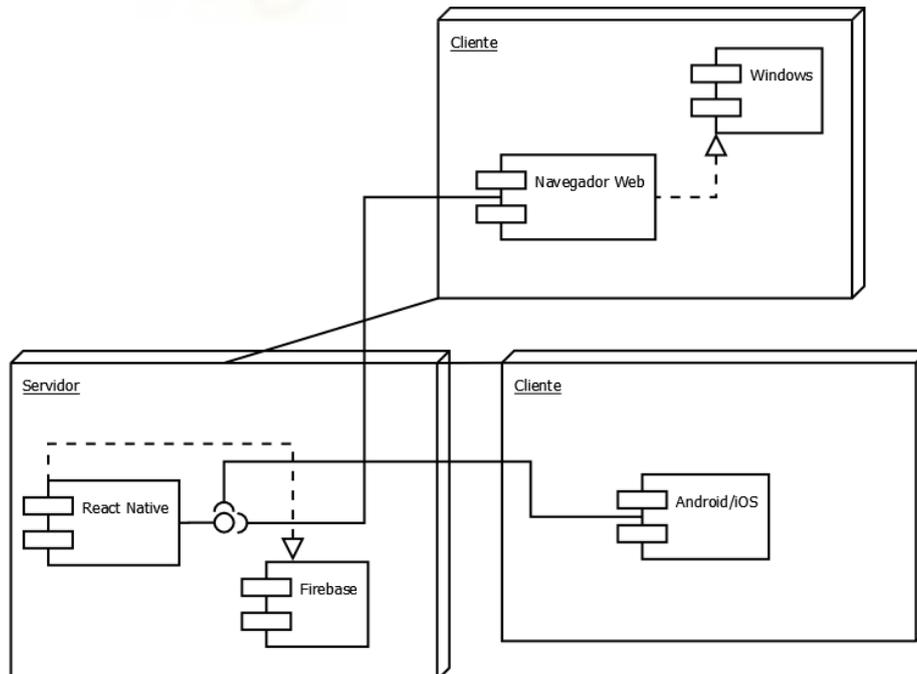


Ilustración 38. Diagrama de distribución/despliegue.

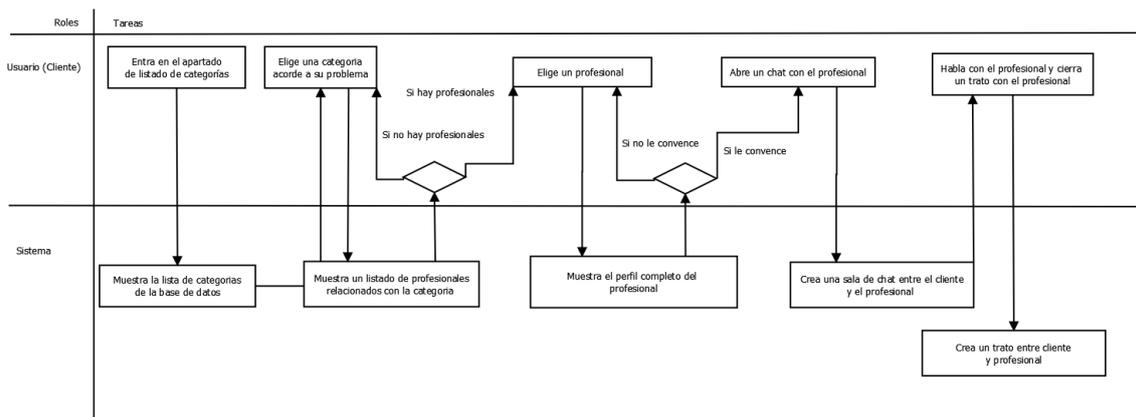


Ilustración 39. Diagrama de carril.

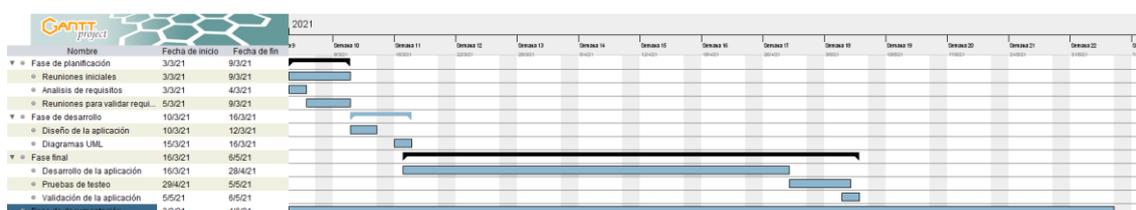


Ilustración 40. Diagrama Gantt.

### 4.3. Desarrollo de la herramienta. Implementación

#### 4.3.1. Descripción de la base de datos

Como se ha comentado previamente en varias ocasiones a lo largo de esta documentación la base de datos de la aplicación es de tipo NoSQL, por lo que no hay tablas, claves primarias o secundarias, relaciones entre tablas...etc. La base de datos se compone de estas colecciones.

- Categorías
- Users
- Valoraciones
- Tratos
- Salas

Cada colección está compuesta por una serie de documentos que actuarán de tablas (si estuviésemos hablando de una BDD de tipo SQL) y cada documento está compuesto por una serie de atributos.

Para empezar a hablar de las colecciones la más idónea sería la de users. Los documentos de esta colección tienen esta serie de atributos: Contraseña, domicilio, email, id, nombreCompleto, numTelefono y tipoUsuario. El atributo más interesante de esta colección es tipoUsuario porque dependiendo de lo que elija el usuario a la hora de registrarse, puede poseer más atributos en su documento o no. En el caso de que elija ser un profesional, se le añaden al documento estos atributos: averiasCompletadas, categoriasProfesional, valoración y valoracionAcumulada. Los atributos más complejos añadidos al profesional son categoriasProfesional y valoracionAcumulada, categoriasProfesional es un array de strings que recogerá todas las categorías escogidas por el profesional a la hora de registrarse. Por otro lado, tenemos valoracionAcumulada, que recogerá la suma de todas las valoraciones recibidas por el profesional para el cómputo de las insignias.

La colección categorías es una colección donde se almacena toda la información relacionada con las categorías que existen en la aplicación, los documentos de esta colección están compuestos por un id numérico y por el nombre de la categoría.

Otra de las colecciones más importantes de la BDD es la colección de las salas. Esta colección contendrá toda la información relacionada con el creador de la sala, con que profesional está hablando...etc. Esta colección tiene los siguientes atributos: creadorSala, emailCliente, emailProfesionalEncargado, idProfesional, lastestMessage, salaID.

