



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E**  
**INDUSTRIAL**  
**CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN**

**Tema:**

---

APLICACIÓN MÓVIL USANDO TECNOLOGÍAS MULTIPLAFORMA PARA  
LA DIFUSIÓN Y BÚSQUEDA DE ARRIENDOS UNIVERSITARIOS EN LOS  
ALREDEDORES DE LA UNIVERSIDAD TÉCNICA DE AMBATO CAMPUS  
HUACHI

---

Trabajo de titulación modalidad Proyecto de investigación, presentado previo a la  
obtención del título de Ingeniera en Tecnologías de la Información.

**ÁREA:** Software

**LÍNEA DE INVESTIGACIÓN:** Desarrollo de software

**AUTOR:** Daniela Estefanía García Cherres

**TUTOR:** Ing. Mg. Hernán Fabricio Naranjo Ávalos

Ambato - Ecuador

agosto - 2023

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del trabajo de titulación con el tema: APLICACIÓN MÓVIL USANDO TECNOLOGÍAS MULTIPLAFORMA PARA LA DIFUSIÓN Y BÚSQUEDA DE ARRIENDOS UNIVERSITARIOS EN LOS ALREDEDORES DE LA UNIVERSIDAD TÉCNICA DE AMBATO CAMPUS HUACHI, desarrollado bajo la modalidad Proyecto de Investigación por la señorita Daniela Estefanía García Cherres, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que la estudiante ha sido tutorada durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, agosto 2023.

---

Ing. Mg. Hernán Fabricio Naranjo Ávalos

TUTOR

## AUTORÍA

El presente trabajo de titulación titulado: APLICACIÓN MÓVIL USANDO TECNOLOGÍAS MULTIPLAFORMA PARA LA DIFUSIÓN Y BÚSQUEDA DE ARRIENDOS UNIVERSITARIOS EN LOS ALREDEDORES DE LA UNIVERSIDAD TÉCNICA DE AMBATO CAMPUS HUACHI es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2023.



---

Daniela Estefanía García Cherres  
C.C. 0250060449  
AUTOR

## DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, agosto 2023.



Daniela Estefanía García Cherres  
C.C. 0250060449  
AUTOR

## **APROBACIÓN DEL TRIBUNAL DE GRADO**

En calidad de par calificador del informe final del trabajo de titulación presentado por la señorita Daniela Estefanía García Cherres, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado **APLICACIÓN MÓVIL USANDO TECNOLOGÍAS MULTIPLAFORMA PARA LA DIFUSIÓN Y BÚSQUEDA DE ARRIENDOS UNIVERSITARIOS EN LOS ALREDEDORES DE LA UNIVERSIDAD TÉCNICA DE AMBATO CAMPUS HUACHI**, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, agosto 2023.

-----  
Ing. Elsa Pilar Urrutia Urrutia, Mg.  
PRESIDENTE DEL TRIBUNAL

-----  
Ing. Paulo Cesar Torres Abril, Mg  
PROFESOR CALIFICADOR

-----  
Ing. Santiago David Jara Moya, Mg.  
PROFESOR CALIFICADOR

## **DEDICATORIA**

*El presente proyecto está dedicado a Dios por brindarme sabiduría y fortaleza durante cada paso de mi vida y la realización de este trabajo.*

*A mis padres Miguel García y Camita Cherres, quienes siempre estuvieron pendientes de mis necesidades y han sabido educarme con valores y principios que me han servido para afrontar los retos que se me han presentado durante el transcurso de la carrera.*

*A mi hermana Yessenia García por confiar en mi capacidad para culminar esta etapa, impulsándome a esforzarme siempre y perseguir mis sueños.*

*A mis queridos abuelitos, su infinito amor ha sido mi mayor inspiración. Gracias por ser mi guía y apoyo constante.*

## **AGRADECIMIENTO**

*Agradezco principalmente a Dios por brindarme salud y fortaleza para tomar buenas decisiones durante el transcurso de la carrera universitaria y a honrar el esfuerzo que mis padres han hecho para verme triunfar.*

*A mi enamorado Ismael Camacho, quien con su respeto y cariño incondicional me supo apoyar desde el inicio de la carrera hasta culminarla.*

*A mis amigos por su apoyo durante mi carrera, lo cual ha hecho que el camino sea más llevadero, especialmente a mi amigo Bryan Aldás quien supo confiar en mis capacidades y con su buen humor animarme en momentos difíciles de la universidad.*

*A los docentes quienes me han brindado su apoyo incondicional, su estímulo y su comprensión.*

## ÍNDICE GENERAL DE CONTENIDOS

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR .....	iv
APROBACIÓN DEL TRIBUNAL DE GRADO .....	v
DEDICATORIA .....	vi
AGRADECIMIENTO .....	vii
RESUMEN EJECUTIVO .....	xv
ABSTRACT.....	xvi
CAPÍTULO I.- MARCO TEÓRICO .....	1
1.1 Tema de investigación.....	1
1.1.1 Planteamiento del problema.....	1
1.2 Antecedentes investigativos .....	3
1.3 Fundamentación teórica .....	4
1.3.1 Ingeniería de software.....	4
1.3.2 Aplicación multiplataforma .....	5
1.3.3 Xamarin Forms .....	6
1.3.4 React Native.....	6
1.3.5 Flutter.....	7
1.3.6 Dart .....	7
1.3.7 Widgets en Flutter.....	8
1.3.8 Widgets básicos de Flutter .....	8
1.3.9 Estructura de una aplicación Flutter .....	9
1.3.10 Metodología de desarrollo .....	10
1.3.11 Metodologías ágiles. ....	11
1.3.12 Scrumban .....	11
1.3.13 Programación Extrema (XP).....	13
1.3.14 Mobile-D.....	13
1.3.15 Fases en la metodología Mobile-D .....	14
1.3.16 Aplicación Móvil .....	15
1.3.17 Aplicación nativa .....	15



1.3.18 Aplicación web .....	16
1.3.19 Aplicación híbrida.....	16
1.3.20 Ley del inquilinato .....	16
1.3.21 Tecnología publicitaria .....	17
1.3.22 Difusión de arriendos universitarios .....	19
1.4 Objetivos .....	20
1.4.1 Objetivo general.....	20
1.4.2 Objetivos específicos .....	20
CAPÍTULO II.- METODOLOGÍA .....	21
2.1 Materiales .....	21
2.2 Métodos .....	27
2.2.1 Modalidad de la investigación .....	27
2.2.2 Población y muestra.....	28
2.2.3 Recolección de la información .....	29
2.2.4 Procesamiento y análisis de datos.....	58
CAPÍTULO III.- RESULTADOS Y DISCUSIÓN .....	61
3.1 Análisis y discusión de los resultados .....	61
3.1.1 Impacto de los problemas en arrendadores y arrendatarios .....	61
3.1.2 Modelado de procesos de arrendadores y arrendatarios .....	62
3.1.3 Tecnologías multiplataforma para el desarrollo móvil.....	64
3.1.4 Metodologías para el desarrollo móvil. ....	65
3.2 Desarrollo de la propuesta .....	67
3.2.1 Fase 1: Exploración .....	67
3.2.2 Fase 2: Inicialización .....	72
3.2.3 Fase 3: Producción.....	87
3.2.4 Fase 4: Estabilización .....	106
3.2.5 Fase 5: Pruebas del sistema .....	107
CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES .....	122
4.1 Conclusiones .....	122
4.2 Recomendaciones .....	123

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Composición de una aplicación Flutter. ....	9
<b>Figura 2.</b> Combinación entre Kanban y Scrum. ....	11
<b>Figura 3.</b> Fases de Mobile-D. ....	14
<b>Figura 4.</b> Resultados pregunta 1 (arrendadores). ....	34
<b>Figura 5.</b> Resultados pregunta 2 (arrendadores). ....	35
<b>Figura 6.</b> Resultados pregunta 3 (arrendadores). ....	36
<b>Figura 7.</b> Resultados pregunta 4 (arrendadores). ....	37
<b>Figura 8.</b> Resultados pregunta 5 (arrendadores). ....	38
<b>Figura 9.</b> Resultados pregunta 6 (arrendadores). ....	39
<b>Figura 10.</b> Resultados pregunta 7 (arrendadores). ....	40
<b>Figura 11.</b> Resultados pregunta 8 (arrendadores). ....	41
<b>Figura 12.</b> Resultados pregunta 9 (arrendadores). ....	42
<b>Figura 13.</b> Resultados pregunta 10 (arrendadores). ....	43
<b>Figura 14.</b> Resultados pregunta 11 (arrendadores). ....	45
<b>Figura 15.</b> Resultados pregunta 1 (arrendatarios). ....	46
<b>Figura 16.</b> Resultados pregunta 2 (arrendatarios). ....	47
<b>Figura 17.</b> Resultados pregunta 3 (arrendatarios). ....	48
<b>Figura 18.</b> Resultados pregunta 4 (arrendatarios). ....	49
<b>Figura 19.</b> Resultados pregunta 5 (arrendatarios). ....	51
<b>Figura 20.</b> Resultados pregunta 6 (arrendatarios). ....	52
<b>Figura 21.</b> Resultados pregunta 7 (arrendatarios). ....	53
<b>Figura 22.</b> Resultados pregunta 8 (arrendatarios). ....	54
<b>Figura 23.</b> Resultados pregunta 9 (arrendatarios). ....	55
<b>Figura 24.</b> Resultados pregunta 10 (arrendatarios). ....	57
<b>Figura 25.</b> Resultados pregunta 11 (arrendatarios). ....	58
<b>Figura 26 .</b> Procesos para la difusión y búsqueda de arriendos. ....	63
<b>Figura 27.</b> Arquitectura de la aplicación móvil. ....	75
<b>Figura 28.</b> Colecciones en Firebase. ....	76
<b>Figura 29.</b> Subcolección de la lista de arriendos favoritos. ....	78
<b>Figura 30.</b> Storage para almacenar PDF e imágenes en Firebase. ....	78
<b>Figura 31.</b> Esquema de navegabilidad. ....	79

<b>Figura 32.</b> Diagrama de caso de uso arrendadores.....	80
<b>Figura 33.</b> Diagramas de casos de uso arrendatarios. ....	80
<b>Figura 34.</b> Pantalla inicial. ....	81
<b>Figura 35.</b> Registros arrendadores y arrendatarios.....	82
<b>Figura 36.</b> Inicio de sesión arrendadores y arrendatarios. ....	82
<b>Figura 37.</b> Mapa de búsqueda. ....	83
<b>Figura 38.</b> Lista de cuarto de arriendo. ....	83
<b>Figura 39.</b> Filtro de búsqueda. ....	84
<b>Figura 40.</b> Detalle del cuarto de arriendo.....	84
<b>Figura 41.</b> Añadir cuarto. ....	85
<b>Figura 42.</b> Detalles del cuarto. ....	85
<b>Figura 43.</b> Arriendos favoritos. ....	86
<b>Figura 44.</b> Ajustes de cuenta. ....	86
<b>Figura 45.</b> Cerrar sesión. ....	87
<b>Figura 46.</b> Archivo de configuración de Firebase.....	88
<b>Figura 47.</b> Dependencias usadas en Flutter.....	91
<b>Figura 48.</b> Splash Screen.....	108
<b>Figura 49.</b> Pantalla para registro de usuario.....	108
<b>Figura 50.</b> Pantalla para inicio de sesión.....	109
<b>Figura 51.</b> Marcadores en la lista de arriendos. ....	110
<b>Figura 52.</b> Detalle de la publicación. ....	110
<b>Figura 53.</b> Trazo de la ruta. ....	111
<b>Figura 54.</b> Publicación detallada.....	111
<b>Figura 55.</b> Ajustes de perfil.....	112
<b>Figura 56.</b> Filtros de búsqueda.....	113
<b>Figura 57.</b> Paso 1, precio, tamaño, ubicación, amoblado, cocina. ....	114
<b>Figura 58.</b> Paso 1, adjuntar pdf e imágenes. ....	114
<b>Figura 59.</b> Paso 2, servicios básicos y garantía.....	115
<b>Figura 60.</b> Paso 3, género, permiso mascotas, descripción.....	116
<b>Figura 61.</b> Edición y habilitación de arriendos. ....	116
<b>Figura 62.</b> Ajustes perfil arrendatario. ....	117
<b>Figura 63.</b> Lista favoritos.....	117
<b>Figura 64.</b> Lista de arriendos en el mapa. ....	118

<b>Figura 65.</b> Detalle de la publicación. ....	118
<b>Figura 66.</b> Trazo de la ruta hacia los arriendos. ....	119
<b>Figura 67.</b> Publicación detallada del arriendo. ....	119
<b>Figura 68.</b> Filtro de la publicación. ....	120
<b>Figura 69.</b> Lista de arriendos disponibles. ....	120
<b>Figura 70.</b> Detalles del contacto del arrendador. ....	121

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Descripción de widgets básicos de Flutter. ....	8
<b>Tabla 2.</b> Actividades que realiza Scrum, Kanban y Scrumban. ....	12
<b>Tabla 3.</b> Población arrendadores. ....	28
<b>Tabla 4.</b> Simbología de la fórmula de muestra.....	29
<b>Tabla 5.</b> Confiabilidad Kuder-Richardson en la encuesta para arrendadores. ....	30
<b>Tabla 6.</b> Interpretación de la escala de consistencia de Kuder-Richardson. ....	30
<b>Tabla 7.</b> Confiabilidad alfa de Cronbach en la encuesta para arrendadores. ....	31
<b>Tabla 8.</b> Interpretación de la escala de consistencia del alfa de Cronbach. ....	31
<b>Tabla 9.</b> Confiabilidad Kuder-Richardson en la encuesta para arrendatarios. ....	32
<b>Tabla 10.</b> Confiabilidad alfa de Cronbach en la encuesta para arrendatarios. ....	33
<b>Tabla 11.</b> Resultados pregunta 1 (arrendadores).....	33
<b>Tabla 12.</b> Resultados pregunta 2 (arrendadores).....	34
<b>Tabla 13.</b> Resultados pregunta 3 (arrendadores).....	36
<b>Tabla 14.</b> Resultados pregunta 4 (arrendadores).....	37
<b>Tabla 15.</b> Resultados pregunta 5 (arrendadores).....	38
<b>Tabla 16.</b> Resultados pregunta 6 (arrendadores).....	39
<b>Tabla 17.</b> Resultados pregunta 7 (arrendadores).....	40
<b>Tabla 18.</b> Resultados pregunta 8 (arrendadores).....	41
<b>Tabla 19.</b> Resultados pregunta 9 (arrendadores).....	42
<b>Tabla 20.</b> Resultados pregunta 10 (arrendadores).....	43
<b>Tabla 21.</b> Resultados pregunta 11 (arrendadores).....	44
<b>Tabla 22.</b> Resultados pregunta 1 (arrendatarios).....	46
<b>Tabla 23.</b> Resultados pregunta 2 (arrendatarios).....	47
<b>Tabla 24.</b> Resultados pregunta 3 (arrendatarios).....	48
<b>Tabla 25.</b> Resultados pregunta 4 (arrendatarios).....	49
<b>Tabla 26.</b> Resultados pregunta 5 (arrendatarios).....	50
<b>Tabla 27.</b> Resultados pregunta 6 (arrendatarios).....	51
<b>Tabla 28.</b> Resultados pregunta 7 (arrendatarios).....	53
<b>Tabla 29.</b> Resultados pregunta 8 (arrendatarios).....	54
<b>Tabla 30.</b> Resultados pregunta 9 (arrendatarios).....	55
<b>Tabla 31.</b> Resultados pregunta 10 (arrendatarios).....	56

<b>Tabla 32.</b> Resultados pregunta 11 (arrendatarios).....	57
<b>Tabla 33.</b> Problemas y su impacto en el arrendador y arrendatario. ....	62
<b>Tabla 34.</b> Comparación entre tecnologías multiplataforma. ....	65
<b>Tabla 35.</b> Metodologías para el desarrollo móvil.....	66
<b>Tabla 36.</b> Establecimiento de los interesados.....	67
<b>Tabla 37.</b> Requisitos funcionales de los arrendadores. ....	69
<b>Tabla 38.</b> Requisitos funcionales de los arrendatarios. ....	71
<b>Tabla 39.</b> Planificación de fases. ....	75
<b>Tabla 40.</b> Estructura de la Colección publicaciones. ....	76
<b>Tabla 41.</b> Estructura de la Colección usuarios. ....	77
<b>Tabla 42.</b> Estructura de la Subcolección misFavoritos. ....	77
<b>Tabla 43.</b> Storycard - Pantalla principal.....	92
<b>Tabla 44.</b> Registro e inicio de sesión de usuarios. ....	92
<b>Tabla 45.</b> Módulo de registro de usuario. ....	93
<b>Tabla 46.</b> Módulo de mapas. ....	95
<b>Tabla 47.</b> Módulo de perfil de usuario. ....	96
<b>Tabla 48.</b> Función publicar arriendos.....	97
<b>Tabla 49.</b> Listado de arriendos publicados opción propietarios.....	101
<b>Tabla 50.</b> Módulo borrar y editar arriendos. ....	102
<b>Tabla 51.</b> Visualización rápida de detalles del arriendo en el mapa. ....	104
<b>Tabla 52.</b> Funcionalidad de arriendos favoritos y filtros. ....	105
<b>Tabla 53.</b> Implementación de la función de geolocalización.....	106

## RESUMEN EJECUTIVO

La búsqueda de arriendos universitarios constituye una de las necesidades principales para los estudiantes provenientes de provincias durante su período de estudio. Sin embargo, este proceso se ha tornado complicado debido a la descentralización de información por parte de los arrendadores. Se ha observado una diversidad de medios de difusión empleados, como carteles, redes sociales, afiches en postes y paredes. Esta dispersión de información dificulta la recopilación y acceso eficiente a la información.

El presente proyecto tiene como objetivo el desarrollo de una aplicación móvil multiplataforma dirigida a la comunidad estudiantil. Su propósito radica en proporcionar información precisa y actualizada de las opciones de alojamiento disponibles en los alrededores de la Universidad Técnica de Ambato, Campus Huachi. Además, se pretende optimizar y agilizar la difusión de los arriendos disponibles en esta área geográfica.

La aplicación móvil se desarrolló utilizando Flutter, aprovechando su eficiencia como tecnología multiplataforma que se basa en el lenguaje de programación Dart. En el proceso de desarrollo del backend, se ha empleado Firebase para el alojamiento y gestión de la base de datos, así como para la autenticación de usuarios y para facilitar el envío de notificaciones relacionada con los arriendos universitarios. Además, la metodología utilizada fue Mobile-D, la cual ha permitido alcanzar un desarrollo ágil y eficiente mediante entregas periódicas de funcionalidades.

Como resultado de este proceso, se logró obtener una aplicación móvil que optimiza la búsqueda y difusión de arriendos universitarios en los alrededores de la Universidad Técnica de Ambato, mejorando la experiencia de los estudiantes para encontrar un cuarto de arriendo cercano al campus universitario.

**Palabras clave:** Aplicación móvil, arrendadores, arrendatarios, Framework, Flutter, mobile-D.

## ABSTRACT

The search for university leases is one of the main needs of students coming from the provinces during their period of study. However, this process has become complicated due to the decentralization of information by landlords. A diversity of means of dissemination has been observed, such as posters, social networks, and posters on poles and walls. This dispersion of information makes it difficult to gather and access information efficiently.

The objective of this project is the development of a multiplatform mobile application aimed at the student community. Its purpose is to provide precise and updated information about the lodging options available in the surroundings of the Universidad Técnica de Ambato, Campus Huachi. In addition, it aims to optimize and streamline the diffusion of available rentals in this geographic area.

The mobile application was developed using Flutter, taking advantage of its efficiency as a multiplatform technology based on the Dart programming language. In the backend development process, Firebase was used to host and manage the database, as well as for user authentication and to facilitate the sending of notifications related to university leases. In addition, the methodology used was Mobile-D, which has allowed to achieve an agile and efficient development through periodic deliveries of functionalities.

As a result of this process, a mobile application was obtained that optimizes the search and dissemination of university leases in the surroundings of the Universidad Técnica de Ambato, improving the students' experience in finding a room for rent near the university campus.

**Keywords:** Mobile application, landlords, tenants, Framework, Flutter, mobile-D.



## **CAPÍTULO I.- MARCO TEÓRICO**

### **1.1 Tema de investigación**

APLICACIÓN MÓVIL USANDO TECNOLOGÍAS MULTIPLAFORMA PARA LA DIFUSIÓN Y BÚSQUEDA DE ARRIENDOS UNIVERSITARIOS EN LOS ALREDEDORES DE LA UNIVERSIDAD TÉCNICA DE AMBATO CAMPUS HUACHI

#### **1.1.1 Planteamiento del problema**

En la actualidad la capacidad de establecer comunicaciones digitales se ha incrementado especialmente desde el surgimiento de los teléfonos inteligentes, los mismos que resumen todas las formas posibles de información y comunicación en un solo dispositivo [1]. El apareamiento de las Tecnologías de la Información y la Comunicación (TIC) ha mejorado la calidad de vida de las personas, debido a la rapidez con la que se puede comunicar unas con otras y el ingreso al espacio de datos e información llamado Internet [2]. Los medios digitales transmiten información en formatos como: audio, video, texto y fotografía, ocupando el lugar de los medios tradicionales como la radio, los periódicos o la televisión, describiéndolos en Ecuador como una necesidad de estrategia para la difusión [3].

Los estudiantes universitarios que se trasladan a otras ciudades por motivos de realizar sus estudios se ven en la necesidad de buscar un inmueble de alquiler. La demanda de ciertos requisitos al momento de elegir el tipo de habitación, casa o departamento donde puedan hospedarse, tienden a ser dificultosas acorde a sus posibilidades económicas. Por otro lado, los arrendadores invierten poco dinero en publicidad al promover los inmuebles en renta, lo cual limita significativamente el alcance de difusión. Su alcance de información es sesgada, dirigiéndose solo a personas que visualizan el anuncio de manera presencial, generando esfuerzos infructuosos de búsqueda para los posibles arrendatarios [4].

Nicolás Izquierdo, subgerente de Estudios de portalinmobiliario.com, señala que la mayoría de los estudiantes que arriendan una vivienda, específicamente

departamentos, son jóvenes que fluctúan entre 18 y 25 años. Los estudiantes buscan vivir cerca del lugar donde estudian, no consideran relevante que tenga estacionamiento, además que exista acceso a restaurantes. Por ende, el nivel socioeconómico, se encuentra asociado a los gastos comunes, cuentas, alimentación, transporte, entre otros, lo que implica un ítem importante del presupuesto familiar [5].

El usuario digital en Ecuador ha evolucionado y se ha adaptado al crecimiento de la tecnología producto del confinamiento, inclinándose por actividades de comunicación, educación, entretenimiento, compras en línea, servicios públicos, investigación, consumo y producción de contenidos. Entre los principales aceleradores de este proceso se encuentra la seguridad, transparencia, inclusividad, respeto al medio ambiente, comunicación integrada en medios digitales y sin conexión a internet, generando buena experiencia y confianza para el usuario. El 33% de usuarios de internet en Ecuador está concentrado en Quito y Guayaquil, el 59% del total son mayores de 24 años (mostrando incremento de menores de 24 años en comparación con informe de enero 2020 que representaba el 63%) y registran el 98% de ingreso e interacción en redes sociales a través de dispositivos móviles [6].

Según Juan Pablo Del Alcázar Ponce, el uso de la tecnología en Ecuador a través de la encuesta Ecuador Digital, realizada en octubre del 2022, con una población de 610.000 personas, determinó que el 3% de los usuarios de internet en Ecuador están concentrados en la provincia de Tungurahua [6]. En cuanto a la encuesta ENEMDU (Encuesta Nacional de Empleo, Desempleo y Subempleo) realizada por el INEC (Instituto Nacional de Estadística y Censos), dirigida a 192.359 personas en la ciudad de Ambato, determinó que el 50% de la población mayores a los 5 años, tienen acceso a internet y recursos tecnológicos, con ello se evidencia que las cifras significativas son considerables en la interacción de las personas con la tecnología [7].

En efecto, las nuevas tecnologías, redes de comunicación de alta velocidad, el uso de redes sociales, el número de usuarios que disponen de teléfonos inteligentes, entre otros, han facilitado la oferta de productos o servicios, incrementando la probabilidad de llegar a más usuarios por medio de sus dispositivos móviles.

Según el estudio de la Universidad Técnica de Ambato (UTA) sobre la deserción estudiantil universitaria, se da fundamentalmente por tres áreas: económico, personal y académico. El factor económico, representa el 40%, se refleja en el costo diario de vida universitaria de 1 a 20 dólares, cubriendo gastos como transporte, alimentación, materiales. A pesar de que el Estado ecuatoriano proporciona un modelo de educación gratuita, las limitaciones económicas afectan el desempeño académico, lo que conlleva al abandono estudiantil, impactando el desarrollo económico y social del país [8].

## **1.2 Antecedentes investigativos**

Los antecedentes de la investigación están basados en la búsqueda de previos estudios relacionados con el desarrollo de aplicaciones móviles, búsqueda de arriendos, prestación de servicios, entre otros los cuales sirven como referencia para justificar el desarrollo del presente proyecto. A continuación, se exponen aquellos documentos relevantes:

Según Jaramillo, Camila Soledad y Sánchez Ostaiza, Yajaira, los dispositivos móviles se han convertido es una herramienta esencial en la rutina de las personas. Por esta razón, el uso de estos dispositivos para la difusión de información y la creación de canales de comunicación se muestra valioso para la construcción de relaciones. Resalta la relevancia de los dispositivos móviles al posibilitar la implementación de nuevas estrategias comunicativas que promueven una mayor interacción entre los consumidores y los proveedores [9].

Según Reinoso López, David Vinicio y Quisimalín Santamaría, Mauricio, la mayoría de los propietarios de pequeñas y medianas empresas (PYMES) consideran que la falta de conocimiento acerca de las herramientas, la carencia de personal capacitado y las dificultades para comprender el funcionamiento de las plataformas digitales son factores determinantes que los llevan a evitar la adopción del comercio electrónico. Por lo tanto, este estudio refleja la limitada comprensión por parte del grupo investigado respecto al uso de herramientas tecnológicas, lo cual resulta en la oportunidad desaprovechada para promover sus productos o servicios a través de estas plataformas [10].

En el trabajo de Burbano Santamaría, María Elena, se aboga por fomentar la adopción de tecnologías innovadoras de múltiples plataformas. Estas tecnologías tienen el propósito de agilizar y elevar la calidad en la creación de aplicaciones móviles compatibles con diversos sistemas operativos. Durante un largo periodo, este proceso había representado un área confusa y de gran complejidad. Esto resalta la relevancia de incorporar nuevas tecnologías de desarrollo en el entorno móvil, estableciendo así una base sólida para investigaciones venideras [11].

En la investigación realizado por Matango Guajan, Wilman Gerardo, resalta que la creación de aplicaciones móviles en un entorno multiplataforma como Xamarin Forms conlleva ventajas económicas notables. Esto se refleja tanto en la economización del proceso de desarrollo, ya que se emplea un único código de programación, como en la reducción de los costos, al producir productos finales para diversas plataformas móviles utilizando un solo código fuente. A partir de este estudio, se resalta el valor de utilizar Xamarin Forms en proyectos móviles, ya que permite la ejecución en plataformas como iOS y Android, lo que la convierte en una solución versátil para los desarrolladores [12].

En la investigación de Torres Perdigón, Juan Esteban, se destaca que convertirse en proveedores de servicios para empresas inmobiliarias o de alquiler de departamentos aporta un valor agregado para el usuario final. Esto no solo implica un cambio de perspectiva en la relación entre competidores y colaboradores, sino que también resalta la ventaja de utilizar una aplicación móvil en este contexto [4].

### **1.3 Fundamentación teórica**

#### **1.3.1 Ingeniería de software**

Es el área donde se desenvuelven los profesionales utilizando matemáticas, física, química y conocimientos tecnológicos para resolver problemas y / o generar productos. Cada vez más artículos integran elementos de software, es crucial tener en cuenta que, de manera similar a cómo los físicos contribuyen a la base de la Ingeniería Eléctrica, la ciencia informática debe ser un respaldo para la Ingeniería de Software, y no mezclarse con ella. El objetivo es formar a los ingenieros para que estén capacitados

en la utilización de los conocimientos tecnológicos generados por la ciencia de la computación para construir productos de software [13].

Con el término "software" hace referencia al conjunto de aplicaciones informáticas que se desarrollan en el contexto de una computadora, y generalmente se dividen en tres categorías:

- Programas de control, responsables de supervisar y gestionar la ejecución de todas las actividades y procesos en la computadora, como el Sistema Operativo.
- Programas de proceso, destinados a permitir al usuario crear sus propios programas, como compiladores, intérpretes y ensambladores.
- Programas de aplicación, creados por y para los usuarios de la computadora para abordar problemas específicos y resolver necesidades concretas [14].

Los primeros dos tipos son denominados "software del sistema" debido a que son programas creados con el propósito de asegurar el funcionamiento adecuado de la computadora. Por otro lado, el tercer tipo se identifica como "software de aplicación", ya que consisten en programas diseñados para abordar los desafíos de los usuarios mediante soluciones computacionales, conocidas como aplicaciones informáticas. [14].

### **1.3.2 Aplicación multiplataforma**

Se denomina aplicación multiplataforma a un tipo de software capaz de funcionar en distintos sistemas operativos. El propósito de la creación multiplataforma es mejorar la relación entre los costos y beneficios al compartir el mismo código entre versiones destinadas a diferentes plataformas. Entre sus ventajas se encuentran: un menor tiempo y gasto de desarrollo, características similares a las aplicaciones nativas con acceso al hardware del dispositivo y la disponibilidad de herramientas de desarrollo sólidas como Visual Studio. También se puede optar por emplear tecnologías conocidas por los desarrolladores web, como HTML5, Javascript y CSS, lo que les permite trasladar su experiencia al ámbito móvil. No obstante, es importante mencionar que el rendimiento de las aplicaciones y la interfaz de usuario pueden afectar la experiencia del usuario [15].

### **1.3.3 Xamarin Forms**

Xamarin es una plataforma de código abierto destinada a la creación de aplicaciones modernas, con mejor rendimiento tanto para iOS, Android y Windows utilizando la infraestructura de .NET. Actuando como una capa de abstracción, Xamarin se encarga de la gestión de la interacción entre el código compartido y el código específico de cada plataforma. Esta tecnología opera en un entorno administrado que conlleva beneficios tales como el manejo de la asignación de memoria y la eliminación de elementos no empleados [16].

Según Jim Bennett, citado por Rodríguez [17], con Xamarin, se puede usar C# para implementar la lógica de negocio, así como también proporciona controles de interfaz personalizados para cada plataforma, haciendo que el porcentaje de código compartido sea mucho menor que en otras soluciones, como Flutter.

La cantidad de desarrolladores que usan esta tecnología es amplia, proporcionando recursos y soluciones en línea. Además, cuenta con una comunidad activa para resolver problemas. Así como también el respaldo y mantenimiento por Microsoft. Con este soporte y conocimientos experimentados en C# y .Net la curva de aprendizaje es media [18].

### **1.3.4 React Native**

Según Bonnie Eisenman citado por Rodríguez [17], otra herramienta para el desarrollo de aplicaciones multiplataforma es React Native el cual es un framework usado para el desarrollo de plataformas móviles como también para soluciones Windows. El fin es utilizar la librería React para la construcción de interfaces de usuario basados en HTML.

Permite usar las librerías que se utilizan para el desarrollo de aplicaciones web en plataformas móviles, tanto para la interfaz y código JavaScript en la lógica, logrando una herramienta atractiva para organizaciones y desarrolladoras que posean

aplicaciones web implementadas en dichas tecnologías. React Native presenta una curva de aprendizaje alta, dificultando su adopción para nuevos usuarios[17].

### **1.3.5 Flutter**

Este framework utiliza un lenguaje de programación compilado conocido como Dart. El código escrito en este lenguaje se compila previamente (AOT) en código nativo específico para cada plataforma. Esto posibilita que las aplicaciones desarrolladas con Flutter puedan interactuar directamente con la plataforma, sin requerir la intermediación de un puente. La conversión del código en código nativo para las plataformas resulta en un rendimiento superior para las aplicaciones construidas con Flutter en comparación con las desarrolladas utilizando React Native.[17].

Flutter se distingue como el único SDK de desarrollo móvil que permite la creación de interfaces reactivas sin la necesidad de recurrir a un puente de JavaScript. Esto lo diferencia como la elección sobresaliente frente a herramientas como Xamarin y React Native [17].

Como SDK compatible por Google, Flutter proporciona documentación detallada que es fácil de implementar, ya que permite leer documentos estándar, capacitarse con videos e incluso completar ejercicios para dominar sus habilidades. Flutter tiene una ligera curva de aprendizaje leve y cualquier persona que haya usado C# y Java antes, descubrirá rápidamente cómo codificar en él [19].

### **1.3.6 Dart**

Según Dzenan Ridjanovic e Ivo Balbaert, citado por Rodríguez [17], Dart representa un lenguaje de programación de código abierto desarrollado por Google en 2011, con el propósito de ofrecer una opción más moderna en comparación con JavaScript, destinada a mejorar la eficiencia en la creación de interfaces de usuario.

Su flexibilidad resulta altamente adecuada para su implementación junto a Flutter. Dart posibilita la compilación Ahead Of Time (AOT), lo que conlleva a la obtención de

código nativo con un desempeño notablemente superior al que se lograría empleando un lenguaje interpretado como JavaScript. Además, el código escrito en Dart también puede ser compilado en Tiempo de Ejecución (JIT), permitiendo así el respaldo a la función de recarga instantánea de Flutter, la cual facilita la actualización del código durante la ejecución. [17].

Dart se caracteriza por su legibilidad y su capacidad de facilitar la visualización, evitando la necesidad de recurrir a otro lenguaje como XML o JSX al momento de crear interfaces [17].

### 1.3.7 Widgets en Flutter

Para Flutter los widgets son los componentes básicos de la interfaz de usuario de una aplicación. Forman una jerarquía basada en la composición. Cada widget anida dentro de su padre y puede recibir contexto del padre [20].

Los Widgets describen la apariencia que debería tener la interfaz de usuario (UI) considerando su configuración y estado. Si el estado de un widget se modifica, se reconstruye su descripción, donde el marco de trabajo contrasta la descripción actual con la anterior para discernir los cambios mínimos requeridos en la estructura de renderizado subyacente, facilitando la transición de un estado al siguiente [21].

### 1.3.8 Widgets básicos de Flutter

<b>Widget</b>	<b>Descripción</b>
<b>Text</b>	Permite crear una cadena de texto con estilo, dentro de la aplicación.
<b>Row, Column</b>	Permiten crear layouts flexibles tanto en la dirección horizontal (Row) como en la vertical (Column).
<b>Stack</b>	Permite apilar los widgets uno encima del otro.
<b>Container</b>	Permite crear un elemento visual rectangular.

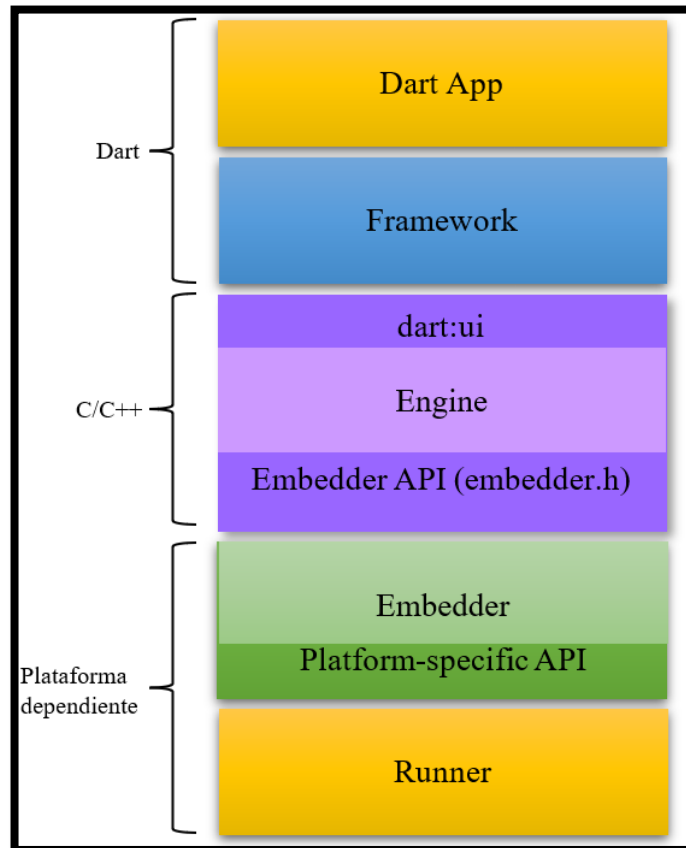
**Tabla 1.** Descripción de widgets básicos de Flutter.

**Elaborado por:** La investigadora.



### 1.3.9 Estructura de una aplicación Flutter

La **Figura 1**, muestra la ubicación de Flutter Engine en la pila e identifica los repositorios individuales, lo que compone una aplicación Flutter.



**Figura 1.** Composición de una aplicación Flutter.

**Elaborado por:** La investigadora.

Cada capa ofrece funciones diferentes las cuales, integradas en una sola pila, constituye la funcionalidad completa de Flutter, entre algunas de las funciones de las capas están:

Dart App

- Ofrece widgets en la UI deseada.
- Implementa la lógica de negocio.
- Propiedad del desarrollador de la aplicación.

#### Framework

- Proporciona aplicaciones de alta calidad con ayuda de una API de nivel superior.
- Compone el árbol de widgets de la aplicación en una escena.

#### Engine

- Rasteriza escenas compuestas.
- Posee una implementación de bajo nivel de las API's principales de Flutter.
- Expone su funcionalidad al framework usando el API dart:ui.
- Se integra con una plataforma específica usando Embedder API.

#### Embedder

- Coordina con el sistema operativo subyacente, accediendo a servicios como superficies de representación, accesibilidad y entrada.
- Administra el bucle de eventos.
- La API de la plataforma la integra con el Embedder.

#### Runner

- Específica de la plataforma de Embedder en un paquete de aplicación ejecutable.
- Parte de la plantilla de la aplicación generada por flutter create.

### **1.3.10 Metodología de desarrollo**

Una metodología de desarrollo de sistemas o software (SDM) se puede definir como una colección documentada de políticas, procesos y procedimientos para mejorar el proceso de desarrollo de software con respecto al aumento de la productividad del personal de Tecnología de la Información (TI) y obtener mayor calidad de las soluciones de TI finales [22].

Usualmente, al crear aplicaciones móviles, se tiende a seleccionar metodologías ágiles debido a las cualidades que brindan para ajustarse a las nuevas situaciones que surgen

durante el desarrollo del proyecto. Las metodologías dirigidas al desarrollo de aplicaciones móviles se basan en la orientación del desarrollo ágil. A diferencia de los métodos tradicionales, este enfoque permite una mayor adaptabilidad a los cambios, y el período para entregar o producir resultados es más breve. [22].

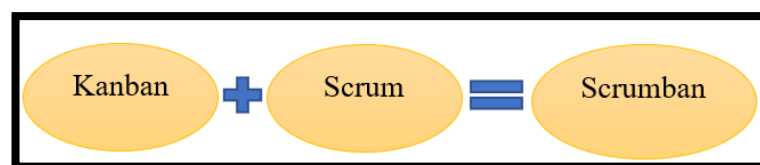
### 1.3.11 Metodologías ágiles.

En la investigación de Y. Daniel, A. Balaguera, y D. A. Balaguera, citado por [23]: Las metodologías ágiles surgieron en respuesta a las metodologías tradicionales , que presentaban características como pesadez, poca o inexistente flexibilidad al cambio, reacción lenta y excesiva producción de documentación.

Estas prácticas ágiles para ser catalogadas como un método de desarrollo empleado de manera eficiente en la industria de aplicaciones móviles tienen presentar los siguientes rasgos: Agilidad, conciencia de mercado, soporte de la línea de productos de software, desarrollo basado en arquitectura, soporte para la reutilización, inclusión de sesiones de repaso y aprendizaje, y especificación temprana de la arquitectura física [23].

### 1.3.12 Scrumban

En la investigación de Pérez, citado por [24], la metodología Scrum deriva de Scrum y Kanban. Es una metodología híbrida que contempla componentes y conceptos de estas metodologías que se contemplan entre sí, optimizando el proceso de desarrollo y adoptar una combinación de reglas que ambas metodologías por separado no permiten.



**Figura 2.** Combinación entre Kanban y Scrum.

**Elaborado por:** La investigadora.

Scrumban ofrece una mayor flexibilidad al permitir comenzar con una base sencilla y avanzar gradualmente hacia una estructura más compleja. Además, es conocido por

ser apropiado en situaciones donde surgen errores de programación inesperados a lo largo de todo el proceso de desarrollo del producto. En todos estos escenarios, la utilización exclusiva de los sprints en el enfoque Scrum no resulta viable [24].

Según T. Dyba, T. Dingsoyr, y N. B. Moe citados por [23], concluye que Scrumban es la gestión de proyectos Kanban y la simplificación de Scrum, manteniendo las reuniones diarias de Scrum y el tablero Kanban, eliminando las actividades de planificación y medición de la velocidad. Orientado hacia proyectos más simples: el tablero Kanban presenta cada tarea aisladamente, por lo que no hay una manera fácil de representar interacciones complejas entre tareas, no existen herramientas especiales o convenciones de informes; no hay necesidad de grandes programas de capacitación.

La **Tabla 2**, describe las principales actividades que se realiza en Scrum, Kanban y su metodología derivada Scrumban.