Una cualidad que tiene la colección Salas es que contiene una subcolección llamada Mensajes, los documentos de esta subcolección tendrán los atributos: createdAt, text y user. Básicamente la función que cumple esta subcolección es de contener todo el registro de mensajes entre el usuario cliente y el usuario profesional, con los atributos text y user se puede saber quién ha escrito cada mensaje y a qué hora los ha escrito (Con createdAt).

Para seguir la explicación de las colecciones, es adecuado explicar ahora la colección de tratos ya que se crean cuando un usuario cliente está hablando con un profesional a través de una sala de chat. Esta colección (y la explicada anteriormente) podría compararse con una relación 1:1 si estuviésemos hablando de una BDD de tipo SQL debido a que estamos recogiendo en esta colección tanto el id del cliente como el id del profesional (como si fuesen claves primarias). Aparte de recoger los ids de ambos usuarios, los documentos de

esta colección contienen: fechaTratoTerminado, tituloTrato, tratoCliente, tratoCreado, tratoEmailCliente, tratoEmailProfesional, tratoID, tratoProfesional, tratoTerminado.

Algunos atributos de la colección Tratos dignos de explicar son como, por ejemplo, fechaTratoTerminado, que, a la hora de crearse el trato obtiene el mismo valor que tratoCreado, pero cuando el usuario finaliza el trato, se actualiza a la fecha que el usuario haya completado la acción. Otro atributo interesante es tituloTrato, el cual pilla un valor por defecto cuando se crea el trato, pero se actualiza cuando el usuario edita el título del trato desde su perfil. Por último, tenemos tratoTerminado que tiene por defecto el valor “no” cuando se crea y se actualiza a “sí” cuando el usuario finaliza el trato.

Para terminar este apartado, explicaré la última colección a mencionar, la colección Valoraciones. Esta colección contendrá toda la información relacionada con las opiniones que dan los usuarios, a quien le dan esa valoración...etc. Los atributos de los documentos de la colección Valoraciones son: cliente, emailCliente, emailProfesional, fechaValoracion, opinionCliente, profesional y valorValoracion.

En el formulario que rellenará el usuario a la hora de darle una valoración al profesional escribirá una opinión breve sobre él y lo valorará sobre 5, dichos valores se guardarán en los atributos opinionCliente y valorValoracion respectivamente.

En definitiva, como se puede comprobar, una BDD de tipo NoSQL no se diferencia mucho de una de tipo SQL ya que se puede simular algunas características de la BDD de tipo SQL como por ejemplo las relaciones 1:1 (Teniendo una colección donde los documentos contengan los ids de los documentos a relacionar), las claves primarias (Siendo estas los ids generados automáticamente por Cloud Firestone)...etc.

### **4.3.2. Implementación**

Este apartado va a ir dedicado a comentar algunos elementos interesantes que se han implementado en la aplicación, ya sea para añadirle funcionalidad, adaptarlo a varias situaciones,etc.

El principal elemento interesante que hay que comentar es el chat implementado en la aplicación. El chat es básicamente el componente React Native Gifted Chat. Quise

implementar un chat de manera síncrona ya que la función principal de la aplicación es que un usuario sea capaz de ponerse en contacto con un profesional de manera inmediata, sin tener que enviar correos y esperar 2-3 días a que le conteste. Con un chat síncrono da la sensación de que hay más acercamiento y rapidez entre el usuario cliente y el usuario profesional.

Otro elemento digno de mencionar (aunque se haya explicado anteriormente) es la implementación de una BDD de tipo NoSQL. Cuando estuve desarrollando la aplicación en React Native estuve informándome de que base de datos era la más idónea de implementar, una de las opciones era Firebase y como en la carrera la suelen mencionar bastante, decidí implementarla. Firebase a su vez tiene varias herramientas de trabajo de base de datos, siendo una de ellas Cloud Firestore. Investigué sobre ella y como trabaja sobre una BDD de tipo NoSQL la elegí por motivos como innovación, facilidad a la hora de trabajar con ella y autoaprendizaje.



## 5. CONCLUSIONES Y PROPUESTAS

### 5.1. Conclusiones sobre el proyecto

Quiero empezar este apartado diciendo que este proyecto me ha servido como reto para seguir siendo autodidacta ya que he trabajado durante estos meses con tecnologías que no conocía en profundidad como son React Native (y todas sus librerías) o las bases de datos de tipo NoSQL. Investigando día a día toda la documentación posible sobre dichos campos ha hecho que vaya adaptándome a la manera de trabajar con frameworks y librerías.

También es cierto que gracias a varias asignaturas que se imparten en la universidad he podido seguir desarrollando este proyecto, ya sea por la parte del desarrollo de la aplicación (Con asignaturas como Desarrollo de Aplicaciones Móviles, Desarrollo de Aplicaciones para Internet, Diseño de Aplicaciones Multimedia...etc.) como por la parte del desarrollo de la documentación (Con asignaturas como Diseño de Sistemas Interactivos, Ingeniería del Software, Gestión de Proyectos de Ingeniería del Software,etc.)

### 5.2. Propuestas de futuro

En el supuesto caso de que se siga implementando mas funcionalidades a la aplicación se podría implementar que el usuario sea capaz de poder enviarle imágenes, videos o incluso notas de audio al profesional a la hora de chatear con él para poder explicarle con más detalle el problema a solucionar.

Otra funcionalidad sería implementar un sistema de tracking, similar al sistema de tracking de Amazon, donde puedes visualizar por donde circula el conductor hasta llegar al destino. Sería interesante ver el camino que realiza el profesional para futuras conclusiones a la hora de valorarle, por si ha sido rápido o se ha tomado caminos innecesarios.

También sería interesante un espacio donde los profesionales puedan demostrar sus habilidades a la hora de reparar cosas, para atraer a futuros clientes. Este espacio podría traducirse en forma de blog para los clientes, que vayan indagando en busca del profesional indicado.

## 6. BIBLIOGRAFÍA

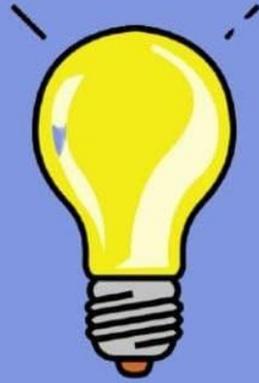
- [1]: Documento oficial extraído por la url: <https://reactnative.dev/>
- [2]: Documento oficial extraído por la url: <https://nodejs.org/es/about/>
- [3]: Documento oficial extraído por la url: <https://docs.npmjs.com/about-npm>
- [4]: Documento oficial extraído por la url: <https://expo.io/>
- [5]: Documento oficial extraído por la url: <https://firebase.google.com/docs>
- [6]: Documento oficial extraído por la url:  
<https://firebase.google.com/docs/firestore?hl=es>
- [7]: Documento oficial extraído por la url: <https://github.com/FaridSafi/react-native-gifted-chat>
- [8]: Documento oficial extraído por la url: <https://github.com/Monte9/react-native-ratings>
- [9]: Documento oficial extraído por la url: <https://github.com/react-native-picker/picker>
- [10]: Libro: Ingeniería del software. (2005). Autores: Ian Sommerville. Pearson Addison Wesley
- [11]: Libro: [Effective prototyping for software makers \(2007\). Jonathan Arnowitz, Michael Arent y Nevin Berger. ProQuest Ebook Central.](#)
- [12]: Libro: El lenguaje unificado de modelado: guía del usuario. (2006). G. Booch, J. Rubaugh y I. Jacobson. Pearson Addison Wesley

## 7. ANEXO

Aquí se mostrará un recorrido visual por toda la aplicación, observando todo lo que pueden hacer tanto el usuario “cliente” como el usuario “profesional. Lo primero que podrá encontrarse el usuario es con un elemento Onboarding que muestra lo esencial de la aplicación.



*Ilustración 41. Primera página del elemento Onboarding.*



## Sé tu propio jefe

Ahora con AveriasApp, podrás anunciarte como profesional y estar a disposición de cualquier usuario.

Skip



Next

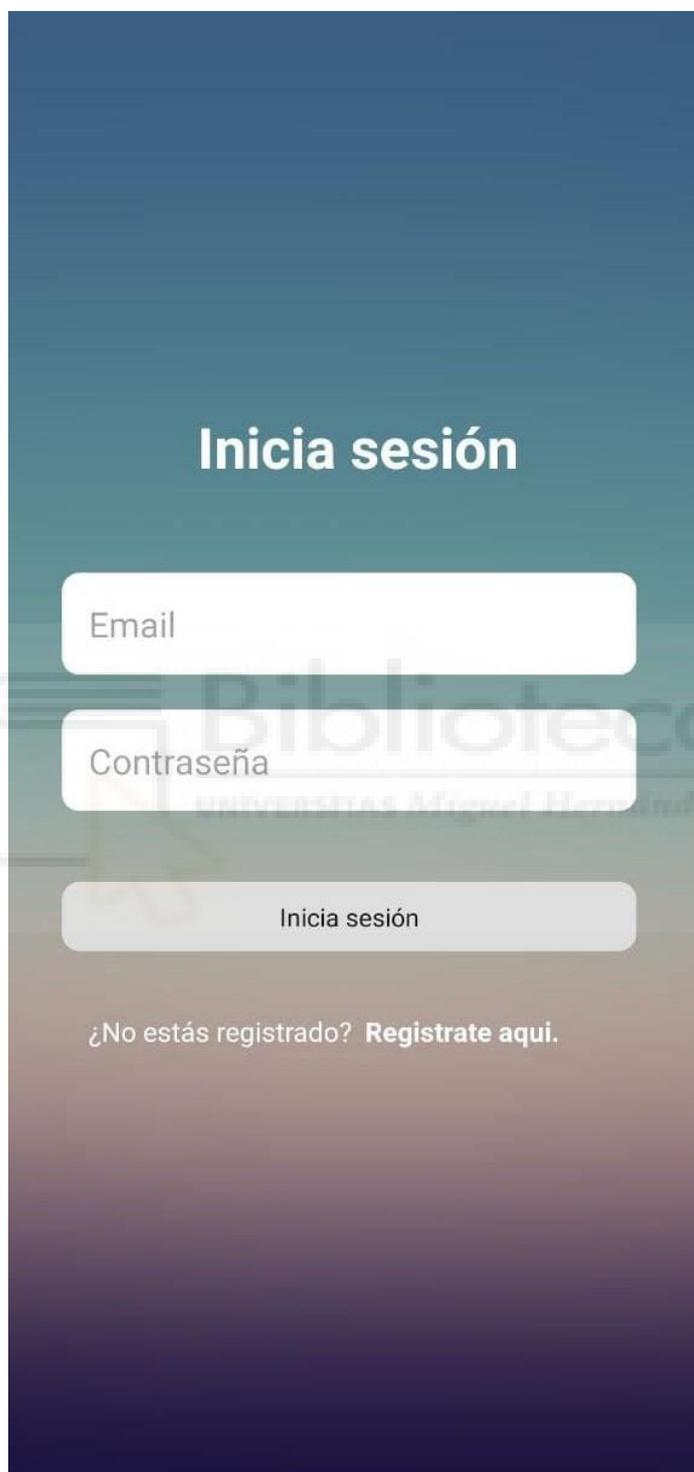
Ilustración 42. Segunda página del elemento Onboarding.



*Ilustración 43. Última página del elemento Onboarding.*

Tras ver las tres páginas del Onboarding, el usuario pasará al formulario para iniciar sesión, donde escribirá su email y su contraseña.

Si se da la ocasión de que el usuario no está registrado en la aplicación, hay un enlace que le llevará al formulario para registrarse.



The image shows a mobile application login screen. At the top, the title "Inicia sesión" is displayed in white text on a dark blue background. Below the title are two white input fields: "Email" and "Contraseña". A grey button labeled "Inicia sesión" is positioned below the password field. At the bottom of the form, there is a link that reads "¿No estás registrado? **Regístrate aquí.**". A faint watermark "Biblioteca UNIVERSIDAD Miguel Hernández" is visible in the background of the form area.

*Ilustración 44. Formulario de inicio de sesión.*

# Bienvenido/a a AveriasApp!

Nombre completo

Carmelo Peñalver Sánchez

Dirección de correo

carmelotest@email.com

Contraseña

.....

Repita la contraseña

.....

Número de teléfono

Regístrate!

*Ilustración 45. Formulario de registro del usuario.*

# Bienvenido/a a AveriasApp!

.....

Repite la contraseña

.....