Scrum	Kanban	Scrumban
<ul style="list-style-type: none"> <li>• Roles: Cliente, equipo (con los perfiles necesarios).</li> <li>• Reuniones: reunión diaria.</li> <li>• Herramientas: tablero [24].</li> </ul>	<ul style="list-style-type: none"> <li>• Flujo visual.</li> <li>• Hacer lo que sea necesario, cuando sea necesario y solo la cantidad necesaria.</li> <li>• Limitar la cantidad de trabajo (WIP).</li> <li>• Optimización del proceso [24].</li> </ul>	<ul style="list-style-type: none"> <li>• Visualizar el flujo de trabajo.</li> <li>• Cola de trabajo.</li> <li>• Limitar el trabajo en progreso (WIP).</li> <li>• Reglas explícitas [24].</li> </ul>

**Tabla 2.** Actividades que realiza Scrum, Kanban y Scrumban.

**Elaborado por:** La investigadora.

### **1.3.13 Programación Extrema (XP)**

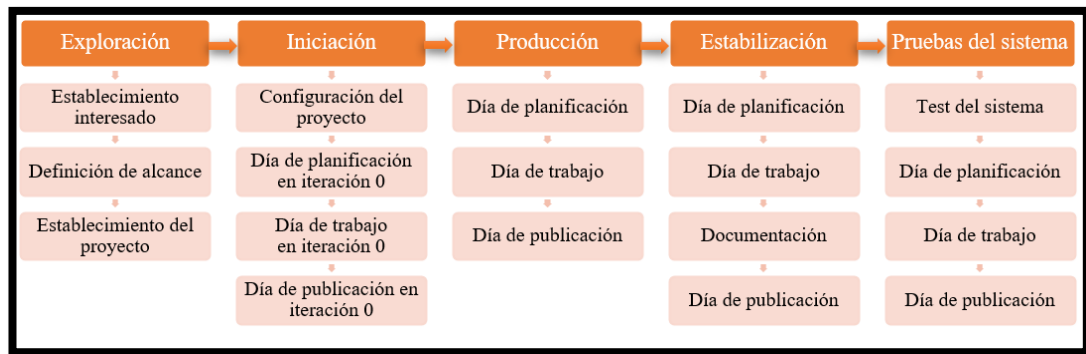
XP representa una metodología ágil que pone su enfoque en potenciar las relaciones interpersonales como elemento fundamental para lograr el exitoso desarrollo de software. Su énfasis está en fomentar la colaboración entre equipos, promover el aprendizaje de los desarrolladores y crear un entorno laboral positivo. XP se sustenta en el constante intercambio de retroalimentación entre el cliente y el equipo de desarrollo, garantizando una comunicación fluida entre todas las partes involucradas. Además, se destaca por la simplicidad en las soluciones adoptadas y afrontar con valentía las modificaciones y ajustes que puedan surgir. Es apta para proyectos con requisitos poco definidos y riesgo técnico alto [25].

El desafío en la creación de software radica en la creciente complejidad que surge de las demandas en constante evolución por parte del cliente. Las modificaciones en el proyecto resultan en retrasos y sobrepasos presupuestarios, lo que aumenta su nivel de complejidad. Esto resalta la necesidad de una sólida comprensión y aplicación de enfoques ágiles, así como una comunicación y colaboración eficiente entre los miembros del equipo [23].

### **1.3.14 Mobile-D**

Según Abrahamsson, citado por Amaya [26], las prácticas relacionadas con Mobile-D incluyen desarrollo basado en pruebas, la programación en parejas, integración continua y refactorización, así como las tareas de mejora de procesos de software; Mobile-D debe ser utilizado por un equipo de no más de diez desarrolladores que trabajen en conjunto durante no más de diez semanas para suministrar un producto. El método se basa en prácticas ágiles como Extreme Programming y crystal.

Aplicando Mobile-D se obtiene un producto funcional lo más rápido posible, ya que es una metodología de resultados rápidos. Sin embargo, como en todas las metodologías ágiles, el equipo de desarrollo debe ser lo suficientemente maduro para tomar decisiones y estar lo suficientemente capacitado en las tecnologías a utilizar para asegurar los tiempos estimados y planificados [27].



**Figura 3.** Fases de Mobile-D.

**Elaborado por:** La investigadora.

### 1.3.15 Fases en la metodología Mobile-D

#### Fase 1: Exploración

- Definición de los interesados.
- Definición de alcance.
- Plan del proyecto.

#### Fase 2: Iniciación

- Descripción del software.
- Configuración del entorno de desarrollo.
- Establecimiento de requisitos iniciales.
- Diseño de mockups.
- Diagramas de casos de uso.

#### Fase 3: Producción

- Días de planificación: analizar, mejorar y priorizar los requisitos.
- Días de trabajo.
- Días de lanzamiento o publicación.

#### Fase 4: Estabilización

- Puesta en marcha de la aplicación.
- Documentación.

## **Fase 5: Pruebas del sistema**

- Pruebas de unidad.
- Pruebas de aceptación [28].

### **1.3.16 Aplicación Móvil**

Se define como aplicación móvil a aquel software diseñado para dispositivos móviles. El término "móvil" alude a la capacidad de acceder a datos, aplicaciones y dispositivos desde cualquier ubicación y en cualquier momento. Estas aplicaciones son elaboradas considerando las restricciones inherentes de los dispositivos, como la capacidad de cómputo limitada, el almacenamiento reducido y la limitación del ancho de banda, entre otros. Los dispositivos móviles son lo suficientemente ligeros para ser transportados por las personas y están equipados con una capacidad de batería adecuada para operar de manera autónoma [29].

### **1.3.17 Aplicación nativa**

Las aplicaciones nativas son aquellas que se desarrollan y optimizan utilizando lenguajes de programación específicos para los sistemas operativos en los que serán ejecutadas. Estas aplicaciones cuentan con diversas cualidades positivas, como el acceso completo a todas las funciones del dispositivo (como cámara, GPS, acelerómetro y calendario), un alto rendimiento, la capacidad de operar sin conexión a internet y en segundo plano, además de notificar al usuario únicamente cuando se requiere su atención. No obstante, estas ventajas vienen acompañadas de un coste significativo: no se puede reutilizar el código entre distintas plataformas, lo que resulta en un aumento de la carga de trabajo y en gastos elevados relacionados con el desarrollo, actualización y distribución de nuevas versiones [30].

Según Ramírez, citado por [31], concluye que para desarrollar aplicaciones nativas es necesario utilizar un entorno específico para la plataforma en la cual se va a desplegar la aplicación, ya que estos proporcionan todas las herramientas que son necesarias para la construcción y depuración de aplicaciones.

### **1.3.18 Aplicación web**

Las denominadas aplicaciones web, están diseñadas para ejecutarse en navegadores desarrolladas utilizando tecnologías web conocidas (HTML, CSS y JavaScript), no requieren adecuarse a ningún entorno operativo además la implementación es rápida y fácil [30].

Según Tracy, K.W., citado por [30], las desventajas de las aplicaciones web móviles es su rendimiento. Los tiempos de respuesta se ven afectados por la interacción cliente-servidor y las restricciones de seguridad impuestas a la ejecución de código por los navegadores que limitan el acceso a todas las funciones del dispositivo. Además, al poseer un restrictivo estilo visual debido a las características físicas de los dispositivos móviles, resultan poco atractivas para el uso de los usuarios.

### **1.3.19 Aplicación híbrida**

Las aplicaciones híbridas constituyen otra evolución multiplataforma fundamentada en tecnologías web (HTML, Javascript y CSS) pero a diferencia de las aplicaciones anteriores, no son ejecutadas por el navegador. En cambio, se ejecutan en un contenedor web dedicado que brinda acceso adicional a la funcionalidad del dispositivo a través de una API particular [30].

Según Delia L., citado por [30] concluye que las aplicaciones híbridas permiten la reutilización de código entre plataformas, el acceso al hardware del dispositivo, y la distribución a través de tiendas de aplicaciones. Sin embargo, conservan algunas de las desventajas de las aplicaciones Web Móviles: el uso de componentes no nativos en la interfaz afecta la experiencia de usuario, y la carga asociada con el contenedor web ralentiza el rendimiento.

### **1.3.20 Ley del inquilinato**

En base a las leyes de inquilinato independientemente del país, ciudad o sector donde se encuentre el inmueble a ser rentado, el arrendador y arrendatario deberán sujetarse a normas establecidas por la autoridad designada para su funcionamiento legal. Se



establecen estas normas debido a que se pretende brindar una protección a las personas las cuales cuentan con pocas posibilidades económicas para poseer una vivienda propia [32].

En Ecuador la ley del inquilinato se aprobó a través del Congreso Nacional, dicha ley se ha mantenido hasta la actualidad como un instrumento por el cual los interesados, tanto arrendatario como arrendador poseen sus derechos y obligaciones debidamente marcados. El ejercicio y aplicación de esta normativa está debidamente avalado para quienes ofrezcan alquiler de viviendas o locales comerciales [33].

Los temas más relevantes los cuales se consideran como primordiales ante el alquiler o arriendo de inmuebles son considerados los servicios básicos (agua potable, luz eléctrica, internet, etc.) mucho dependerá de la instalación la cual se esté prestando para su uso, el que el arrendatario cumpla con estos requisitos antes de generar un contrato debidamente estructurado y legal [33].

Es posible otorgar relevancia a ciertos aspectos de la ley propuesta (Ver Anexo 1).

### **1.3.21 Tecnología publicitaria**

La tecnología publicitaria se caracteriza por su continua innovación y búsqueda de nuevas fórmulas para captar la atención y generar mayor expectación [34].

Dentro de la publicidad digital se pueden englobar aquellas técnicas las cuales emplean las empresas o marcas de productos o servicios para lo cual utilizan plataformas y herramientas digitales como medio de difusión, es decir se apoyan en el internet para alcanzar a un mayor grupo de consumidores con necesidades comunes [35].

La tecnología publicitaria forma parte importante para administrar una campaña publicitaria digital. Según Lavandeira Sonia, de la Universidad de la Coruña, en su publicación denominada publicidad digital afirma que la publicidad digital es que los consumidores pueden interactuar dando opiniones de los productos en foros, blogs, o

en páginas de consumidores como: Ciao, Trnd o Bopki. Las redes sociales como Facebook contienen anuncios y las diferentes marcas de productos crean grupos de seguidores [36].

Las campañas son iniciadas desde el anunciante en las que el usuario recibe una información en forma de Short Message Service (SMS) o Multimedia Messaging Service (MMS) y en la que se le incita a una acción concreta hacia una marca, producto o servicio. Estos mensajes pueden incluir desde ofertas, descuentos especiales o promociones a las que acceder con mensajes de respuesta, o bien invitar a navegar por una web móvil [37].

En el campo de la publicidad móvil, los tipos de formato más empleados son aquellos que se incorporan en plataformas de mensajería populares, como los sistemas de mensajes cortos (SMS) y mensajes multimedia (MMS). Aunque los mensajes multimedia aún no abarcan una amplia audiencia, su adopción está en constante aumento debido a su capacidad para integrar elementos como texto, imagen y sonido [37].

Los portales Wap mejoran la eficiencia de los contenidos al adaptar la información mediante un micro-navegador que presenta de manera óptima los datos de internet en pantallas de menor tamaño y con menor capacidad de memoria. Por lo general, la conexión a internet móvil se establece a través de los portales de entrada Wap predeterminados por la compañía telefónica, independientemente del tipo de portal que utilice el usuario. En todos los casos, el usuario se encontrará con formatos publicitarios integrados, flotantes o expandibles [37].

Una de las formas publicitarias más novedosas aplicadas a través del teléfono móvil, es el marketing de proximidad que basa su eficacia en la posibilidad de enviar mensajes a receptores que se encuentran en el área de interés del emisor. En este campo se están desarrollando interesantes iniciativas en el ámbito de la geolocalización a través de sistemas de triangulación Global System for Mobile communications (GSM) y Global

Positioning System (GPS) que abren interesantes campos de actuación, pero que en el ámbito de la publicidad aún se encuentran en fase experimental [37].

### **1.3.22 Difusión de arriendos universitarios**

Para lograr alcanzar a un grupo determinado de consumidores es necesario tomar en cuenta el ambiente en donde se pretende lanzar la técnica de difusión de algún producto o servicio. Se determinan técnicas de comunicación y alcance al público objetivo a través de estrategias juntamente con el marketing digital y técnicas inbound apoyadas en un amplio contenido [38].

Para el caso de los arriendos de viviendas o inmuebles, es necesario tomar en cuenta y poner en claro aspectos importantes para el cliente potencial en este caso, los cuales a breves términos pueden ser:

- Precio.
- Ubicación.
- Seguridad.
- Personalización.
- Variedad [38].

Las 8 P's del marketing de servicios son una valiosa metodología para identificar elementos cruciales en la oferta de un producto o servicio. A diferencia de las empresas que comercializan bienes, las cuales se centran en aspectos como producto, precio, lugar y promoción, cuando se trata de arrendamiento de inmuebles, para garantizar la satisfacción del cliente se requiere la inclusión de elementos suplementarios tales como entorno físico, proceso, personal y productividad. En este contexto, la utilización de las 8 P's del marketing de servicios facilita la formulación de estrategias destinadas a cubrir las necesidades de los clientes en un mercado de alta competencia [39].

## **1.4 Objetivos**

### **1.4.1 Objetivo general**

Desarrollar una aplicación móvil usando tecnologías multiplataforma para la difusión y búsqueda de arriendos universitarios en los alrededores de la Universidad Técnica de Ambato campus Huachi.

### **1.4.2 Objetivos específicos**

- Analizar las dificultades que conlleva el proceso de arrendar un departamento universitario tanto para el arrendatario como el arrendador.
- Generar los procesos involucrados para la difusión y búsqueda de arriendos universitarios.
- Seleccionar la tecnología multiplataforma para el desarrollo de la aplicación móvil.
- Desarrollar la aplicación móvil usando tecnologías multiplataforma para la difusión y búsqueda de arriendos universitarios.

## **CAPÍTULO II.- METODOLOGÍA**

### **2.1 Materiales**

El material utilizado para recopilar la información de la presente investigación se basó en una encuesta compuesta por 11 preguntas. Estas preguntas abarcaron una variedad de formatos, incluyendo opciones cerradas, selección múltiple y el empleo de la escala de Likert. Estos cuestionamientos fueron dirigidos tanto a los propietarios o arrendadores de los departamentos universitarios, como a los estudiantes o arrendatarios de la Facultad de Ingeniería en Sistemas Electrónica e Industrial.

### **ENCUESTA DIRIGIDA HACIA ARRENDADORES DE LOS DEPARTAMENTOS UNIVERSITARIOS**

**Objetivo:** Analizar las dificultades y necesidades de los arrendadores en la difusión de sus arriendos universitarios en los alrededores de la Universidad Técnica de Ambato (Campus Huachi).

**Marque su respuesta con una X**

**1. ¿Conoce usted de aplicaciones o sitios web donde se puede publicar información de arriendos?**

- Si
- No

**2. ¿A través de qué medio considera que la difusión de arriendos universitarios puede alcanzar a más arrendatarios? Puede elegir más de una opción**

- Redes sociales
- Aplicación de arriendos
- Carteles exteriores
- Radio
- Tv

**3. ¿Cuán complejo se torna para usted el uso de la tecnología?**

- Muy difícil
- Difícil
- Regular
- Fácil
- Muy fácil

**4. Establezca la importancia de dar a conocer los lineamientos básicos sobre los cuales se especificará el contrato de arriendo al arrendatario. Ejm (garantías, tiempo promedio de arrendamiento, reglas, etc.).**

- Nada importante
- De poca importancia
- Moderadamente importante
- Importante
- Muy importante

**5. ¿Usted estaría de acuerdo que sus arrendatarios aporten con una opinión acerca del ambiente que viven en su departamento?**

- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

**6. ¿Tiene alguna preferencia en cuanto al género del arrendatario?**

- Prefiero arrendatarios de género femenino
- Prefiero arrendatarios de género masculino
- No tengo preferencia de género

**7. ¿Cuál es el rango de precios que está dispuesto a ofrecer para el arriendo de una propiedad universitaria?**

- Menos de \$100
- Entre \$100 y \$120
- Entre \$120 y \$140
- Entre \$140 y \$160
- Más de \$160

**8. ¿Sus departamentos de arriendo son desalojados en tiempo de vacaciones por los estudiantes universitarios?**

- Si
- No

**9. ¿Usted estaría de acuerdo que sus posibles arrendatarios accedan a la ubicación exacta de su departamento de arriendo?**

- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

**10. ¿Cuán importante considera usted que su publicidad de arrendamiento se encuentre en una plataforma tecnológica (internet).**

- Nada importante
- De poca importancia
- Moderadamente importante
- Importante
- Muy importante

**11. ¿Cuál es el obstáculo que usted encuentra para incluir a su departamento de arriendos en un sitio o aplicación de publicidad? Puede elegir más de una opción.**

- Desconocimiento de la tecnología
- Falta de acceso a internet
- No poseer un teléfono inteligente
- Gastos por elaboración y mantenimiento de la plataforma

***¡Gracias por su colaboración!***

Utilizando la plataforma Google Forms, se diseñó una encuesta que constó de 11 preguntas dirigidas a los estudiantes o arrendatarios. Estas preguntas que abarcaron una variedad de formatos, incluyendo opciones cerradas, selección múltiple y el empleo de la escala de Likert.

### **ENCUESTA USUARIOS/ARRENDATARIOS**

**Objetivo:** Obtener datos relevantes acerca de las problemáticas que conllevan los estudiantes universitarios en el proceso de búsqueda de departamentos de arriendo, en los alrededores de la Universidad Técnica de Ambato (Campus Huachi).

**1. ¿Conoce usted sobre alguna aplicación móvil para adquirir un departamento de arriendo?**

- Si
- No

**2. ¿Con qué frecuencia busca arrendar un departamento universitario?**

- Nunca
- Casi nunca
- Ocasionalmente
- Usualmente



Siempre

**3. ¿A través de qué medio suele obtener información de arriendos universitarios?**

**Puede elegir más de una opción.**

Redes sociales

Aplicación de arriendos

Carteles exteriores

Radio

Tv

**4. ¿Cuán importante considera usted la ubicación del sitio de arriendo con respecto a la Universidad Técnica de Ambato?**

Nada importante

De poca importancia

Moderadamente importante

Importante

Muy importante

**5. ¿Cuáles son los factores que usted considera importantes el momento de arrendar? Puede elegir más de una opción.**

Ubicación

Precio

Seguridad

Servicios incluidos (luz, agua, internet, etc.)

Accesibilidad (transporte público, estacionamiento, etc.)

**6. ¿Considera que el desplazamiento para la búsqueda de arriendo es una problemática?**

Si

No

**7. ¿Cuán importante considera usted conocer los detalles del espacio que se oferta (habitaciones, baños independientes, espacio de lavado, etc.) previo a su ingreso?**

- Nada importante
- De poca importancia
- Moderadamente importante
- Importante
- Muy importante

**8. ¿Cuál es el rango de precio que estaría dispuesto/a a pagar por un departamento de arriendo para estudiantes universitarios?**

- Menos de \$100
- Entre \$100 y \$120
- Entre \$120 y \$140
- Entre \$140 y \$160
- Más de \$160

**9. ¿Cuáles han sido las principales dificultades que ha encontrado en el proceso de adquirir un departamento? Puede elegir más de una opción.**

- Precios elevados
- Falta de seguridad
- Exigencias y requisitos excesivos de los arrendadores
- Falta de información clara y transparente
- Problemas con los arrendadores o propietarios

**10. ¿Le resultaría útil una aplicación móvil para la búsqueda de arriendos universitarios, características, ubicación, etc?**

- Si
- No

**11. ¿Considera importante la visualización del contrato de arriendo previo al contacto con el arrendador?**

- Nada importante
- De poca importancia
- Moderadamente importante
- Importante
- Muy importante

Los cuestionarios fueron realizados con la finalidad de recopilar información de los arrendadores y arrendatarios sobre sus necesidades, preferencias, capacidad financiera y su relación con las TIC (Tecnologías de la Información y Comunicación).

## **2.2 Métodos**

### **2.2.1 Modalidad de la investigación**

La presente investigación se contextualizó en la modalidad de investigación de campo, documental-bibliográfica y modalidades especiales.

#### **Investigación de campo**

Se aplicó esta modalidad en el sentido de recopilar información sobre los arriendos universitarios, siendo este el objeto de estudio, del cual se obtuvieron datos relevantes que sustentaron el desarrollo del trabajo investigativo.

#### **Investigación bibliográfica – documental**

En este caso este tipo de investigación se ajusta a proyecto, puesto que se requiere apoyo de revistas, artículos científicos, tesis, libros los cuales sirvieron como justificación para apoyar el uso de tecnologías multiplataforma, como herramientas para el desarrollo de aplicaciones móviles y las metodologías para su implementación eficiente.