Número de teléfono

111222333|

Domicilio

Orihuela

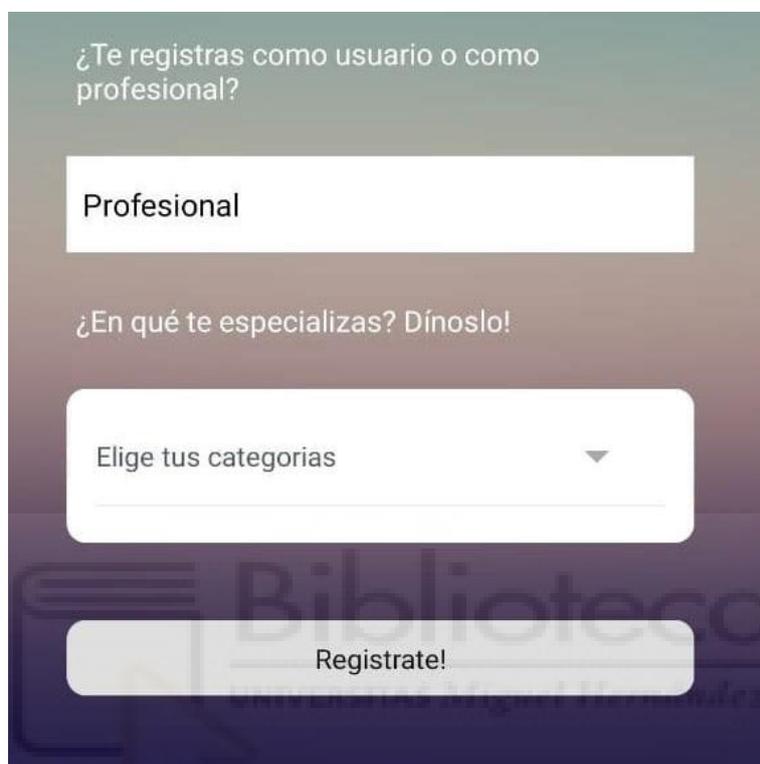
¿Te registras como usuario o como profesional?

Usuario

Regístrate!

Ilustración 46. Formulario de registro del usuario.

Tras haber completado el formulario de registro se le pregunta al usuario si es usuario “cliente” o usuario “profesional”, en el caso de que el usuario se registre como profesional, se le abre un nuevo menú desplegable donde podrá elegir con que categorías se especializa.



¿Te registras como usuario o como profesional?

Profesional

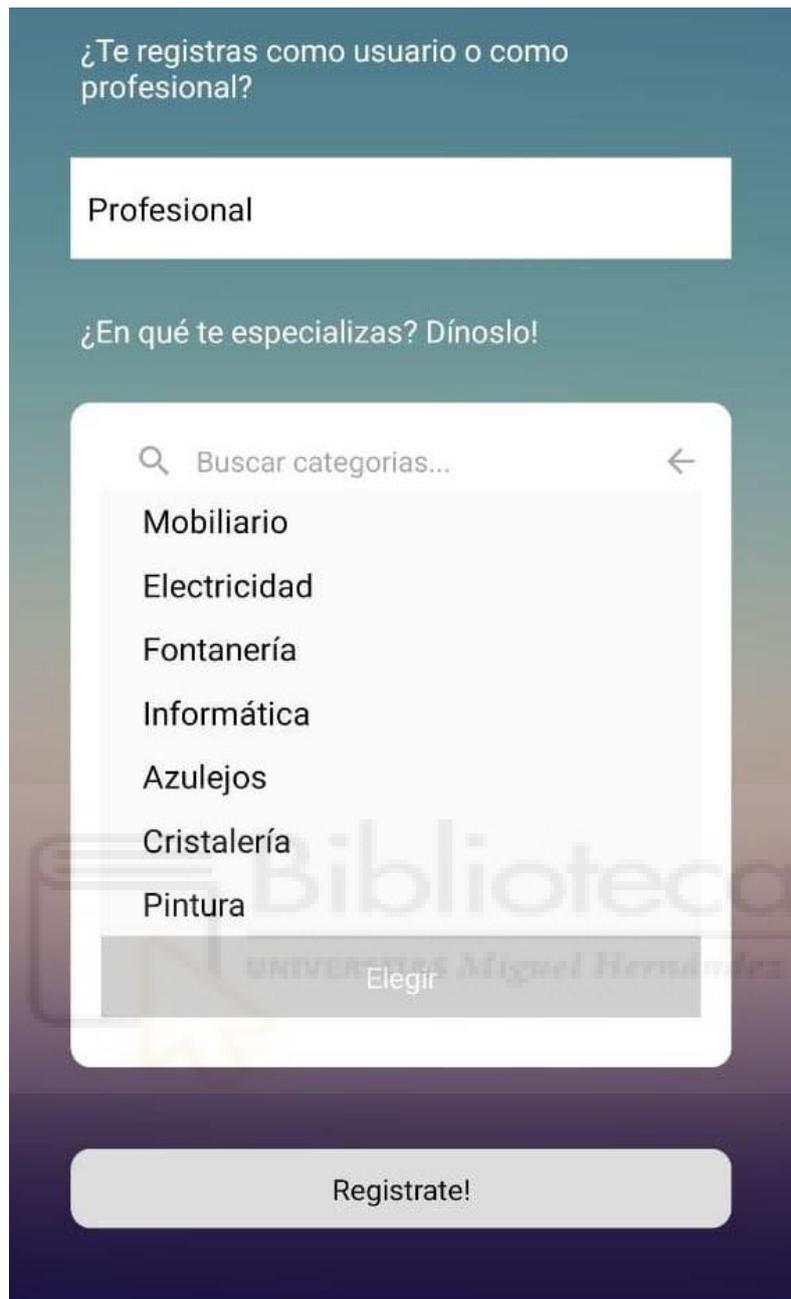
¿En qué te especializas? Dínoslo!

Elige tus categorías

Regístrate!

The image shows a registration form with a dark background. At the top, it asks '¿Te registras como usuario o como profesional?'. Below this is a white input field containing the text 'Profesional'. The next question is '¿En qué te especializas? Dínoslo!'. This is followed by a white dropdown menu with the text 'Elige tus categorías' and a downward arrow. At the bottom of the form is a white button with the text 'Regístrate!'. A faint watermark 'Biblioteca' is visible in the background.