## Investigación de modalidades especiales

Esta modalidad se involucra a la investigación por tratarse el presente proyecto enfocado en desarrollo de software de una aplicación móvil innovadora en el ámbito de los arriendos universitarios, para solventar un problema social en específico.

### 2.2.2 Población y muestra

La población para el presente proyecto consta de dos interesados considerando a los arrendadores y arrendatarios o inquilinos. Para el caso de los arrendadores la población establecida consta de 30 inmuebles que se encuentran ubicados en los alrededores de la Universidad Técnica de Ambato (Campus Huachi). Además, para el caso de arrendatarios la población se define mediante una muestra dirigida a los estudiantes de la Facultad de Sistemas Electrónica e Industrial que son provenientes de provincias.

#### Población de arrendadores:

Población	Número	Porcentaje
Arrendadores	30	100%
<b>Total</b>	<b>30</b>	<b>100%</b>

**Tabla 3.** Población arrendadores.

**Elaborado por:** La investigadora.

Dado que la población a estudiar no supera los 100 elementos no se trabajó con la muestra representativa y se trabajó con la totalidad de la población.

**Población de arrendatarios:** para la población de arrendatarios se basó en los estudiantes de la Facultad de Sistemas Electrónica e Industrial que consta de una población de 1840 estudiantes según las estadísticas realizada por la UTA en el periodo académico abril - septiembre 2021 (Ver Anexo 2), aplicando a la misma la técnica de muestreo no probabilístico.

Fórmula:

$$n = \frac{N(Z)^2PQ}{Z^2PQ + Ne^2}$$

<b>Simbología</b>	<b>Significado</b>	<b>Representación</b>
N	Población	1840
e	Error estándar	0.09
Z	Nivel de confianza del 95%	1.96
n	Tamaño de la muestra	111
P	Probabilidad de éxito	0.5
Q	Probabilidad de fracaso	0.5

**Tabla 4.** Simbología de la fórmula de muestra.

**Elaborado por:** La investigadora.

$$n = \frac{1840(1.96)^2(0.5)(0.5)}{(1.96)^2(0.5)(0.5) + 1840(0.09)^2}$$

$$n = 111.390$$

$$n = 111 \text{ arrendatarios}$$

### **2.2.3 Recolección de la información**

Se recolectó información a través de encuestas aplicadas tanto a arrendadores de los departamentos universitarios y a la muestra de estudiantes de la Facultad de Sistemas, Electrónica e Industrial como arrendatarios, para analizar las dificultades que presenta el proceso de la difusión y búsqueda de arriendos universitarios.

### **Validación del instrumento**

#### **Kuder-Richardson en la encuesta de arrendadores**

La confiabilidad y validez son elementos importantes en la recolección de los datos. Para evaluar la consistencia interna de los datos del conjunto de preguntas elaboradas

en el cuestionario, se aplicó el coeficiente de Kuder-Richardson, aplicado a 2 de las 11 preguntas dirigidas a los arrendadores, las cuales se identifican como dicotómicas es decir que ofrecen dos opciones de respuesta tanto si como no, con el fin de obtener información rápida y clara que aporte al desarrollo de la investigación.

Para el cálculo de Kuder-Richardson se aplicó la siguiente fórmula:

$$r_{kr20} = \left( \frac{k}{k-1} \right) \left( 1 - \frac{\sum pq}{\sigma^2} \right)$$

Los cálculos fueron realizados en Excel para calcular fácilmente la fórmula a partir de la creación de una tabla adjuntando las respuestas recolectadas en las encuestas que se realizaron anteriormente (Ver Anexo 3).

<b>Simbología</b>	<b>Valor</b>
K (Número de ítems) =	2
P (Porcentaje de personas que responden si cada ítem) =	0,33 y 0,13
q (Porcentaje de personas que responden no cada ítem) =	0,67 y 0,87
$\sigma^2$ (Varianza total del instrumento) =	0,55
KR-20 =	0,78

**Tabla 5.** Confiabilidad Kuder-Richardson en la encuesta para arrendadores.

**Elaborado por:** La investigadora.

<b>KR-20</b>	<b>Interpretación</b>
0,9 - 1	Excelente
0,8 - 0,9	Buena
0,7 - 0,8	Aceptable
0,6 - 0,7	Débil
0,5 - 0,6	Pobre
< 0,5	Inaceptable

**Tabla 6.** Interpretación de la escala de consistencia de Kuder-Richardson.

**Elaborado por:** La investigadora.

Como resultado Kuder-Richardson da un valor de 0,78 considerando la confiabilidad de las respuestas de los arrendadores dentro del rango aceptable.

**Alfa de Cronbach en la encuesta de arrendadores.**

Para validar las preguntas restantes que han sido elaboradas en escala de Likert se aplicó otra de las medidas de confiabilidad conocida como alfa de Cronbach.

Para el cálculo de alfa de Cronbach se aplicó la siguiente fórmula:

$$\alpha = \left( \frac{k}{k - 1} \right) \left( 1 - \frac{\sum v_i}{vt} \right)$$

Los cálculos de la fórmula fueron realizados en Excel a partir de la creación de una tabla, adjuntando las respuestas recolectadas en las encuestas que se realizaron anteriormente (Ver Anexo 4).

<b>Simbología</b>	<b>Valor</b>
$\alpha$ (alfa) =	0,808
K (Número de ítems) =	5
$\sum v_i$ (Varianza de cada ítem) =	1,658
$vt$ (Varianza total) =	4,693

**Tabla 7.** Confiabilidad alfa de Cronbach en la encuesta para arrendadores.

**Elaborado por:** La investigadora.

<b>Alfa de Cronbach</b>	<b>Interpretación</b>
0,9	Excelente
0,9 - 0,8	Buena
0,8 - 0,7	Aceptable
0,7 - 0,6	Débil
0,6 - 0,5	Pobre
< 0,5	Inaceptable

**Tabla 8.** Interpretación de la escala de consistencia del alfa de Cronbach.

**Elaborado por:** La investigadora.

Como resultado el alfa de Cronbach da un valor de 0,808 considerando la confiabilidad de las respuestas de los arrendadores dentro del rango aceptable.

**Kuder-Richardson en la encuesta de arrendatarios.**

En primera instancia se aplica el coeficiente Kuder-Richardson a los datos obtenidos de las preguntas 1, 6 y 10 respectivamente, las cuales fueron dirigidas a los arrendatarios, pero se obtiene un valor de confiabilidad menor al aceptable, por lo cual se tomó la decisión de retirar la pregunta 6 debido a que no posee una correlación positiva. Obteniendo así una confiabilidad de las respuestas aceptable con 0,80 entre la pregunta 1 y 10.

Los cálculos fueron realizados en Excel para calcular fácilmente la fórmula a partir de la creación de una tabla adjuntando las respuestas recolectadas en las encuestas que se realizaron anteriormente (Ver Anexo 5).

<b>Simbología</b>	<b>Valor</b>
K (Número de ítems) =	2
P (Porcentaje de personas que responden si cada ítem) =	0,17 y 0,93
q (Porcentaje de personas que responden no cada ítem) =	0,83 y 0,07
$\sigma^2$ (Varianza total del instrumento) =	0,23
KR-20	0,80

**Tabla 9.** Confiabilidad Kuder-Richardson en la encuesta para arrendatarios.

**Elaborado por:** La investigadora.

Como resultado Kuder-Richardson da un valor de 0,80 considerando la confiabilidad de las respuestas de los arrendatarios dentro del rango aceptable.

**Alfa de Cronbach en la encuesta de arrendatarios.**

Se evalúan las preguntas 4 y 7 con el alfa de Cronbach que previamente han sido diseñadas en la escala de Likert. Los cálculos de la fórmula fueron realizados en Excel



Los cálculos de la fórmula fueron realizados en Excel a partir de la creación de una tabla, adjuntando las respuestas recolectadas en las encuestas que se realizaron anteriormente (Ver Anexo 6).

<b>Simbología</b>	<b>Valor</b>
$\alpha$ (alfa) =	0,821
K (Número de ítems) =	2
$\sum v_i$ (Varianza de cada ítem) =	1,084
$vt$ (Varianza total) =	1,84

**Tabla 10.** Confiabilidad alfa de Cronbach en la encuesta para arrendatarios.

**Elaborado por:** La investigadora.

Como se muestra en la **Tabla 10**, el resultado en el alfa de Cronbach da un valor de 0,821, considerando la confiabilidad de las respuestas dentro del rango aceptable.

### **Resultados de la encuesta aplicada a los arrendadores**

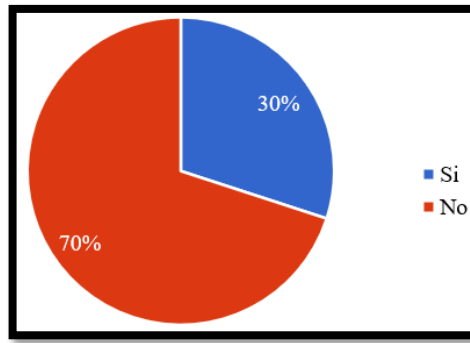
La encuesta fue realizada presencialmente a los arrendadores de los departamentos universitarios en los alrededores de la UTA.

**Pregunta 1:** ¿Conoce usted de aplicaciones o sitios web donde se puede publicar información de arriendos?

<b>Indicador</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Si	21	70%
No	9	30%
<b>Total</b>	30	100%

**Tabla 11.** Resultados pregunta 1 (arrendadores).

**Elaborado por:** La investigadora.



**Figura 4.** Resultados pregunta 1 (arrendadores).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

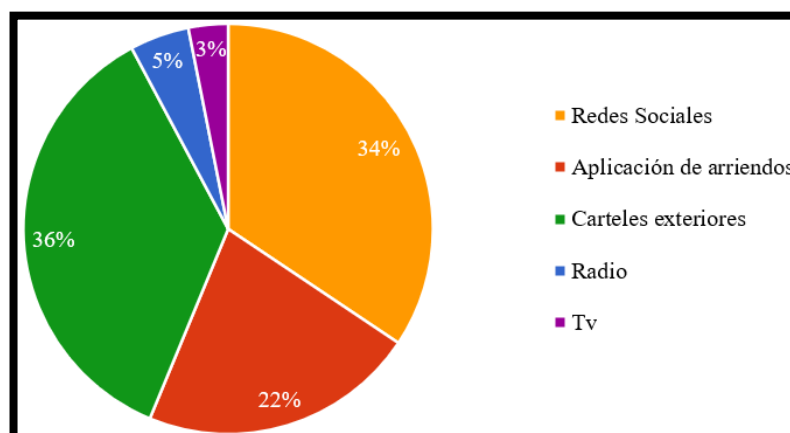
Con los resultados obtenidos en la **Figura 4**, se observa que el 70% de los arrendadores desconocen de aplicaciones o sitios web donde se pueden publicar información de arriendos, mientras que el 30% tienen conocimiento de herramientas tecnológicas. Como resultado **se evidencia el desconocimiento de aplicaciones y sitios web por parte de los arrendadores, ampliando oportunamente el mercado para que las herramientas tecnológicas puedan favorecer al control de sus arriendos.**

**Pregunta 2:** ¿A través de qué medio considera que la difusión de arriendos universitarios puede alcanzar a más arrendatarios? Puede elegir más de una opción.

<b>Indicador</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Redes sociales	22	34%
Aplicación de arriendos	14	22%
Carteles exteriores	23	36%
Radio	3	5%
Tv	2	3%
<b>Total</b>	<b>30</b>	<b>100%</b>

**Tabla 12.** Resultados pregunta 2 (arrendadores).

**Elaborado por:** La investigadora.



**Figura 5.** Resultados pregunta 2 (arrendadores).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

Los resultados de la pregunta formulada mediante opción múltiple se han analizado en la **Figura 5**, donde se observa que el 36% de los arrendadores considera que el medio de difusión de arriendos universitarios que mayor alcance tiene es a través de carteles exteriores, el 34% a través de redes sociales, el 22% a través de aplicaciones móviles, el 5% a través de Tv y finalmente el 3% a través de radio. Como resultado se evidencia que **los carteles exteriores son el medio de difusión por el que mayormente se inclinan los arrendadores, desestimando medios como Tv y radio, los cuales requieren de mayor inversión.**

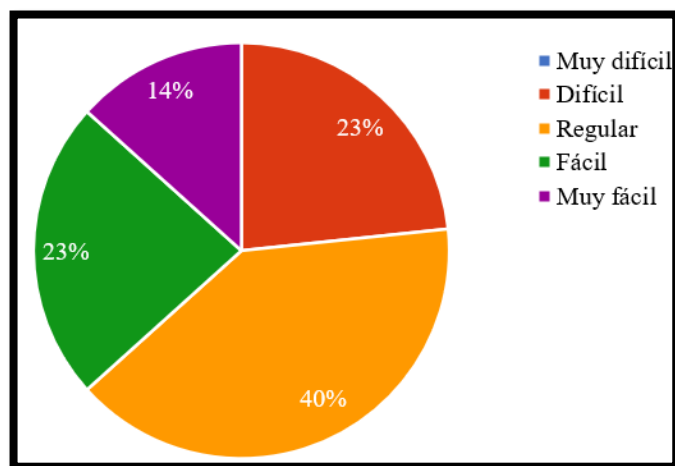
**Pregunta 3:** ¿Cuán complejo se torna para usted el uso de la tecnología?

<b>Indicador</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Muy difícil	0	0%
Difícil	7	23%
Regular	12	40%
Fácil	7	23%

Muy fácil	4	14%
<b>Total</b>	30	100%

**Tabla 13.** Resultados pregunta 3 (arrendadores).

**Elaborado por:** La investigadora.



**Figura 6.** Resultados pregunta 3 (arrendadores).

**Elaborado por:** La investigadora.

### Análisis e interpretación de resultados

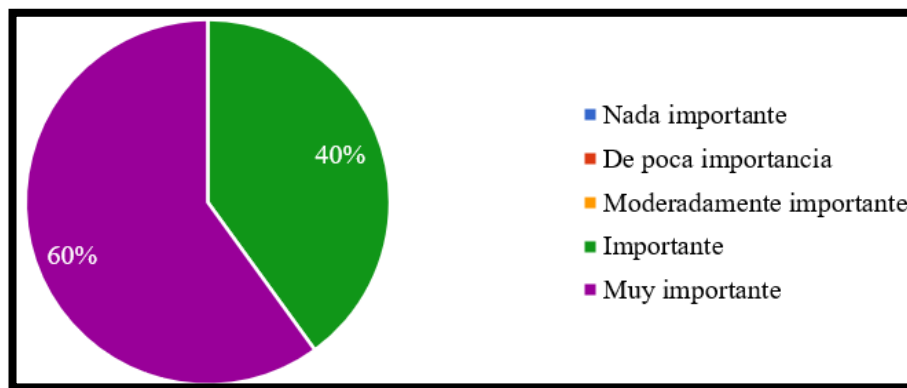
Con los resultados obtenidos en la **Figura 6**, al 40% de los arrendadores se les torna regular la complejidad el uso de la tecnología, al 23.3% difícil, al otro 23.3.% fácil, mientras que al 14% muy fácil. Como resultado **el nivel de dificultad del uso de la tecnología por parte de los arrendadores se encuentra dentro de un rango aceptable para el uso de la aplicación móvil.**

**Pregunta 4:** Establezca la importancia de dar a conocer los lineamientos básicos sobre los cuales se especificará el contrato de arriendo al arrendatario. Ejemplo: (garantías, tiempo promedio de arrendamiento, reglas).

Indicador	Frecuencia	Porcentaje
Nada importante	0	0%
De poca importancia	0	0%
Moderadamente importante	0	0%
Importante	12	40%
Muy importante	18	60%
<b>Total</b>	<b>30</b>	<b>100%</b>

**Tabla 14.** Resultados pregunta 4 (arrendadores)

**Elaborado por:** La investigadora



**Figura 7.** Resultados pregunta 4 (arrendadores).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

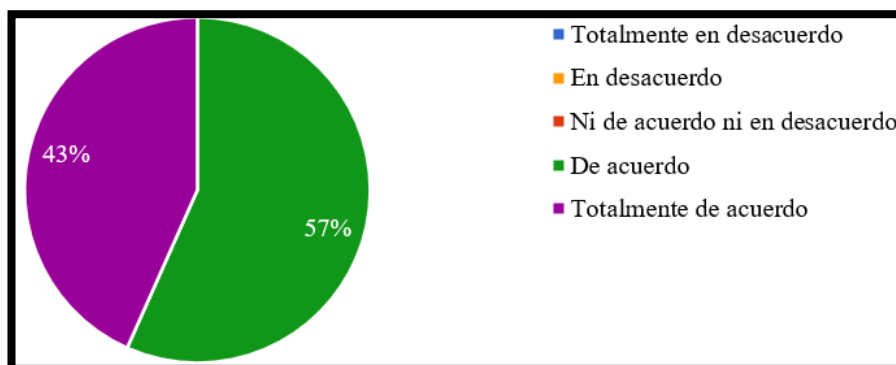
Con los resultados obtenidos en la **Figura 7**, **al 60% de los arrendadores consideran muy importante dar a conocer los lineamientos básicos sobre los cuales se especificará el contrato de arriendo al arrendatario**, mientras que el 40% lo consideran importante. Como resultado se evidencia que los arrendadores consideran muy importante dar a conocer claramente en su contrato de arriendo los derechos y obligaciones a los arrendatarios.

**Pregunta 5:** ¿Usted estaría de acuerdo que sus arrendatarios aporten con una opinión acerca del ambiente que viven en su departamento?

Indicador	Frecuencia	Porcentaje
Totalmente en desacuerdo	0	0%
En desacuerdo	0	0%
Ni de acuerdo ni en desacuerdo	0	0%
De acuerdo	17	57%
Totalmente de acuerdo	13	43%
<b>Total</b>	30	100%

**Tabla 15.** Resultados pregunta 5 (arrendadores).

**Elaborado por:** La investigadora.



**Figura 8.** Resultados pregunta 5 (arrendadores).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

Con los resultados obtenidos en la **Figura 8**, el 57% de los arrendadores están de acuerdo que sus arrendatarios aporten con una opinión acerca del ambiente que viven en el departamento arrendado, mientras que el 43% está totalmente de acuerdo. En conclusión, **los arrendadores están de acuerdo en recibir opiniones sobre la conformidad de los arrendatarios con respecto al cuarto de arriendo**, valorando

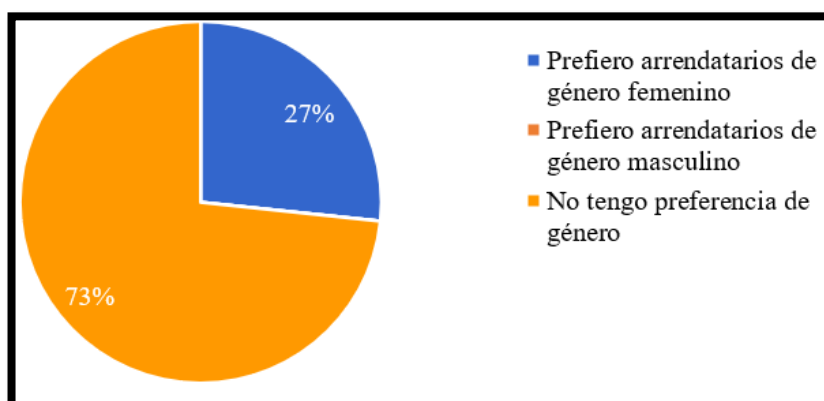
la retroalimentación de estos y la disposición a mejorar la calidad de vida en sus propiedades.

**Pregunta 6:** ¿Tiene alguna preferencia en cuanto al género del arrendatario?

<b>Indicador</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Prefiero arrendatarios de género femenino	8	27%
Prefiero arrendatarios de género masculino	0	0%
No tengo preferencia de género	22	73%
<b>Total</b>	<b>30</b>	<b>100%</b>

**Tabla 16.** Resultados pregunta 6 (arrendadores).

**Elaborado por:** La investigadora.



**Figura 9.** Resultados pregunta 6 (arrendadores).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

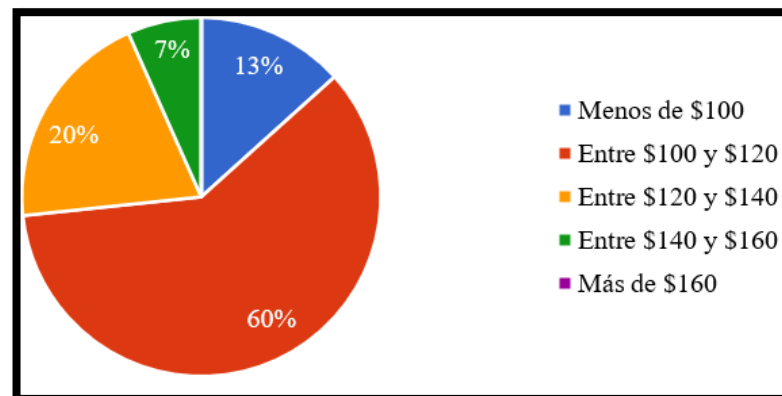
Con los resultados obtenidos en la **Figura 9**, el 73% de los arrendadores no tienen preferencia de género el momento de arrendar sus habitaciones a estudiantes universitarios, mientras que el 27% prefieren arrendatarios de género femenino. Por lo tanto, **la mayoría de los arrendadores proporcionan igualdad de oportunidades el momento de ofrecer cuartos de arriendos a los estudiantes universitarios.**

**Pregunta 7:** ¿Cuál es el rango de precios que estás dispuesto a ofrecer para el arriendo de una propiedad universitaria?

Indicador	Frecuencia	Porcentaje
Menos de \$100	4	13%
Entre \$100 y \$120	18	60%
Entre \$120 y \$140	6	20%
Entre \$140 y \$160	2	7%
Más de \$160	0	0%
<b>Total</b>	<b>30</b>	<b>100%</b>

**Tabla 17.** Resultados pregunta 7 (arrendadores).

**Elaborado por:** La investigadora.



**Figura 10.** Resultados pregunta 7 (arrendadores).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

Con los resultados obtenidos en la **Figura 10**, el 60% de arrendatarios ofrecen a los estudiantes precios entre \$100 y \$120 para arriendos universitarios, el 20% entre \$120 y \$140, el 13% menos de \$100 y finalmente el 7% entre \$140 \$160. Por lo tanto, **la mayoría de los arrendadores ofrecen arriendos con precios razonables y accesibles para los estudiantes universitarios.**

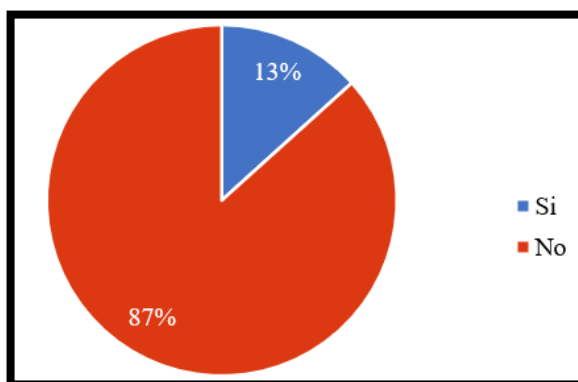


**Pregunta 8:** ¿Sus departamentos de arriendo son desalojados en tiempo de vacaciones por los estudiantes universitarios?

Indicador	Frecuencia	Porcentaje
Si	4	13%
No	26	87%
<b>Total</b>	30	100%

**Tabla 18.** Resultados pregunta 8 (arrendadores).

**Elaborado por:** La investigadora.



**Figura 11.** Resultados pregunta 8 (arrendadores).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

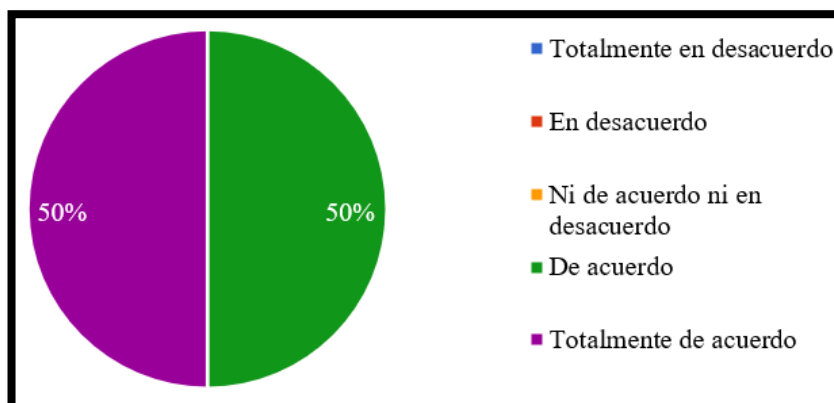
Con los resultados obtenidos en la **Figura 11**, el 87% de los departamentos no son desalojados en tiempo de vacaciones por los estudiantes universitarios, mientras que el 13% si son desalojados. Por lo tanto, la mayoría de los departamentos de arriendos no siempre cumplen con el plazo estipulado de dos años como se establece en la ley del inquilinato, ya que **permanecen ocupados por las pertenencias de los estudiantes durante las vacaciones hasta empezar el próximo periodo académico** para retomar clases, esto determina un bajo nivel de disponibilidad de arriendos para posibles nuevos arrendatarios.

**Pregunta 9:** ¿Usted estaría de acuerdo que sus posibles arrendatarios accedan a la ubicación exacta de su departamento de arriendo?

Indicador	Frecuencia	Porcentaje
Totalmente en desacuerdo	0	0%
En desacuerdo	0	0%
Ni de acuerdo ni en desacuerdo	0	0%
De acuerdo	15	50%
Totalmente de acuerdo	15	50%
<b>Total</b>	<b>30</b>	<b>100%</b>

**Tabla 19.** Resultados pregunta 9 (arrendadores).

**Elaborado por:** La investigadora.



**Figura 12.** Resultados pregunta 9 (arrendadores).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

Con los resultados obtenidos en la **Figura 12**, el 50% de arrendadores están totalmente de acuerdo que sus posibles arrendatarios accedan a la ubicación exacta de su departamento de arriendo, mientras que el otro 50% está de acuerdo. Por lo tanto, **los arrendadores están dispuestos a compartir información relevante sobre su**

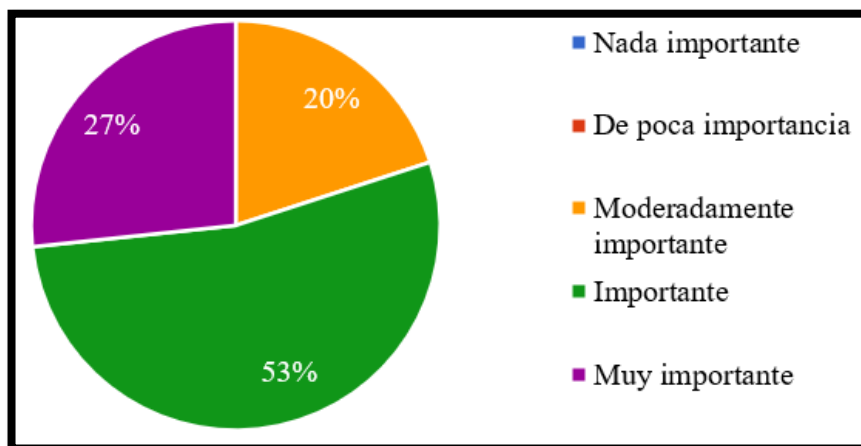
propiedad, proporcionando transparencia y confianza a los posibles arrendatarios.

**Pregunta 10:** ¿Cuán importante considera usted que su publicidad de arrendamiento se encuentre en una plataforma tecnológica (internet).

<b>Indicador</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Nada importante	0	0%
De poca importancia	0	0%
Moderadamente importante	6	20%
Importante	16	53%
Muy importante	8	27%
<b>Total</b>	<b>30</b>	<b>100%</b>

**Tabla 20.** Resultados pregunta 10 (arrendadores).

**Elaborado por:** La investigadora.



**Figura 13.** Resultados pregunta 10 (arrendadores).

**Elaborado por:** La investigadora.

## Análisis e interpretación de resultados

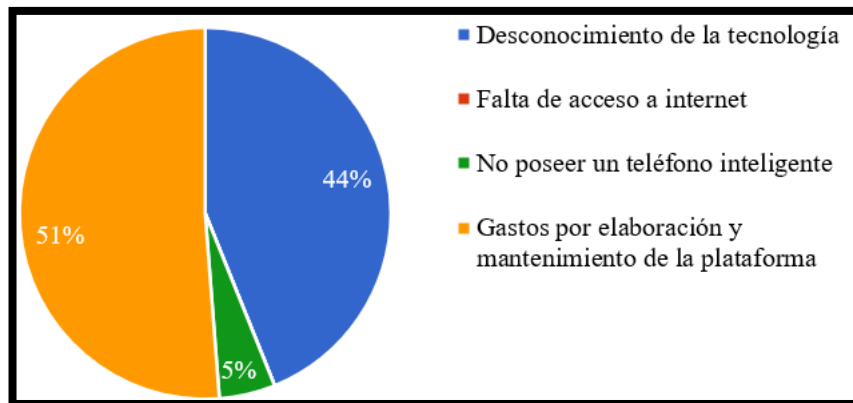
Con los resultados obtenidos en la **Figura 13**, el 53% de los arrendadores considera importante que su publicidad de arrendamiento se encuentre en una plataforma tecnológica, el 27% lo considera muy importante y finalmente el 20% moderadamente importante. Por lo tanto, **los arrendadores reconocen la importancia de la tecnología en el mercado de arriendos y están dispuestos a adaptarse a las tendencias actuales.**

**Pregunta 11:** ¿Cuál es el obstáculo que usted encuentra para incluir a su departamento de arriendos en un sitio o aplicación de publicidad? Puede elegir más de una opción.

<b>Indicador</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Desconocimiento de la tecnología	18	44%
Falta de acceso a internet	0	0%
No poseer un teléfono inteligente	2	5%
Gastos por elaboración y mantenimiento de la plataforma	21	51%
<b>Total</b>	30	100%

**Tabla 21.** Resultados pregunta 11 (arrendadores).

**Elaborado por:** La investigadora.



**Figura 14.** Resultados pregunta 11 (arrendadores).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

Los resultados de la pregunta formulada mediante opción múltiple se han analizado en la **Figura 14**, donde el 51% de los arrendadores no incluyen aún a su departamento de arriendos en un sitio o aplicación de publicidad porque encuentran como obstáculo los gastos por la elaboración y mantenimiento de una plataforma, el 44% debido al desconocimiento de la tecnología, y finalmente el 5% debido a que no poseen un teléfono inteligente. Por lo tanto, **los arrendadores no están seguros de que mediante el gasto que realicen en la plataforma sus arriendos atraigan a suficientes arrendatarios.**

### **Resultados de la encuesta aplicada a los estudiantes**

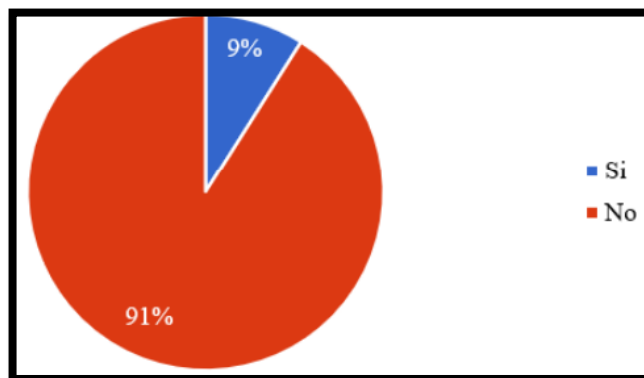
Los estudiantes pertenecen a la Facultad de Sistemas, Electrónica e Industrial como arrendatarios de los departamentos universitarios en los alrededores de la UTA, utilizando formularios de Google Forms debido a su facilidad de alcance.

**Pregunta 1:** ¿Conoce usted sobre alguna aplicación móvil para adquirir un departamento de arriendo?

Indicador	Frecuencia	Porcentaje
Si	10	10%
No	101	91%
<b>Total</b>	<b>111</b>	<b>100%</b>

**Tabla 22.** Resultados pregunta 1 (arrendatarios).

**Elaborado por:** La investigadora.



**Figura 15.** Resultados pregunta 1 (arrendatarios).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

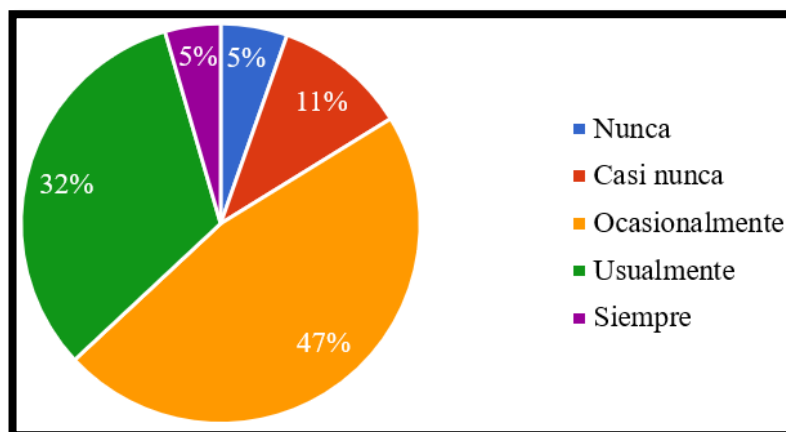
Con referencia a la **Figura 15**, el 91% de los encuestados no conoce acerca de alguna aplicación móvil para acceder a un arriendo, el 9% por otro lado si hace referencia al conocimiento de dichas aplicaciones. En consecuencia, **es viable el desarrollo de una aplicación móvil destinada a la búsqueda y difusión de cuartos o departamentos de arriendo.**

**Pregunta 2:** ¿Con qué frecuencia busca arrendar un departamento universitario?

Indicador	Frecuencia	Porcentaje
Nunca	6	5%
Casi nunca	12	11%
Ocasionalmente	52	47%
Usualmente	36	32%
Siempre	5	5%
<b>Total</b>	<b>111</b>	<b>100%</b>

**Tabla 23.** Resultados pregunta 2 (arrendatarios).

**Elaborado por:** La investigadora.



**Figura 16.** Resultados pregunta 2 (arrendatarios).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

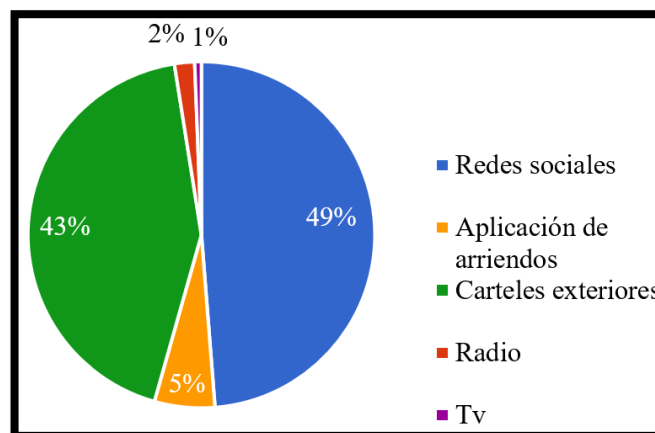
Con referencia a la **Figura 16**, donde el 47% de los estudiantes encuestados han frecuentado buscar ocasionalmente el arriendo de un departamento universitario, por otro lado, el 32% menciona que usualmente lo ha realizado, así como el 11% casi nunca ha buscado arrendar departamentos universitarios y el 5% de personas lo ha realizado casi nunca o nunca. En conclusión, **existe una adecuada frecuencia con la que los estudiantes realizan búsquedas de departamentos de arriendo lo cual aporta significativamente a la investigación.**

**Pregunta 3:** ¿A través de qué medio suele obtener información de arriendos universitarios? Puede elegir más de una opción.

Indicador	Frecuencia	Porcentaje
Redes sociales	78	49%
Aplicación de arriendos	9	5%
Carteles exteriores	69	43%
Radio	3	2%
Tv	1	1%
<b>Total</b>	<b>111</b>	<b>100%</b>

**Tabla 24.** Resultados pregunta 3 (arrendatarios).

**Elaborado por:** La investigadora.



**Figura 17.** Resultados pregunta 3 (arrendatarios).

**Elaborado por:** La investigadora.

### Análisis e interpretación de resultados

Los resultados de la pregunta formulada mediante opción múltiple se han analizado en la **Figura 17**, donde el 49% de los encuestados utilizan redes sociales para la búsqueda de información de arriendos, el 43% lo realiza a través de carteles exteriores, mientras el 5% únicamente utiliza aplicaciones de arriendos, los medios tradicionales como



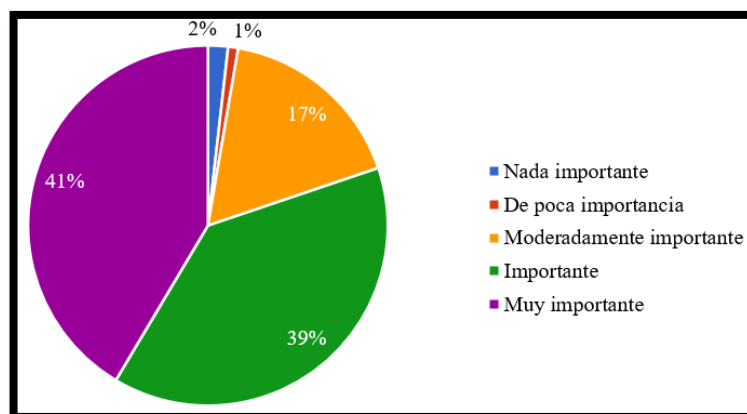
radio y Tv intervienen en un 2% y 1% respectivamente. Por lo tanto, **el uso de plataformas tecnológicas predomina en este caso, siendo la alternativa más accesible para conseguir información de arriendos.**

**Pregunta 4:** ¿Cuán importante considera usted la ubicación del sitio de arriendo con respecto a la Universidad Técnica de Ambato?

Indicador	Frecuencia	Porcentaje
Nada importante	2	2%
De poca importancia	1	1%
Moderadamente importante	19	17%
Importante	43	39%
Muy importante	46	41%
<b>Total</b>	<b>111</b>	<b>100%</b>

**Tabla 25.** Resultados pregunta 4 (arrendatarios).

**Elaborado por:** La investigadora.



**Figura 18.** Resultados pregunta 4 (arrendatarios).

**Elaborado por:** La investigadora.

## Análisis e interpretación de resultados

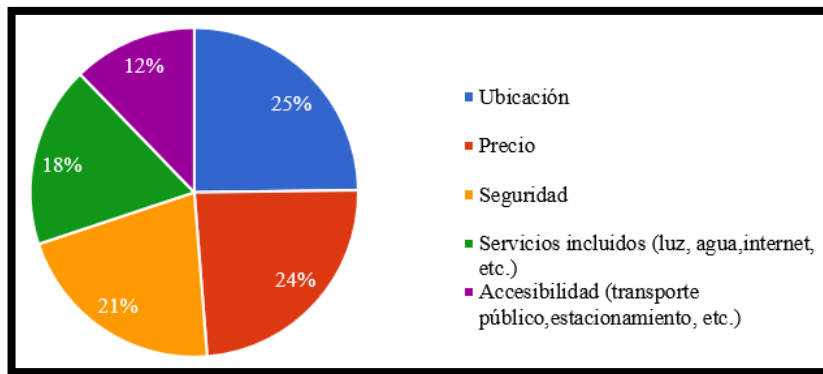
En base a los resultados representados en la **Figura 18**, el 41% de los estudiantes encuestados consideran muy importante la ubicación del sitio para arrendar con respecto a la Universidad, el 39% menciona lo importante que es la ubicación, por otro lado, el 17% considera moderadamente importante, el 1% afirma que es poco importante y el 2% nada importante. En consecuencia, desde la perspectiva de los estudiantes, **la ubicación, es una de las características a tomar en cuenta al momento de buscar un sitio para arrendar.**

**Pregunta 5:** ¿Cuáles son los factores que usted considera importantes el momento de arrendar? Puede elegir más de una opción.