*Ilustración 47. Menú de categorías.*



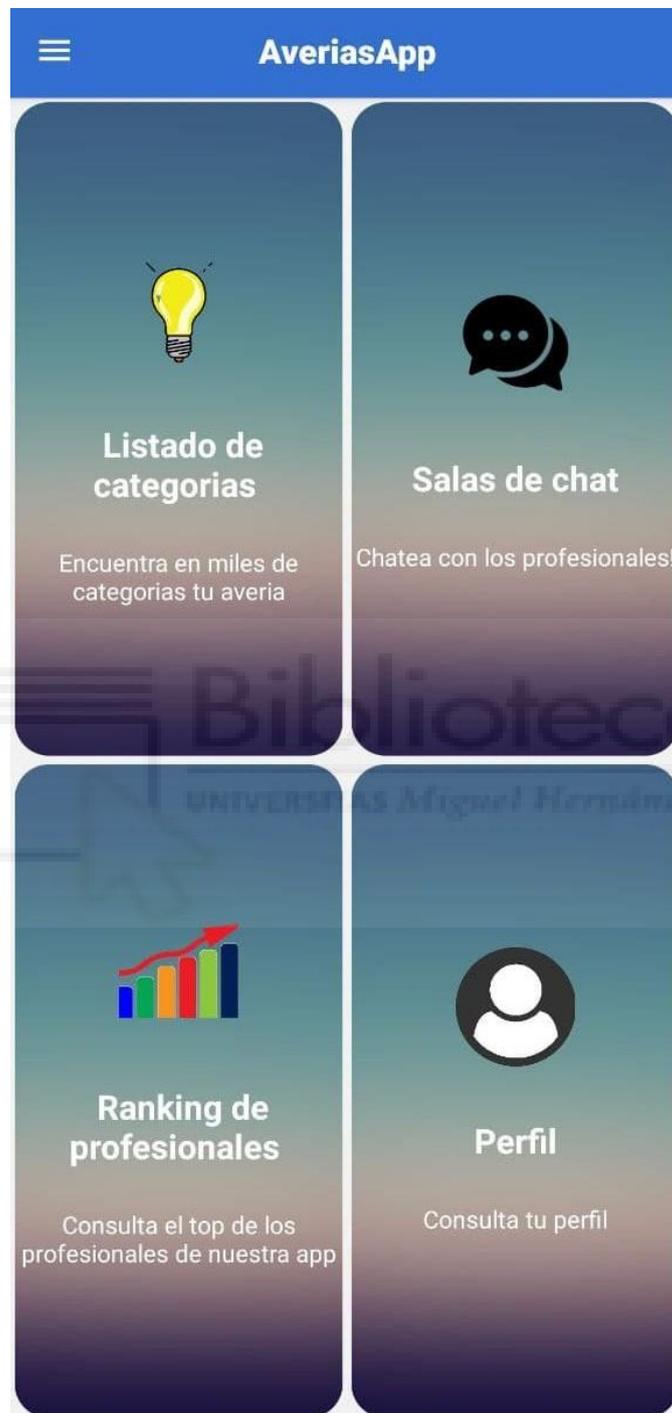
*Ilustración 48. Opciones disponibles del menú desplegable.*

### **7.1. Punto de vista del usuario “cliente”**

Tras haberse registrado el usuario se le lleva a la página principal de la aplicación donde podrá encontrar las 4 secciones más importantes de la aplicación, siendo estas:

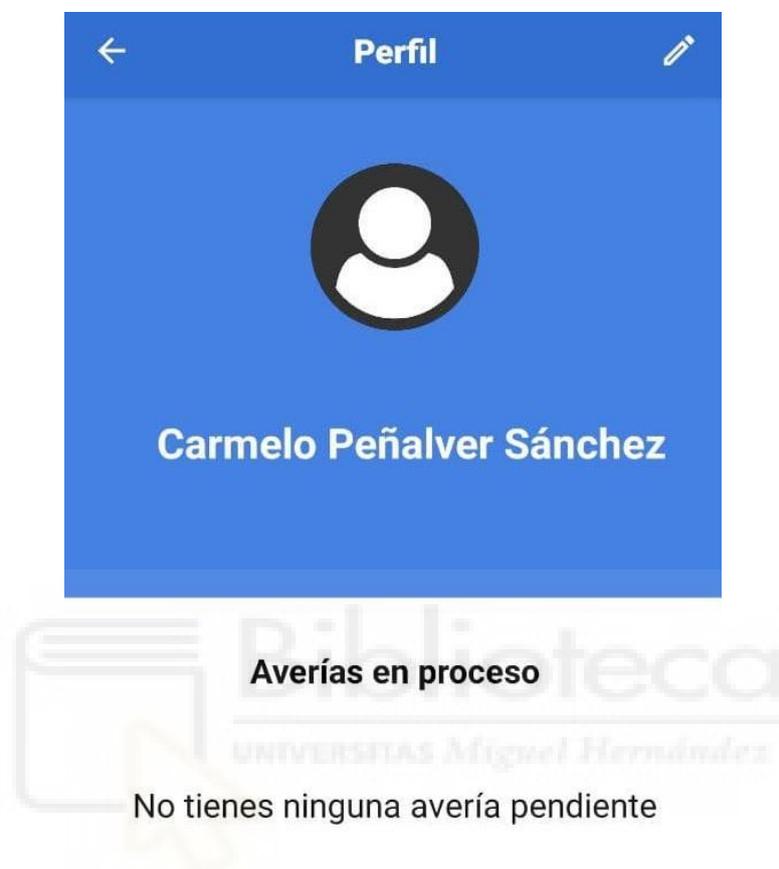
- Listado de categorías, donde podrá encontrar todas las categorías de la aplicación
- Salas de chat, donde estarán las salas de chat con los profesionales con los que haya hablado
- Ranking de profesionales, en esta sección se encuentra un listado de todos los profesionales suscritos a la aplicación, se les podrá filtrar según varios criterios

- Perfil, donde el usuario encontrará toda su información correspondiente a los datos personales y todos los tratos que haya hecho con cualquier profesional.



*Ilustración 49. Pantalla principal de la aplicación.*

Empezando por el perfil, lo que encontrará el usuario será su información personal y toda la actividad que haya hecho en la aplicación (averías pendientes, historial de averías completadas y valoraciones dadas a los profesionales).



*Ilustración 50. Pantalla principal de la sección Perfil.*



### **Averías en proceso**

No tienes ninguna avería pendiente



*Ilustración 51. Datos personales del usuario.*

Como se puede comprobar en ambas imágenes, nos encontramos con un botón con un diseño de un lápiz que si el usuario pulsa le llevará a un formulario donde podrá editar todos sus datos personales.

← **Editar perfil**

**Edita tus datos**

Carmelo Peñalver Sánchez

carmelotest@email.com

111222333

Orihuela

**¿Quiere cambiar la contraseña?**

Nueva contraseña

Repita la nueva contraseña

**Editar datos**

Biblioteca  
UNIVERSIDAD Miguel Hernández

*Ilustración 52. Formulario para editar los datos personales.*

Al editar cualquier dato y registrar el cambio la aplicación le redirigirá a su perfil donde se habrán realizado todos los cambios pertinentes.

Continuamos el recorrido por la sección de ranking de profesionales, donde el usuario encontrará una lista con todos los profesionales registrados en la aplicación con información relacionada con la valoración que tienen y el número de averías que hayan completado trabajando con la aplicación.

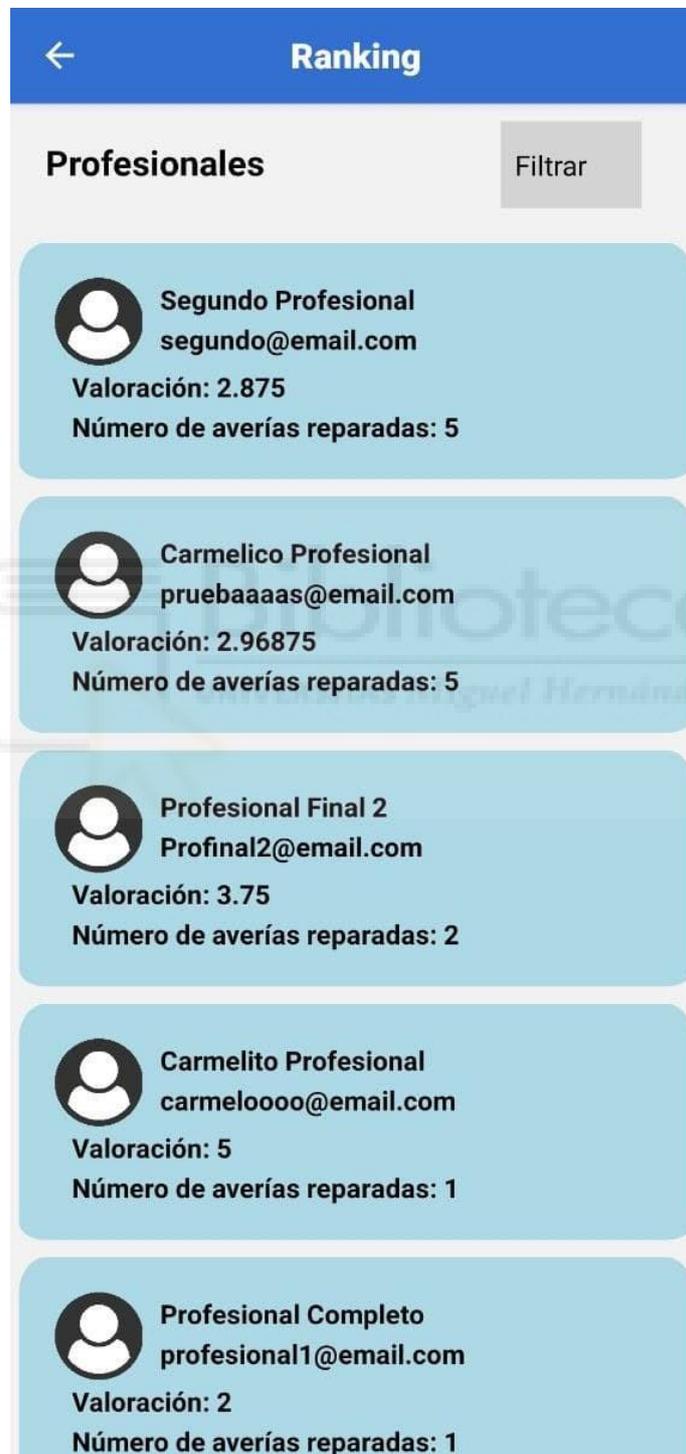


Ilustración 53. Listado de profesionales en el ranking.

Aparte de la lista de profesionales encontrará un menú dropdown donde el usuario podrá filtrar la lista de profesionales según la valoración más alta o el número de averías reparadas.

Tras analizar este breve sector de la aplicación es el momento de explicar la sección del listado de categorías que irá ligado a la sección de las salas de chats. En la zona de listado de categorías el usuario encontrará todas las categorías disponibles con una pequeña descripción de la categoría y algunas subcategorías que se engloban dentro de una categoría.



*Ilustración 54. Listado de categorías.*

Si un usuario pulsa en alguna categoría en concreto encontrará un listado de profesionales relacionados con esa categoría, puede pulsar en cualquier profesional para visualizar su perfil, datos personales e insignias, también podrá visualizar todas las valoraciones recibidas por los distintos usuarios de la aplicación.

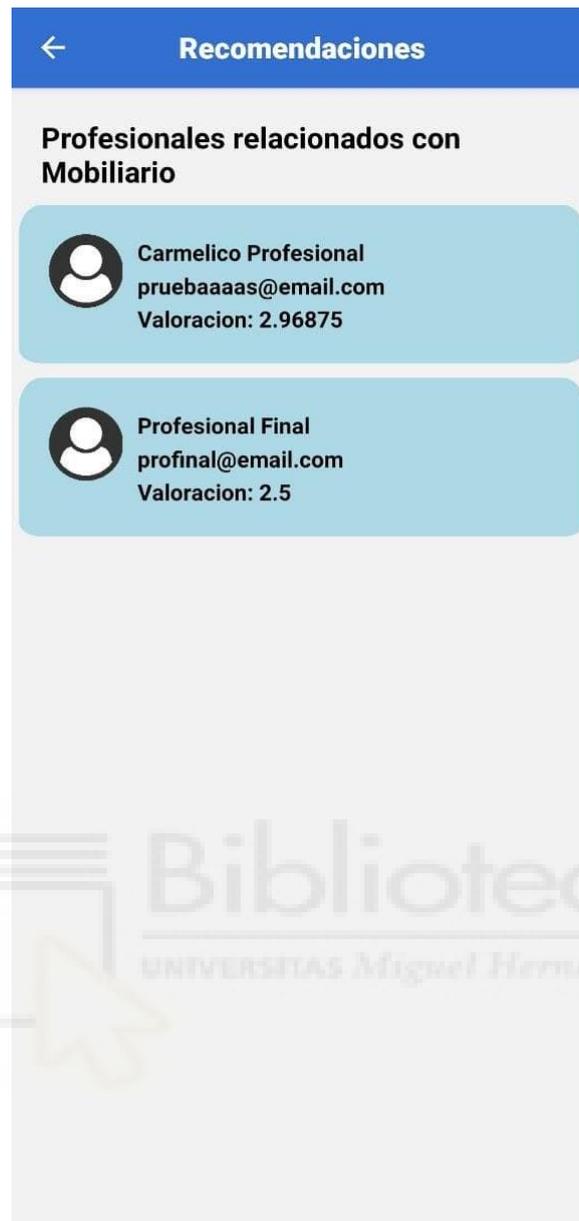


Ilustración 55. Listado de recomendaciones de profesionales relacionados con Mobiliario.



*Ilustración 56. Perfil del profesional Carmelico Profesional.*

Si el usuario “cliente” quiere chatear con un profesional en concreto se le brinda un botón en la parte de debajo del perfil del profesional, este botón le llevará a una sala de chat entre dicho usuario y el profesional que haya elegido.