<b>Indicador</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Ubicación	89	25%
Precio	86	24%
Seguridad	76	21%
Servicios incluidos (luz, agua, internet, etc.)	64	18%
Accesibilidad (transporte público, estacionamiento, etc.)	44	12%
<b>Total</b>	111	100%

**Tabla 26.** Resultados pregunta 5 (arrendatarios).

**Elaborado por:** La investigadora.



**Figura 19.** Resultados pregunta 5 (arrendatarios).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

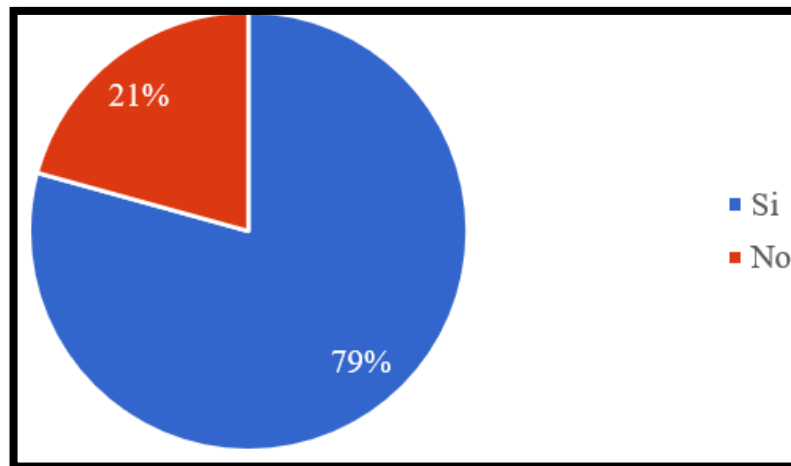
Los resultados de la pregunta formulada mediante opción múltiple se han analizado en la **Figura 19**, donde el 25% de los encuestados considera la ubicación como un factor predominante al momento de arrendar, seguidamente el 24% considera el precio, por otro lado, el 21% tiene a optar por la seguridad, mientras el 18% toma en cuenta servicios incluidos y el 12% accesibilidad. Por ende, **predomina la ubicación como el factor más considerado al momento de acceder a un arriendo por parte de los estudiantes.**

**Pregunta 6:** ¿Considera que el desplazamiento para la búsqueda de arriendo es una problemática?

<b>Indicador</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Si	88	79%
No	23	21%
<b>Total</b>	111	100%

**Tabla 27.** Resultados pregunta 6 (arrendatarios).

**Elaborado por:** La investigadora.



**Figura 20.** Resultados pregunta 6 (arrendatarios).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

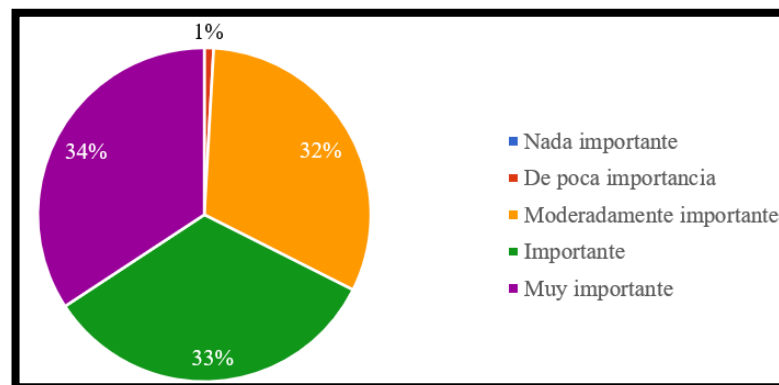
Según los resultados de la **Figura 20**, se muestra que el 79% de encuestados encuentran el desplazamiento para la búsqueda de arriendos como una problemática importante, por otro lado, el 21% contrarresta el planteamiento desestimando este factor. Por lo tanto, **se evidencia la presencia de una dificultad con respecto a la movilidad de la persona que busca el sitio para arrendar**, con lo cual a través del desarrollo de una funcionalidad en la aplicación móvil se puede solventar esta problemática.

**Pregunta 7:** ¿Cuán importante considera usted conocer los detalles del espacio que se oferta (habitaciones, baños independientes, espacio de lavado, etc.) previo a su ingreso?

Indicador	Frecuencia	Porcentaje
Nada importante	0	0%
De poca importancia	1	1%
Moderadamente importante	35	32%
Importante	37	33%
Muy importante	38	34%
<b>Total</b>	<b>111</b>	<b>100%</b>

**Tabla 28.** Resultados pregunta 7 (arrendatarios).

**Elaborado por:** La investigadora.



**Figura 21.** Resultados pregunta 7 (arrendatarios).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

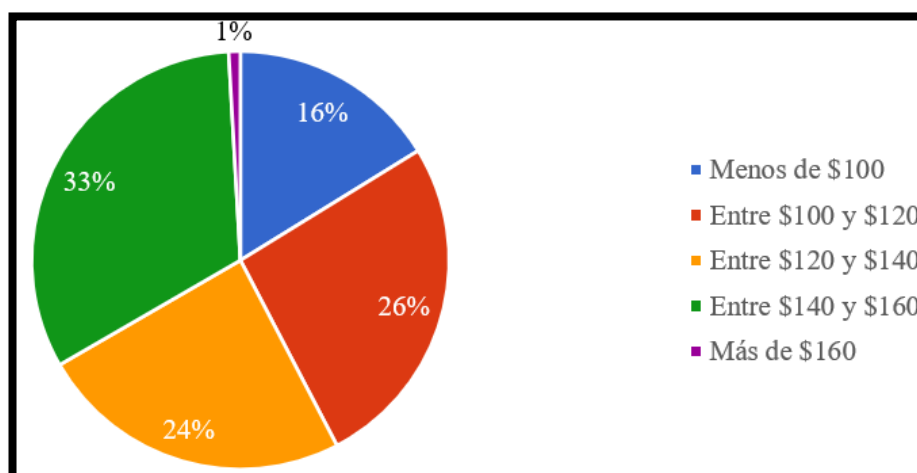
Con los resultados representados en la **Figura 21**, el 34% de estudiantes considera muy importante el detalle de la oferta de arriendo a la cual se pretende acceder, el 33% estima importante, mientras el 32% considera moderadamente importante, únicamente el 1% de los encuestados menciona de poca importancia. Como consecuencia, se puede inferir que **los detalles del espacio a arrendar son una característica muy valiosa para el arrendatario previo a acceder a concretar el arriendo.**

**Pregunta 8:** ¿Cuál es el rango de precio que estaría dispuesto/a a pagar por un departamento de arriendo para estudiantes universitarios?

Indicador	Frecuencia	Porcentaje
Menos de \$100	18	16%
Entre \$100 y \$120	29	26%
Entre \$120 y \$140	27	24%
Entre \$140 y \$160	36	33%
Más de \$160	1	1%
<b>Total</b>	<b>111</b>	<b>100%</b>

**Tabla 29.** Resultados pregunta 8 (arrendatarios).

**Elaborado por:** La investigadora.



**Figura 22.** Resultados pregunta 8 (arrendatarios).

**Elaborado por:** La investigadora.

### Análisis e interpretación de resultados

Según la **Figura 22**, el 33% de los encuestados están abarcados en presupuestos entre \$140 y \$160, mientras el 26% optan por valores entre \$100 y \$120, el 24% se encuentra en presupuestos entre \$120 y \$140, el 16% accede a valores menores a \$100 y solamente el 1% se encuentra en posibilidades de acceder a valores mayores a \$160. Por lo tanto, los presupuestos de los estudiantes ingresan en el rango de valores que

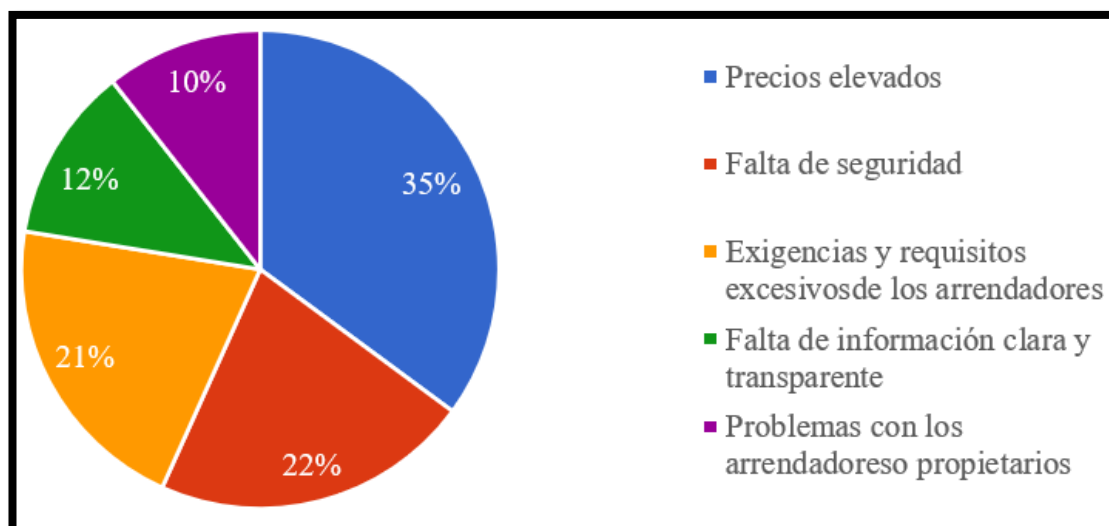
los arrendadores consideran más accesible para estudiantes, además **tienen a optar por rangos medios de precios con respecto a sus presupuestos con la finalidad de acceder a mejores sitios de arriendo.**

**Pregunta 9:** ¿Cuáles han sido las principales dificultades que ha encontrado en el proceso de adquirir un departamento? Puede elegir más de una opción.

Indicador	Frecuencia	Porcentaje
Precios elevados	86	35%
Falta de seguridad	53	22%
Exigencias y requisitos excesivos de los arrendadores	51	21%
Falta de información clara y transparente	29	12%
Problemas con los arrendadores propietarios	26	10%
<b>Total</b>	<b>111</b>	<b>100%</b>

**Tabla 30.** Resultados pregunta 9 (arrendatarios).

**Elaborado por:** La investigadora.



**Figura 23.** Resultados pregunta 9 (arrendatarios).

**Elaborado por:** La investigadora.

## Análisis e interpretación de resultados

Los resultados de la pregunta formulada mediante opción múltiple se han analizado en la **Figura 23**, donde el 35% de los estudiantes encuestados considera los precios elevados una dificultad al momento de acceder a un arriendo, el 22% la falta de seguridad, mientras el 21% las exigencias de los arrendadores, el 12% la falta de información clara y transparente por parte de los arrendadores y el 10% indica los problemas con los propietarios. En consecuencia, la mayor parte de los encuestados toma como prioridad los elevados costos con respecto al sitio que se pretende arrendar, además también **resaltan la seguridad que pueda existir en la zona donde se encuentra la oferta de arriendo.**

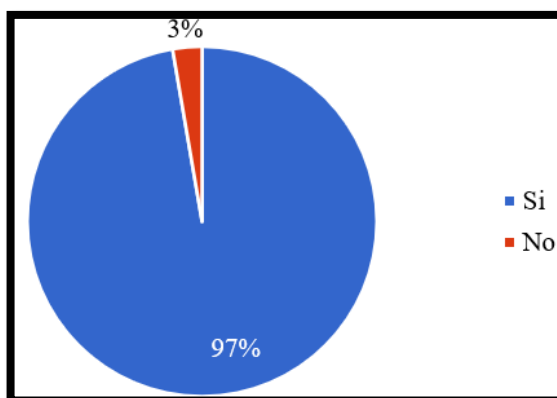
**Pregunta 10:** ¿Le resultaría útil una aplicación móvil para la búsqueda de arriendos universitarios, características, ubicación, etc.?

<b>Indicador</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Si	108	97%
No	3	3%
<b>Total</b>	111	100%

**Tabla 31.** Resultados pregunta 10 (arrendatarios).

**Elaborado por:** La investigadora.





**Figura 24.** Resultados pregunta 10 (arrendatarios).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

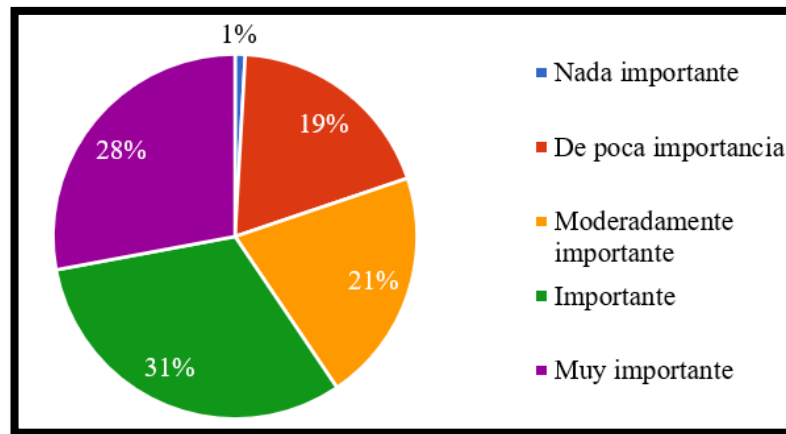
Con los resultados representados en la **Figura 24**, se evidencia **un 97% de encuestados los cuales consideran útil una aplicación móvil para la búsqueda de arriendos universitarios**, por otro lado, el 3% de los mismos desestiman la utilidad de la aplicación. Por lo tanto, con los resultados obtenidos la mayoría de estudiantes considera viable el desarrollo de la aplicación móvil.

**Pregunta 11:** ¿Considera importante la visualización del contrato de arriendo previo al contacto con el arrendador?

<b>Indicador</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Nada importante	1	1%
De poca importancia	21	19%
Moderadamente importante	23	21%
Importante	35	31%
Muy importante	31	28%
<b>Total</b>	<b>111</b>	<b>100%</b>

**Tabla 32.** Resultados pregunta 11 (arrendatarios).

**Elaborado por:** La investigadora.



**Figura 25.** Resultados pregunta 11 (arrendatarios).

**Elaborado por:** La investigadora.

### **Análisis e interpretación de resultados**

Según la **Figura 25**, el 31% de los encuestados considera muy importante la visualización del contrato de arriendo previo al contacto con el arrendador, mientras el 28% menciona la importancia de aquello, el 21% lo considera moderadamente importante, el 19% menciona que es un punto de poca importancia y únicamente el 1% lo considera nada importante. Por lo tanto, **es importante el proceso de comunicación entre arrendatario y arrendador ya que contribuye significativamente a la obtención de información apropiada por parte de la persona interesada.**

#### **2.2.4 Procesamiento y análisis de datos**

La información recolectada a través de la encuesta aplicada hacia los arrendadores demostró:

- La mayoría de los arrendadores o propietarios de los sitios de arriendo, no tienen el suficiente acercamiento hacia la tecnología o sitios donde pueden publicar la suficiente información.
- Por medio de carteles exteriores se ha dinamizado dar a conocer de cuartos o departamentos de arriendos, siendo el medio más utilizado y un método

tradicional, pero con un limitante de descripciones adecuadas sobre el sitio de arriendo.

- Un factor que desestiman los arrendadores en su mayoría es el género de la persona quien va a arrendar, sin dejar de lado la existencia de sitios donde el género femenino está por preferencia.
- Una limitante que ha sido considerada por parte de los arrendadores se basa en los gastos por elaboración y mantenimiento de una plataforma propia o exclusiva para un arrendador.
- Los propietarios de sitios de arriendos o arrendadores consideran factible el uso de una aplicación móvil donde puedan difundir la oferta de arriendo de un cuarto o departamento, debidamente descrito y con la suficiente información para el posible arrendatario.

La información recolectada a través de la encuesta aplicada hacia los arrendatarios demostró:

- La mayoría de los estudiantes desconoce de la disponibilidad de una aplicación que facilite la adquisición de arriendos universitarios.
- Las redes sociales son el medio por el que mayormente los estudiantes suelen obtener información de arriendos universitarios.
- Los estudiantes consideran muy importante la ubicación del sitio de arriendo con respecto a la Universidad Técnica de Ambato debido a que esto les permite ahorrar tiempo y dinero en desplazamientos diarios.
- Los factores de ubicación y precio predominan por parte de los estudiantes el momento de elegir un sitio para arrendar, sin dejar de lado la seguridad como el punto subsiguiente a las anteriores, de esto dependerá la adecuada selección por parte del arrendatario.
- Se evidenció la prioridad de contar con los detalles del espacio que se pretende arrendar, siendo previamente detallados antes de formalizar el contrato con el arrendador.

- La tendencia de rangos de precios conforme al presupuesto de los estudiantes se estableció en media alta, considerando que, a mayores precios, mejores prestaciones.
- Los estudiantes reconocieron la conveniencia de una aplicación móvil destinada a la búsqueda de arriendos universitarios, con el fin de optimizar las búsquedas basados en las preferencias de estos.

## CAPÍTULO III.- RESULTADOS Y DISCUSIÓN

### 3.1 Análisis y discusión de los resultados

#### 3.1.1 Impacto de los problemas en arrendadores y arrendatarios

Mediante la encuesta aplicada a los propietarios de arriendos universitarios que se encuentran alrededor de la Universidad Técnica de Ambato se analizaron las problemáticas que enfrentan los arrendadores para ofrecer el servicio de arriendo universitario, así como los desafíos para que su propiedad tenga un mayor alcance. Por otro lado, se evidenciaron las dificultades que tienen los arrendatarios para rentar, la limitada información de habitaciones, altos costos, preferencia de género, ubicación, entre otros.

<b>Problema</b>	<b>Impacto arrendador</b>	<b>Impacto arrendatario</b>
<b>Complejidad del uso de la tecnología.</b>	<ul style="list-style-type: none"> <li>• Menor alcance.</li> <li>• Menor eficiencia.</li> </ul>	<ul style="list-style-type: none"> <li>• Menor oportunidad de información.</li> </ul>
<b>Desconocimiento de aplicaciones y sitios web de arriendos.</b>	<ul style="list-style-type: none"> <li>• Menor visibilidad.</li> <li>• Pérdidas económicas.</li> <li>• Disminuye rentabilidad.</li> <li>• Menor alcance geográfico.</li> </ul>	<ul style="list-style-type: none"> <li>• Riesgo de fraude.</li> <li>• Búsqueda prolongada y cansada.</li> <li>• Pérdida de tiempo.</li> <li>• Menor oferta de arriendos</li> </ul>
<b>Preferencia de género.</b>	<ul style="list-style-type: none"> <li>• Menor demanda de arrendatarios.</li> </ul>	<ul style="list-style-type: none"> <li>• Menor oportunidad de arriendos.</li> <li>• Discriminación.</li> </ul>
<b>Limitada disponibilidad de arriendos.</b>	<ul style="list-style-type: none"> <li>• Ninguna.</li> </ul>	<ul style="list-style-type: none"> <li>• Pérdida de tiempo.</li> <li>• Búsqueda prolongada y cansada.</li> </ul>

		<ul style="list-style-type: none"> <li>• Aumento de precios.</li> </ul>
<b>Ubicación del sitio de arriendo.</b>	<ul style="list-style-type: none"> <li>• Menor demanda de arrendatarios.</li> <li>• Bajos precios de arriendo.</li> </ul>	<ul style="list-style-type: none"> <li>• A mayor distancia, mayor tiempo y costo de transporte.</li> <li>• Fatiga y estrés.</li> </ul>
<b>Precios elevados de arriendos.</b>	<ul style="list-style-type: none"> <li>• Menor demanda de arrendatarios.</li> </ul>	<ul style="list-style-type: none"> <li>• Dificultad para afrontar el gasto fuera del presupuesto.</li> <li>• Limitaciones de oportunidad de arriendos.</li> <li>• Impacto en la salud mental.</li> </ul>

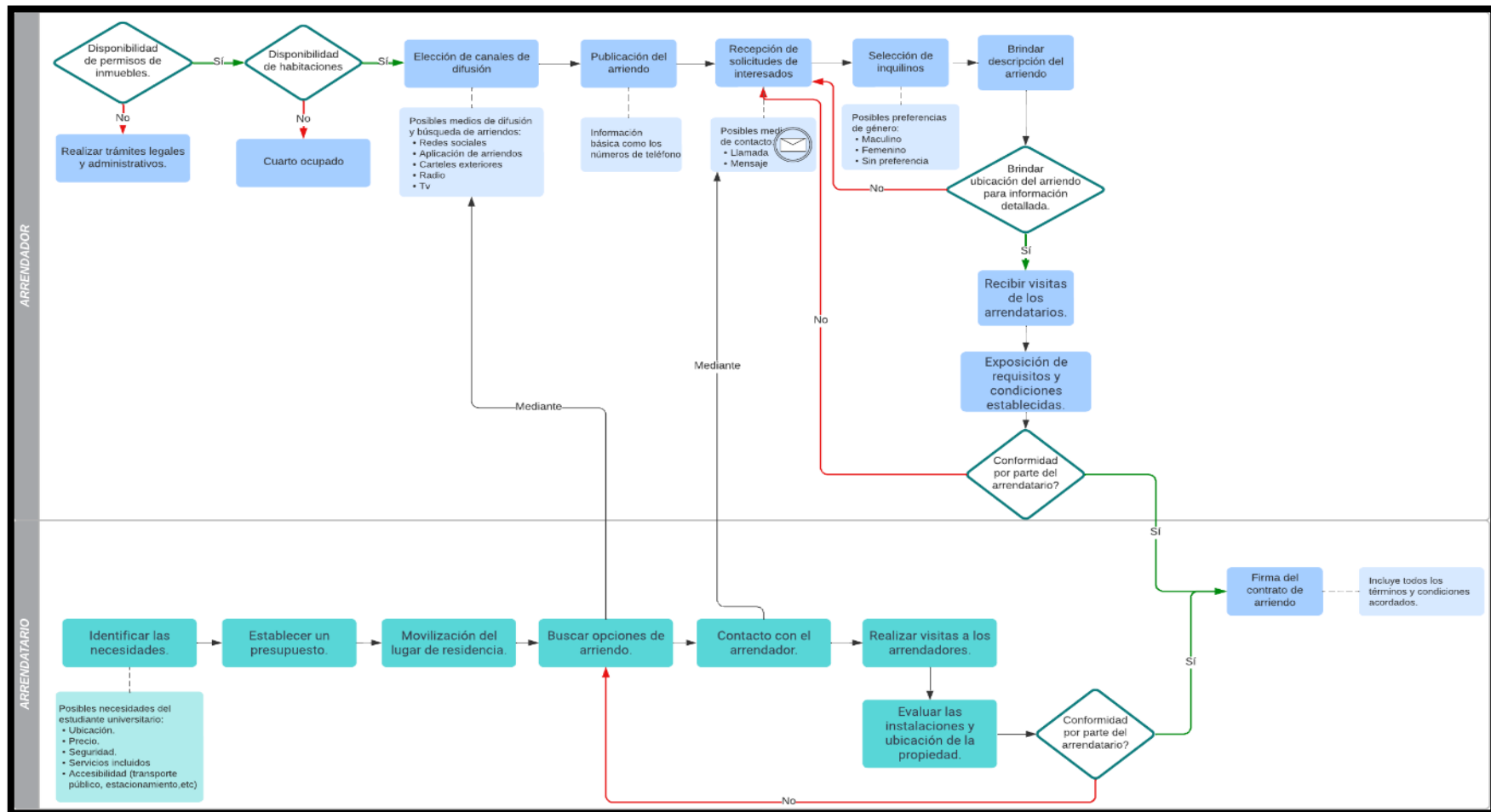
**Tabla 33.** Problemas y su impacto en el arrendador y arrendatario.