*Ilustración 57. Chat entre el usuario cliente y el usuario profesional.*

Como se puede comprobar en la imagen, el usuario cliente tiene un botón cuyo dibujo se asemeja a un apretón de manos cuya función será la creación de tratos entre el cliente y el profesional, si el cliente presiona dicho botón salta una advertencia avisándole de si está seguro de si quiere crear un trato, en caso afirmativo, se le redirigirá a la página principal con una confirmación de la creación del trato, en caso negativo, volverá al chat con el profesional.



*Ilustración 58. Advertencia de la creación del trato entre cliente y profesional.*

Tras la creación del trato el cliente podrá verlo en su perfil, concretamente en la sección de averías pendientes donde podrá ver cuando se creó el trato, el título de la avería y que profesional es el encargado de arreglar dicha avería.

Si el usuario presiona sutilmente alguna avería saltará una advertencia diciéndole si quiere editar el título de la avería.

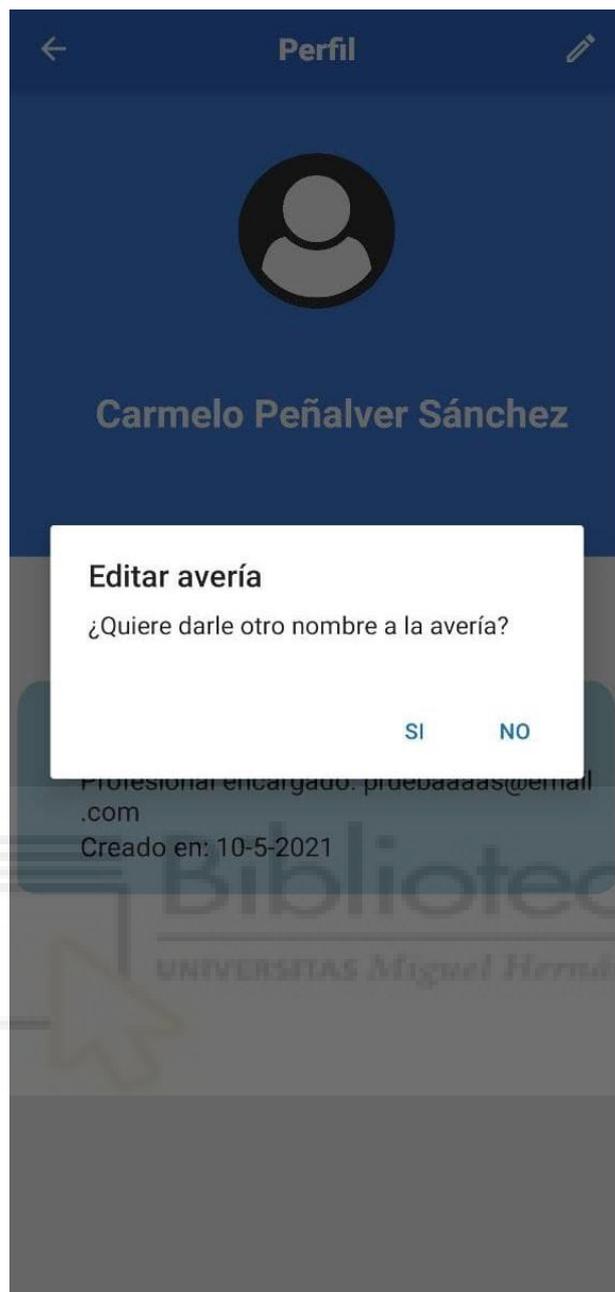


Ilustración 59. Advertencia para editar el título de la avería.

The screenshot shows a mobile application interface. At the top, there is a blue header bar with a white back arrow on the left and the text "Editar título" in white. Below the header, the main content area has a light gray background. It starts with the instruction "Ponle un título a tu avería!" in bold black text. Underneath is a white rounded rectangular text input field containing the text "Sofá roto" with a blue cursor at the end. At the bottom of the form is a gray rounded rectangular button with the text "Editar datos" in black.

Ilustración 60. Formulario para cambiarle el título a la avería.

## Averías en proceso

The screenshot shows a mobile application interface for a report card. The background is light blue with a faint "Biblioteca" watermark. The report card itself is a darker blue rounded rectangle. It contains the following text: "Sofá roto" in bold black, "Profesional encargado: pruebaaaaas@email.com", and "Creado en: 10-5-2021".

Ilustración 61. Cambios registrados tras haber registrado el nuevo título.

En cambio, si el usuario presiona la avería durante un par de segundos saltará otra advertencia diciéndole si da por hecho el trato, en caso afirmativo, se le redirigirá a una pantalla donde podrá comentar que le ha parecido el profesional y darle una puntuación sobre 5.

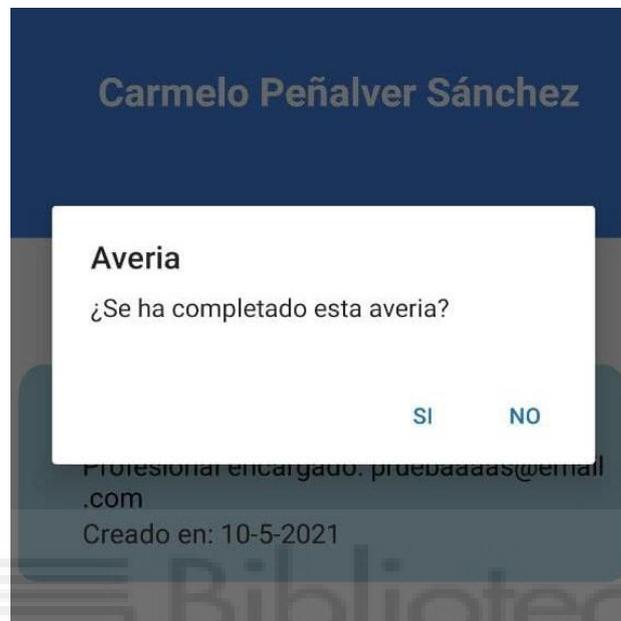


Ilustración 62. Advertencia sobre notificar la finalización de la avería.