**Elaborado por:** La investigadora.

De acuerdo con el análisis realizado en la **Tabla 33**, tanto el arrendador como arrendatario desconocen de aplicaciones y sitios web de arriendos, entre otros hallazgos los cuales a través del desarrollo de la aplicación móvil se pueden solventar, con la finalidad de contribuir positivamente en el proceso difusión y búsqueda de arriendos universitarios.

### **3.1.2 Modelado de procesos de arrendadores y arrendatarios**

En la **Figura 26**, se detallan los procesos que realiza tanto el arrendador para la difusión de su arriendo, así como el arrendatario para la búsqueda de una habitación que se acople con su presupuesto y necesidades.



**Figura 26 .** Procesos para la difusión y búsqueda de arriendos.

**Elaborado por:** La investigadora.

### 3.1.3 Tecnologías multiplataforma para el desarrollo móvil.

Para definir la mejor tecnología multiplataforma para el desarrollo de aplicaciones móviles, es importante considerar características como: el tipo de aplicación, la plataforma, el lenguaje de programación, las herramientas y recursos disponibles. Dichos factores ayudan a definir la adecuada tecnología para el desarrollo del proyecto.

<b>Tema</b>	<b>Xamarin Forms</b>	<b>Flutter</b>	<b>React Native</b>
<b>Código</b>	Open source	Open source	Open source
<b>Lenguaje de programación.</b>	C#	Dart	JavaScript
<b>Plataformas</b>	Android, iOS y Windows	Android, iOS, Windows, Mac OS, Linux	Android, iOS, UWP (Windows)
<b>Curva de aprendizaje</b>	Media [18].	Baja [17], [40], [19].	Alta [17].
<b>Tipo de aplicaciones</b>	Multiplataforma	Multiplataforma	Multiplataforma
<b>Rendimiento</b>	Alto rendimiento y velocidad en la creación de aplicaciones móviles nativas. Desarrollo de la interfaz de usuario (UI) requiere mucho tiempo. Baja actividad comunitaria.	Alto rendimiento y capacidad para generar vistas de alta calidad y altamente personalizables. Permite utilizar código nativo para lograr un mejor rendimiento	Alto rendimiento en aplicaciones nativas. La navegación de la aplicación móvil no sería equivalente a la navegación nativa.



<b>Componente de interfaz de usuario</b>	Interfaces de usuario nativas del dispositivo.	Motor gráfico personalizado.	Proporciona controles de interfaz de usuario nativa y acceso completo a la plataforma nativa.
--	--	------------------------------	---

**Tabla 34.** Comparación entre tecnologías multiplataforma.

**Elaborado por:** La investigadora.

Mediante el análisis realizado en la **Tabla 34**, detallando características técnicas que poseen cada una de las tecnologías, se decidió optar por el desarrollo en Flutter ya que a través de Dart como el framework de programación permite desarrollar una aplicación para distintos sistemas operativos, ofreciendo así una curva de aprendizaje inferior en comparación a las demás tecnologías, principalmente por el uso de un único lenguaje simple y sencillo.

Además, facilita el uso de código nativo del dispositivo para un mejor rendimiento, posee la capacidad de generar vistas de alta calidad y altamente personalizables y utiliza su propio motor de renderizado de alto rendimiento para dibujar widgets.

Flutter cuenta con una comunidad creciente y entusiasta de la tecnología la cual puede apoyar en el desarrollo de la aplicación, contando también con forros de control de calidad. Sitios web y documentación variada lo que hace que el aprendizaje de la tecnología se torne rápido y sencillo [40].

### **3.1.4 Metodologías para el desarrollo móvil.**

El siguiente cuadro comparativo posee las características más relevantes de cada una de las metodologías ágiles con el fin de enfatizar la entrega rápida y continua de entregables.

<b>Criterio</b>	<b>Mobile D</b>	<b>XP</b>	<b>Scrumban</b>
<b>Tamaño de los proyectos</b>	Pequeños	Pequeños y medianos.	Adaptable a proyectos grandes y pequeños.
<b>Tamaño de equipo</b>	No más de 10	Menor que 10	Menos de 12 miembros.
<b>Estilo de desarrollo</b>	Iterativo e incremental.	Iterativo e incremental.	Ágil e iterativo.
<b>Estilo de código</b>	Limpio y sencillo	Limpio y sencillo	No especificado
<b>Fases o etapas</b>	<ul style="list-style-type: none"> <li>• Exploración</li> <li>• Iniciación</li> <li>• Producción</li> <li>• Estabilización</li> <li>• Pruebas del Sistema</li> </ul>	<ul style="list-style-type: none"> <li>• Planeación</li> <li>• Diseño</li> <li>• Desarrollo</li> <li>• Pruebas</li> </ul>	<ul style="list-style-type: none"> <li>• Establecer los objetivos.</li> <li>• Backlog de producto.</li> <li>• Planificación.</li> <li>• Standup diario.</li> <li>• Revisión.</li> <li>• Retrospectiva.</li> <li>• Entrega continua.</li> </ul>
<b>Duración</b>	No más de 10 semanas.	Dura entre una y cuatro semanas.	Sprints con duración de 1 a 4 semanas.
<b>Complejidad</b>	Media [27].	Alta [23].	Media [23].

**Tabla 35.** Metodologías para el desarrollo móvil.

**Elaborado por:** La investigadora.

Mediante el análisis realizado en la **Tabla 35**, de las metodologías ágiles para el desarrollo de aplicaciones móviles, se eligió a la metodología Mobile-D considerando sus beneficios y características como: rapidez en el tiempo de desarrollo, flexibilidad en torno a los cambios, estilo de código y el tamaño pequeño del equipo de desarrollo, con lo cual ayuda a cumplir con los requerimientos y funcionalidades del desarrollo de la aplicación móvil.

### 3.2 Desarrollo de la propuesta

Para el desarrollo del proyecto se aplicarán las 5 fases que posee la metodología Mobile-D: exploración, iniciación, producción, estabilización y prueba del sistema. Las cuales permiten un desarrollo más rápido y flexible de las aplicaciones móviles.

#### 3.2.1 Fase 1: Exploración

En esta fase se genera un plan y se establecen las características del proyecto. Lo cual se realiza en tres etapas: establecimiento de los interesados, definición del alcance y el establecimiento de proyectos.

##### A) Establecimiento de los interesados (Stakeholders)

<b>Interesados</b>	<b>Descripción</b>
<b>Desarrollador</b>	Persona que se dedica a desarrollar aplicaciones informáticas con ayuda de lenguajes de programación, herramientas de desarrollo y metodologías ágiles, con el fin de crear soluciones para el usuario, cumpliendo sus requisitos y necesidades.
<b>Arrendador</b>	Persona que posee una propiedad, la cual se alquila o arrienda a las demás personas con el fin de obtener una renta periódica, el mismo que se encarga de mantener la propiedad en condiciones habitables.
<b>Arrendatario</b>	Persona que alquila una propiedad, teniendo la responsabilidad de pagar la renta que se acuerde con el arrendador y cumplir los términos y condiciones previamente aceptadas.

**Tabla 36.** Establecimiento de los interesados.

**Elaborado por:** La investigadora.

##### B) Alcance

Las siguientes características definen el alcance que pretende cumplir el presente proyecto.

### B.1) Limitaciones

La aplicación móvil posee limitaciones como:

- Para que la aplicación móvil se ejecute, el dispositivo debe mantener la conexión a internet.
- Los dispositivos deben poseer un sistema operativo 7.0 o superior en Android, mientras que para iOS la versión del sistema operativo debe ser igual o superior a la versión 10.
- La aplicación móvil está destinada para la difusión y búsqueda de arriendos universitarios, más no para la reservación de estos.

### B.2) Establecimiento de dependencias

Las dependencias encontradas en el presente proyecto son las siguientes:

- La aplicación móvil es destinada solo a los arrendadores que poseen un arriendo universitario alrededor la de UTA.
- La búsqueda esta centralizada hacia arriendos universitarios en los alrededores de la UTA.

### B.3) Requisitos funcionales para arrendadores/propietarios

Mediante la investigación de campo en donde las encuestas fueron aplicadas a los arrendadores, se ha destacado la inclusión de las siguientes funcionalidades que la aplicación móvil poseerá, detalladas en la **Tabla 37**.

<b>Número de requerimiento</b>	<b>Requerimiento</b>	<b>Descripción</b>
RF-1	Registro del arrendador.	Los arrendadores necesitan registrarse previamente en la aplicación para realizar una publicación de un arriendo, detallando información como: nombre, teléfono, correo personal, contraseña.
RF-2	Inicio de sesión.	El arrendador podrá iniciar sesión utilizando las credenciales definidas previamente en el registro.

RF-3	Publicar arriendo.	El arrendador del arriendo puede publicar el arriendo detallando características como: fotos, ubicación, precio, tamaño.
RF-4	Publicar contrato de arrendamiento.	El arrendador podrá agregar el contrato en formato PDF para su visualización.
RF-5	Publicar detalle de servicios.	La aplicación ofrece el detalle de los servicios básicos y adicionales que se oferta, así como: internet, agua, luz, parqueadero e información adicional de la garantía.
RF-6	Publicar disponibilidad de cuartos.	El arrendador podrá definir la visibilidad de la disponibilidad del cuarto o departamento de arriendo.
RF-7	Gestión de arriendos.	El arrendador podrá administrar la información de los distintos cuartos de arriendo, así como adicionar nuevos o eliminarlos.
RF-8	Editar información personal.	El arrendador tendrá la disponibilidad de editar su información personal en los ajustes de la aplicación.

**Tabla 37.** Requisitos funcionales de los arrendadores.

**Elaborado por:** La investigadora.

### **B.3) Requisitos funcionales para arrendatarios**

Mediante la investigación de campo en donde las encuestas fueron aplicadas a los arrendatarios, se ha destacado la inclusión de las siguientes funcionalidades que la aplicación móvil poseerá, detalladas en la **Tabla 38**.

<b>Número de requerimiento</b>	<b>Requerimiento</b>	<b>Descripción</b>
RF-1	Búsqueda de arriendos.	Los arrendatarios tienen la posibilidad de buscar propiedades de alquiler utilizando criterios propios.
RF-2	Filtros de Búsqueda Avanzada.	Los arrendatarios dispondrán de distintos filtros como: la ubicación, precio, género, permisos de mascotas, entre otros con la finalidad de hacer eficiente la búsqueda.
RF-3	Visualización de Resultados de Búsqueda.	Los arrendatarios podrán visualizar propiedades del arriendo que se acoplen a sus necesidades y criterios de búsqueda.
RF-4	Detalles de los arriendos.	Los arrendatarios tendrán la posibilidad de acceder a la información detallada del arriendo, incluyendo fotos, precio, tamaño, servicios básicos y la visualización del contrato de arriendo.
RF-5	Mensajería.	La aplicación ofrece la posibilidad de contactarse con el propietario a través de WhatsApp, o llamada telefónica.
RF-6	Calificaciones y reseñas.	Los arrendatarios podrán aportar con una opinión acerca de la experiencia en un departamento o cuarto de arriendo en específico.
RF-7	Soporte al usuario.	La aplicación permitirá contactarse con el desarrollador en caso de suscitarse problemas con la misma.
RF-8	Registro de Usuario.	Los arrendatarios tendrán la posibilidad de registrarse en la aplicación, detallando información básica como: nombre, teléfono, correo personal, contraseña.

RF-9	Inicio de Sesión.	El arrendatario podrá iniciar sesión utilizando las credenciales definidas previamente en el registro.
------	-------------------	--

**Tabla 38.** Requisitos funcionales de los arrendatarios.

**Elaborado por:** La investigadora.

### C) Definición del proyecto

Esta etapa define el tipo de herramientas que se utilizarán en el desarrollo de la aplicación móvil:

- **Visual Studio Code**

Visual Studio Code es un editor de código potente disponible para Windows, macOS y Linux. Posee una interfaz intuitiva, variedad de extensiones y compatibles con múltiples lenguajes de programación (como C++, C#, Java, Python, PHP, Go, .NET). Posee un soporte integrado para JavaScript, TypeScript y Node.js.

- **Firebase**

Firebase se define como una plataforma para el desarrollo de aplicaciones tanto web como móviles, su finalidad es ayudar a compilar y desarrollar distintas aplicaciones. Cuenta con el respaldo de Google y ofrece una gestión de usuario segura, permitiendo a los desarrolladores el uso de métodos como la autenticación de usuario.

- **Firebase Database**

Es una base de datos que usa un modelo NoSQL, almacenando datos como objetos JSON. Permitiendo a los desarrolladores obtener datos en tiempo real, convirtiéndola en ideal para aplicaciones móviles. Es producto de Google que incluye informe de fallos, supervisión de rendimiento, mensajería, autenticación y más.

- **Firebase Authentication**

Ofrece la atención a los usuarios ya sea en sus aplicaciones móviles o web con proveedores de identidad como Google Facebook, Twitter y cuentas de correo electrónico.

- **Flutter**

Flutter representa un marco de desarrollo de código abierto creado por Google con el propósito de construir diversas aplicaciones móviles multiplataforma que se compilan de manera nativa, empleando una base de código única. Entre sus beneficios se encuentra su aptitud para generar interfaces de usuario fluidas y atractivas mediante widgets personalizables.

- **Dart**

Es el lenguaje de programación utilizado por el framework Flutter para desarrollar aplicaciones móviles, web o de escritorio, su sintaxis similar a los demás lenguajes de programación determina un punto clave a la hora de aprender dicho lenguaje.

### 3.2.2 Fase 2: Inicialización

En esta fase se identifican los recursos necesarios para el diseño y desarrollo de la aplicación móvil.

#### A) Configuración del ambiente de desarrollo

Se detallan las herramientas que se necesitaron instalar y configurar para el desarrollo del proyecto.

- **Tipo de aplicación:** Aplicación multiplataforma.
- **Lenguaje de programación:** Dart.
- **Desarrollo del front-end:** Flutter.
- **Servicio de Autenticación:** Firebase Authentication.
- **Sistema gestor de base de datos:** Firestore Database.



### A.1) Preparación del ambiente

Se detallan las herramientas instaladas para el correcto desarrollo de la aplicación móvil.

- Visual Studio Code.
- Framework Flutter.
- Dart.
- Firebase.

### B) Planificación de fases

Fase	Iteración	Descripción
Exploración	Iteración 0	<ul style="list-style-type: none"><li>• Establecer interesados (stakeholders).</li><li>• Definir alcances y limitaciones.</li><li>• Definir el proyecto.</li></ul>
Iniciación	Iteración 1	<ul style="list-style-type: none"><li>• Definición y establecimiento de requerimientos.</li><li>• Configuración del ambiente de desarrollo.</li></ul>
Producción	Iteración 2	<ul style="list-style-type: none"><li>• Implementación de la interfaz de inicio o Splash Screen.</li><li>• Elaboración de storycard.</li><li>• Pruebas de funcionalidad.</li></ul>
	Iteración 3	<ul style="list-style-type: none"><li>• Implementación de la función de registro e inicio de sesión.</li><li>• Elaboración de storycard.</li><li>• Pruebas de funcionalidad.</li></ul>
	Iteración 4	<ul style="list-style-type: none"><li>• Implementación del módulo de mapas.</li><li>• Elaboración de storycard.</li><li>• Pruebas de funcionalidad</li></ul>
	Iteración 5	<ul style="list-style-type: none"><li>• Implementación del módulo de perfil de usuario.</li></ul>

		<ul style="list-style-type: none"> <li>• Elaboración de storycard.</li> <li>• Pruebas de funcionalidad.</li> </ul>
	Iteración 5	<ul style="list-style-type: none"> <li>• Implementación de la función de publicar arriendos.</li> <li>• Elaboración de storycard.</li> <li>• Pruebas de funcionalidad.</li> </ul>
	Iteración 6	<ul style="list-style-type: none"> <li>• Implementación del módulo para obtener listado de arriendos publicados opción propietarios.</li> <li>• Elaboración de storycard.</li> <li>• Pruebas de funcionalidad.</li> </ul>
	Iteración 7	<ul style="list-style-type: none"> <li>• Implementación del módulo borrar y editar arriendos de la lista del arrendador.</li> <li>• Elaboración de storycard.</li> <li>• Pruebas de funcionalidad.</li> </ul>
	Iteración 8	<ul style="list-style-type: none"> <li>• Implementación de la funcionalidad de visualización rápida de detalles del arriendo en el mapa.</li> <li>• Elaboración de storycard.</li> <li>• Pruebas de funcionalidad.</li> </ul>
	Iteración 9	<ul style="list-style-type: none"> <li>• Implementación de la funcionalidad de filtros de usuario.</li> <li>• Elaboración de storycard.</li> <li>• Pruebas de funcionalidad</li> </ul>
	Interacción 10	<ul style="list-style-type: none"> <li>• Implementación de la función de geolocalización.</li> <li>• Elaboración de storycard.</li> <li>• Pruebas de funcionalidad.</li> </ul>
Estabilización	Iteración 11	<ul style="list-style-type: none"> <li>• Corrección de errores generales.</li> <li>• Adaptación de datos.</li> </ul>

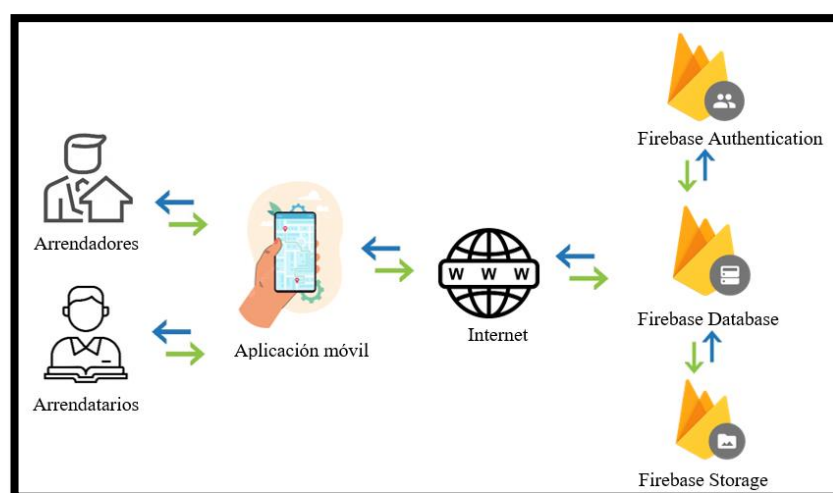
Pruebas del sistema	Iteración 12	<ul style="list-style-type: none"> <li>• Realización de pruebas de aceptación a la aplicación móvil.</li> <li>• Pruebas de unidad a la aplicación.</li> </ul>
---------------------	--------------	---

**Tabla 39.** Planificación de fases.

**Elaborado por:** La investigadora.

### C) Diseño de la arquitectura de la aplicación móvil.

La arquitectura de la aplicación móvil ha sido diseñada de forma que sea funcional para dos tipos de usuarios, los cuales requieren conectividad a internet para acceder a los servicios proporcionados por Firebase como: autenticación, almacenamiento y esquema de base de datos. En la **Figura 27**, se observa el proceso que interviene en el funcionamiento de la aplicación.



**Figura 27.** Arquitectura de la aplicación móvil.

**Elaborado por:** La investigadora.

### D) Diseño de la base de datos.

La base de datos en la cual se aloja el proyecto es Firebase Database desarrollada por Google, la misma que es una base de datos NoSQL que a diferencia de una base de datos SQL no posee tablas ni registros sino más bien con una estructura de datos basada en documentos y colecciones.