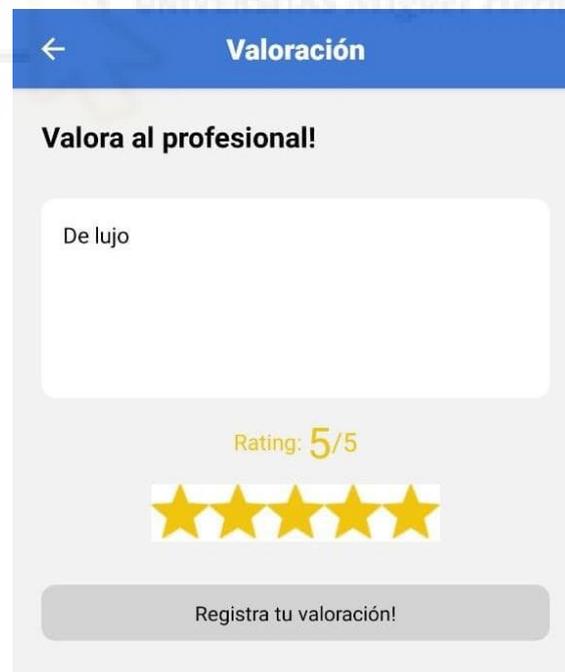


Ilustración 63. Formulario para valorar al profesional encargado de la avería.

Tras haber registrado y valorado al profesional, dicha avería saldrá en el historial de averías completadas y la valoración en la sección de valoraciones dadas.

## 7.2. Punto de vista del usuario “profesional”

A diferencia del usuario “cliente”, el usuario “profesional” tiene una pantalla principal un poco más escueta ya que no tiene por qué visualizar el listado de categorías para averías o el ranking de profesionales, así que el profesional solo tendrá dos sectores importantes, siendo estos los trabajos pendientes y el perfil.

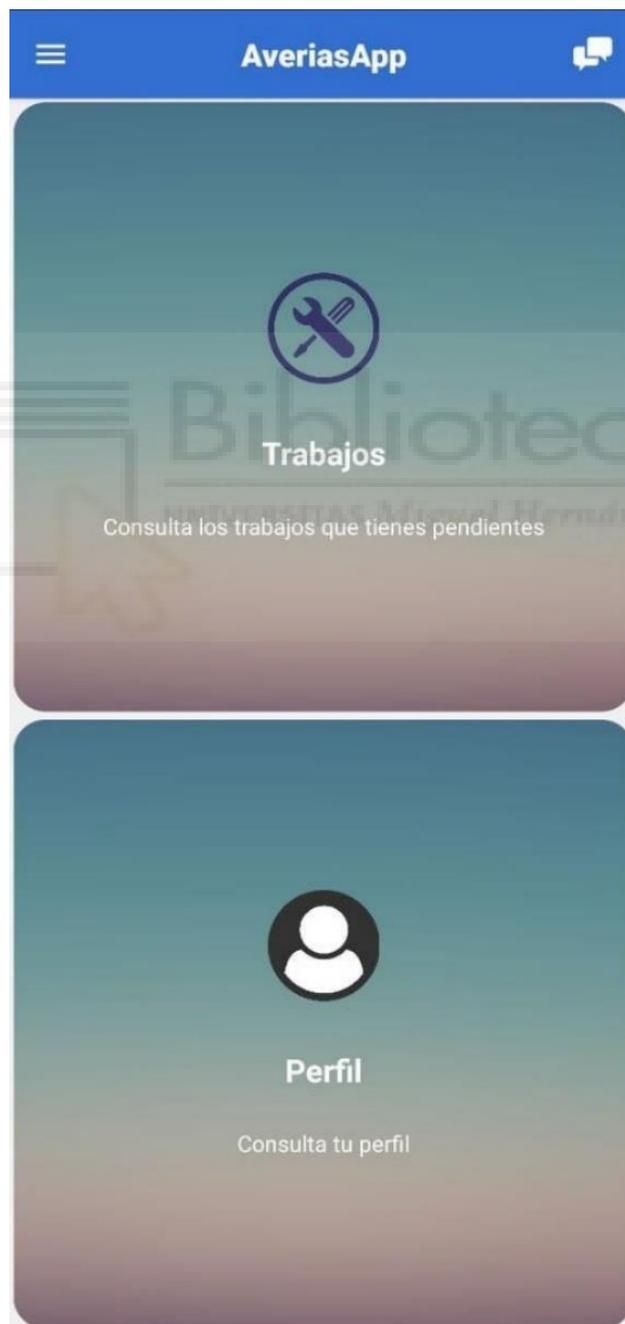


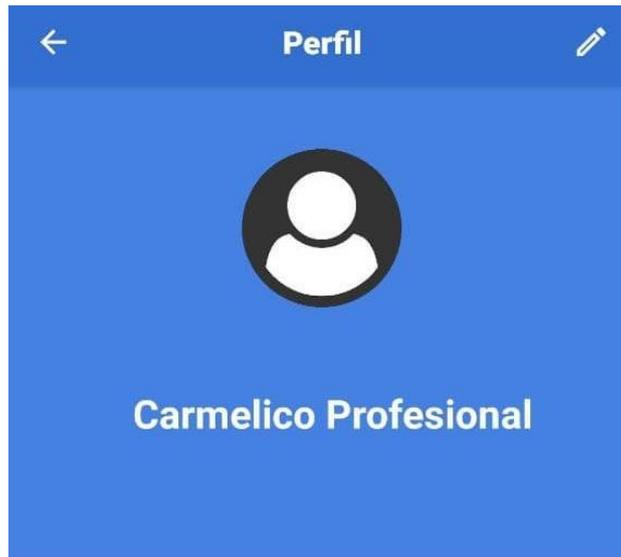
Ilustración 64. Pantalla principal del usuario “profesional”.

En la sección de trabajos encontrará todos los trabajos pendientes que tenga con cualquier usuario “cliente”, si no tiene ningún trabajo activo la app le dirá que no existe ningún trabajo pendiente.

En cuanto a la sección del perfil no hay mucha diferencia respecto al perfil del usuario “cliente”, encontrará su historial de averías terminadas y las valoraciones recibidas por los clientes. Además, el profesional podrá encontrar todas las insignias ganadas y visualizar información sobre ellas.



Ilustración 65. Perfil del profesional con el historial de averías completadas.



### Valoraciones

Cliente: Usuariotest@email.com  
Opinion: De lujo  
Valoracion dada: 5/5

Cliente: javilicoo@email.com  
Opinion: Muuuuuuy mal  
Valoracion dada: 0/5

*Ilustración 66. Perfil con el listado de valoraciones dadas por los clientes.*



*Ilustración 67. Información sobre una de las insignias del profesional.*

Como se puede observar en la pantalla principal, concretamente en la parte derecha del header, encontramos un botón con un diseño de un bocadillo donde llevará al profesional a las salas de chat que tiene con todos los clientes con los que ha mantenido conversación, sin importar si han cerrado un trato o no.



Ilustración 68. Listado de salas de chat del profesional con los clientes.