- **Colecciones principales**



**Figura 28.** Colecciones en Firebase.

**Elaborado por:** La investigadora.

**Colección publicaciones**

<b>Campo</b>	<b>Tipo de dato</b>
id	string
comentarios	array
contrato	string
descripcion	string
habilitado	booleano
habitacion	mapa
imagenes	array
ubicacion	mapa
uidUsuario	string

**Tabla 40.** Estructura de la Colección publicaciones.

**Elaborado por:** La investigadora.

- **Colección usuarios**

<b>Campo</b>	<b>Tipo de dato</b>
apellido	String
contraseña	String
email	String
fecha_creacion	String
nombre	String
phone	String
rol	String
uid	String

**Tabla 41.** Estructura de la Colección usuarios.

**Elaborado por:** La investigadora.

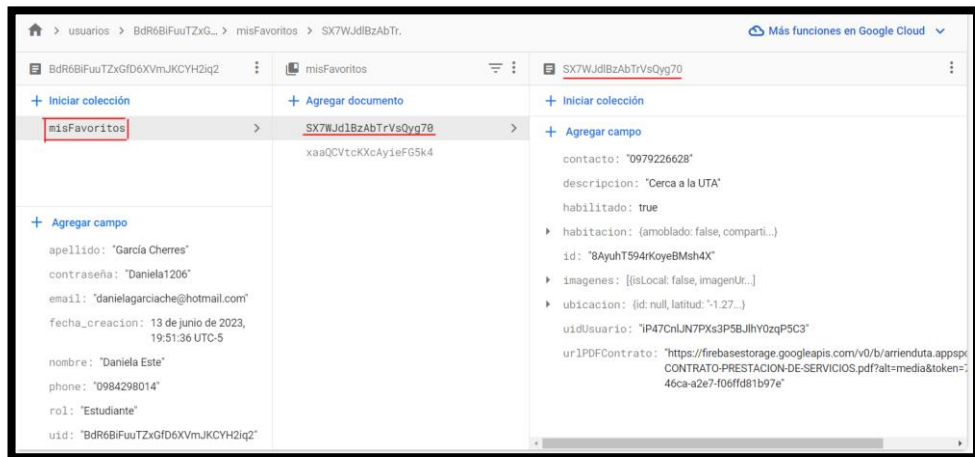
- **Subcolección misFavoritos**

<b>Campo</b>	<b>Tipo de dato</b>
uidUsuario	string
publicacion	mapa

**Tabla 42.** Estructura de la Subcolección misFavoritos.

**Elaborado por:** La investigadora.

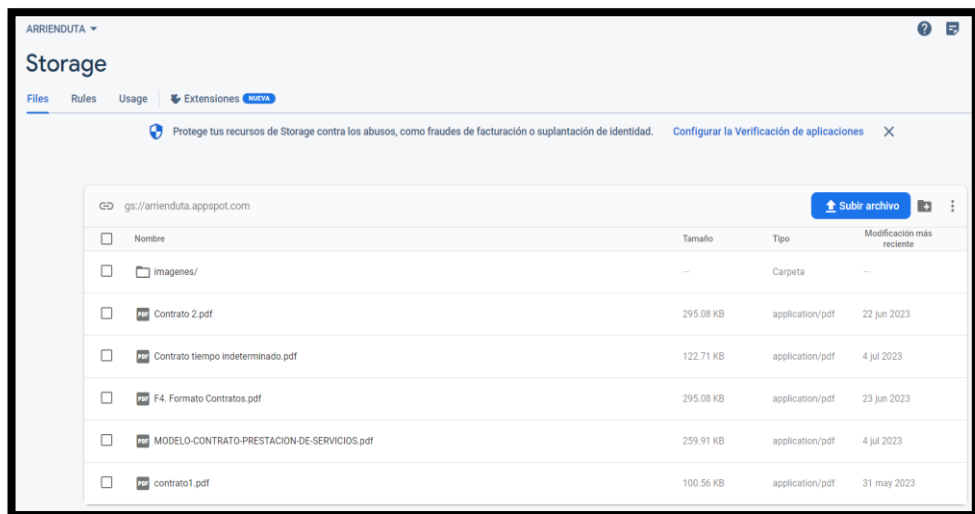
- **Subcolección favoritos**



**Figura 29.** Subcolección de la lista de arriendos favoritos.

**Elaborado por:** La investigadora.

- **Storage de Firebase**



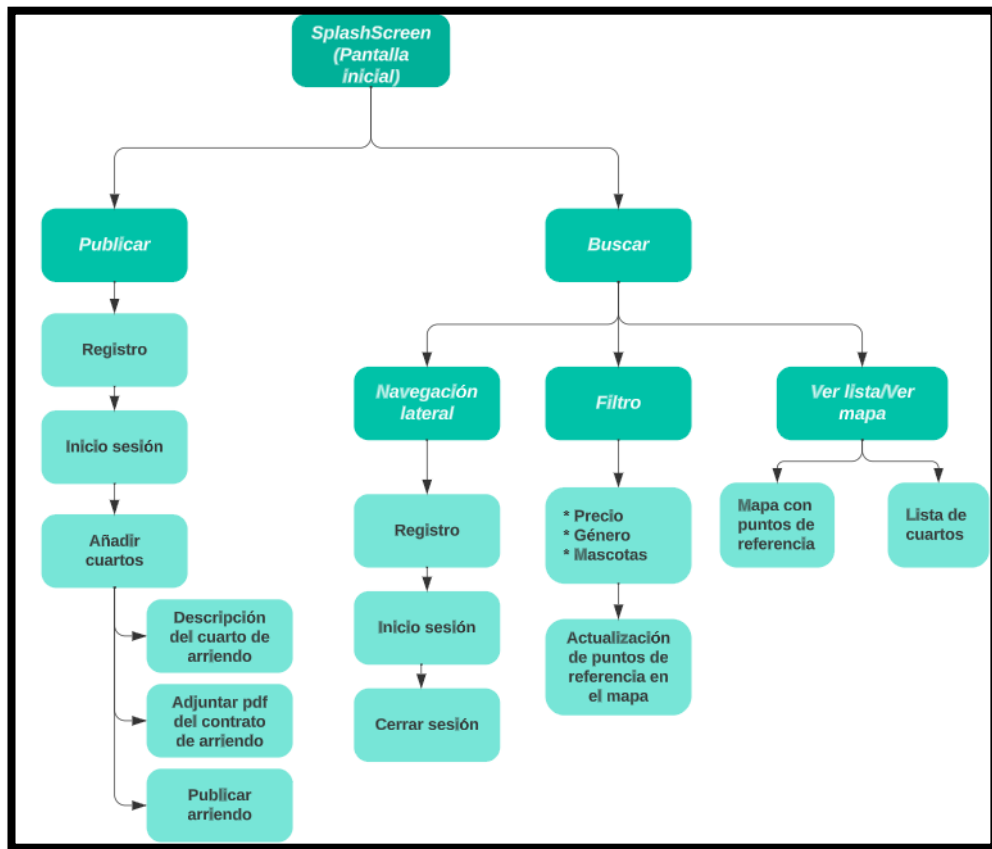
**Figura 30.** Storage para almacenar PDF e imágenes en Firebase.

**Elaborado por:** La investigadora.

### **E) Esquema de Navegabilidad**

En la **Figura 31**, se representa la estructura y flujo de navegación de la aplicación móvil, la cual proporciona la descripción de las funciones y posibles rutas de navegación entre ellas. Además de brindar una clara comprensión de su arquitectura y

la interacción que tendrá tanto el arrendador como el arrendatario con la aplicación móvil.

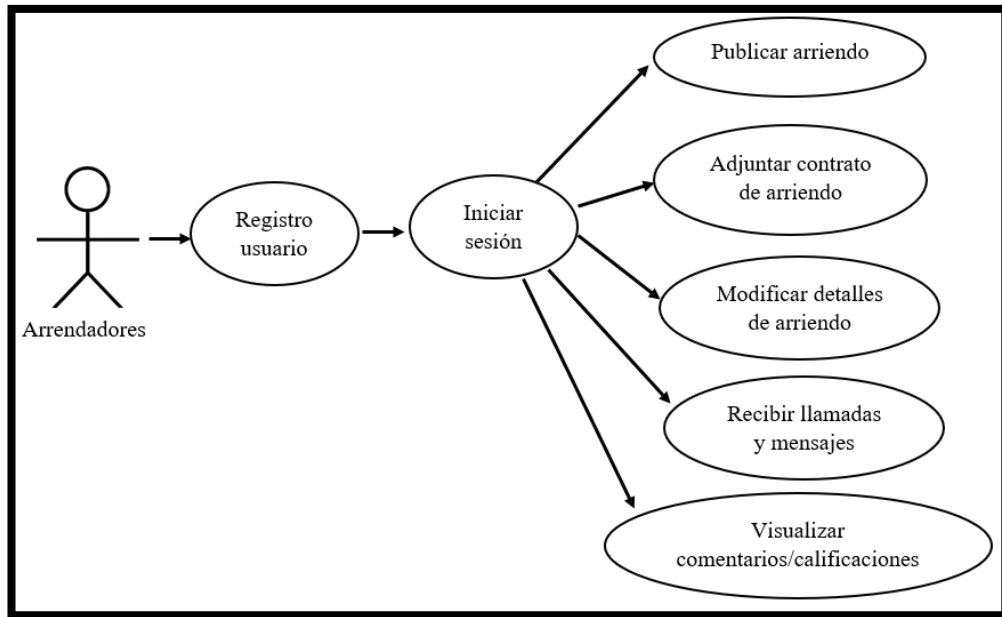


**Figura 31.** Esquema de navegabilidad.

**Elaborado por:** La investigadora.

## F) Diagramas de casos de uso

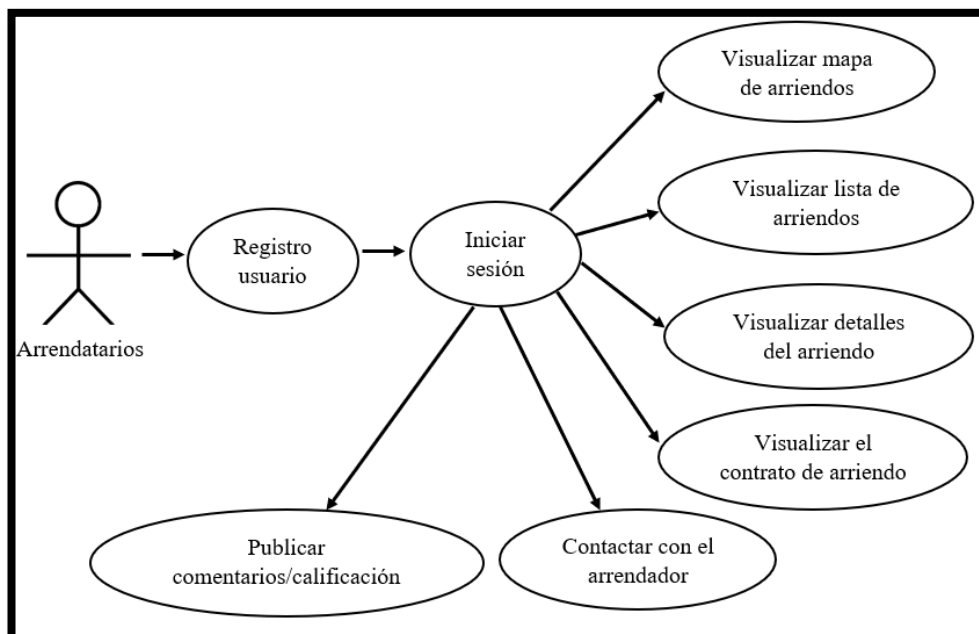
Los diagramas de caso de uso describen las interacciones entre los actores (arrendadores y arrendatarios) y la aplicación móvil. En la **Figura 32**, se describe un escenario en el que se encuentra el arrendador como involucrado y las acciones o funciones que la aplicación móvil le permite realizar.



**Figura 32.** Diagrama de caso de uso arrendadores.

**Elaborado por:** La investigadora.

En la **Figura 33**, se describe un escenario en el que se encuentra el arrendatario como involucrado y las acciones o funciones que la aplicación móvil le permite realizar.



**Figura 33.** Diagramas de casos de uso arrendatarios.

**Elaborado por:** La investigadora.



## G) Prototipo de la aplicación

El diseño de interfaz de usuario tiene un impacto significativo en el usuario ya que es influye en la experiencia y percepción de herramienta de la calidad del producto, por lo que se desea crear una interfaz gratificante con la herramienta de diseño Balsamiq al cual permite diseñar, visualizar y comunicar ideas de diseño de manera efectiva y rápida.

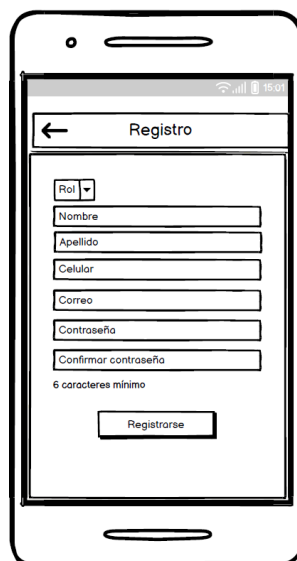
**Pantalla inicial:** El usuario al abrir la aplicación móvil visualiza una pantalla de inicio o Splash Screen que dura aproximadamente 3 segundos la cual se encuentra representada en la **Figura 34**, con el fin de retroalimentar al usuario que la aplicación se carga en segundo plano.



**Figura 34.** Pantalla inicial.

**Elaborado por:** La investigadora.

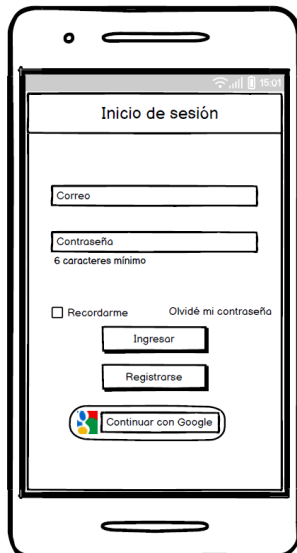
**Registro:** Antes de realizar el inicio de sesión el usuario debe registrarse y proporcionar datos como el rol que desea adquirir ya sea el de estudiante o el de arrendador, seguido del nombre, apellido, teléfono, correo electrónico, contraseña y confirmar la contraseña como se detalla en la **Figura 35**, esto permitirá al arrendador publicar los detalles de su cuarto de arriendos y al arrendatario la búsqueda de un cuarto de arriendo.



**Figura 35.** Registros arrendadores y arrendatarios.

**Elaborado por:** La investigadora.

**Inicio de sesión:** una vez que el arrendador o arrendatario hayan creado una cuenta, es necesario iniciar sesión por medio del correo y contraseña como se muestra en la **Figura 36.**



**Figura 36.** Inicio de sesión arrendadores y arrendatarios.

**Elaborado por:** La investigadora.

**Mapa de búsqueda:** en la **Figura 37**, se muestra que el arrendatario posee un mapa al cual se le puede aplicar un filtro que agilizará la búsqueda. Además, posee una lista de cuartos de arriendos como se indica en la **Figura 38.**



**Figura 37.** Mapa de búsqueda.

**Elaborado por:** La investigadora.



**Figura 38.** Lista de cuarto de arriendo.

**Elaborado por:** La investigadora.

**Filtro de búsqueda:** Una vez aplicado el filtro de búsqueda, el mapa muestra los resultados obtenidos como indica la **Figura 39**, y el arrendatario puede seleccionar los punteros de ubicación y dirigirse a los detalles del cuarto seleccionado en el cual se detalla el tamaño en metros cuadrados, precio, los servicios básicos disponibles, las preferencias de género y si se permiten mascotas, una descripción general del cuarto,

además de la opción de contactarse con el arrendador y calificar la experiencia si alguna vez ya ha arrendado en dicha habitación como se muestra en la **Figura 40**.



**Figura 39.** Filtro de búsqueda.

**Elaborado por:** La investigadora.



**Figura 40.** Detalle del cuarto de arriendo.

**Elaborado por:** La investigadora.

**Publicación de arriendos:** Una vez que el arrendador haya iniciado sesión, se le permite publicar el cuarto de arriendo como indica la **Figura 41**, añadiendo características como los metros cuadrados, precio, ubicación, si es amoblado, disponibilidad de cocina, la opción de adjuntar el contrato de arriendo en formato de documento portable (PDF), adjuntar imágenes, servicios básicos que incluyen,

disponibilidad de parqueadero, los meses de garantía, preferencias de género, y la posibilidad de la aceptación de mascotas que se presenta en la **Figura 42**.



**Figura 41.** Añadir cuarto.

**Elaborado por:** La investigadora.



**Figura 42.** Detalles del cuarto.

**Elaborado por:** La investigadora.

**Arriendos favoritos:** el arrendatario puede agregar departamentos preferidos ya sea por servicio o precios que se ajusten a sus necesidades, las cuales se añadirán a la lista de cuartos favoritos como indica la **Figura 43**.



**Figura 43.** Arriendos favoritos.

**Elaborado por:** La investigadora.

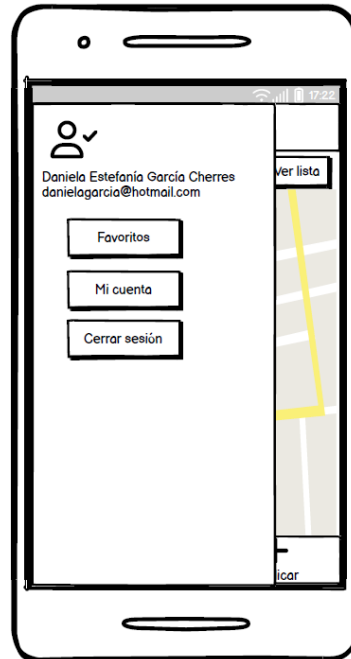
**Ajustes de cuenta:** esta opción mostrará las opciones de actualización de nombre, apellido y celular como indica la **Figura 44**.



**Figura 44.** Ajustes de cuenta.

**Elaborado por:** La investigadora.

**Cerrar sesión:** en este proceso el arrendador o arrendatario finaliza sesión para desconectarse de su cuenta en la aplicación móvil, la cual se visualiza en la **Figura 45**.



**Figura 45.** Cerrar sesión.

**Elaborado por:** La investigadora.

### 3.2.3 Fase 3: Producción

Durante esta fase, se llevan a cabo las iteraciones descritas en la **Tabla 38**, mediante la entrega de Storyboards detallados como entregables de las actividades.

#### A) Backend

Firestore fue utilizado como plataforma BaaS (Backend as a Service), permitiendo crear la aplicación móvil sin la necesidad de programar en el servidor, y logrando un desarrollo más ágil y simplificado.

Los datos en Firestore se almacenan en tiempo real como objetos JSON. Considerando a un árbol JSON alojado en la nube como una base de datos que a diferencia de una base de datos SQL no cuenta con tablas ni registros sino más bien con una estructura

de datos basada en documentos y colecciones. Una colección es semejante a una tabla en una base de datos SQL y un documento como semejante a un registro en esa tabla.

El árbol con estructura JSON tiene asociada una clave la cual puede ser reemplazada por una propia con un ID de usuario, o se puede obtener mediante el método push(). Para crear claves propias se debe tomar en cuenta la codificación UTF-8, con un máximo de 768 bytes y no pueden contener los caracteres., \$, #, [, ], / ni los caracteres de control ASCII del 0 al 31 ni el 127, [41].

Se empleó el archivo de configuración de Firebase el cual contiene información como el ID del proyecto, las claves de API y los identificadores de cliente utilizados para establecer conexión entre la aplicación con los servicios de Firebase.

```
{ } google-services.json X
android > app > {} google-services.json > [ ] client > {} > [ ] oauth_client > {} 0
1
2 {
3   "project_info": {
4     "project_number": "529301871128",
5     "project_id": "arrienduta",
6     "storage_bucket": "arrienduta.appspot.com"
7   },
8   "client": [
9     {
10    "client_info": {
11      "mobilesdk_app_id": "1:529301871128:android:672a558002e399933c416d",
12      "android_client_info": {
13        "package_name": "com.example.arriendos_uta"
14      }
15    },
16    "oauth_client": [
17      {
18        "client_id": "529301871128-e1qatnp1f4cejkcg10r281237m6g135t.apps.googleusercontent.com",
19        "client_type": 1,
20        "android_info": {
21          "package_name": "com.example.arriendos_uta",
22          "certificate_hash": "fe048be62aa7366d570318c19c346bd2ffacb14f"
23        }
24      },
25      {
26        "client_id": "529301871128-tfh130u3jnn3aja8gaq7t73sv4fsd6pu.apps.googleusercontent.com",
27        "client_type": 3
28      }
29    ],
30    "api_key": [
31      {
32        "current_key": "AIzaSyBnaJpCMxEX6gAcjldpHFahgr1NS90Z1j0"
33      }
34    ],
35    "services": {
36      "appinvite_service": {
37        "other_platform_oauth_client": [
38          {
39            "client_id": "529301871128-mdv60cc808sap4f17j2dn5gm14jj7opq.apps.googleusercontent.com",
40            "client_type": 3
41          }
42        ]
43      }
44    }
45  ],
46  "configuration_version": "1"
}
```

**Figura 46.** Archivo de configuración de Firebase.

**Elaborado por:** La investigadora.



Se utilizó el patrón de diseño Modelo-Vista-Controlador (MVC) con el objetivo de obtener un código escalable. El modelo se encarga de la gestión de datos y la lógica de negocio, la vista de la presentación e interacción en la interfaz de usuario.

## **Modelo**

En esta capa se trabaja con los datos, cuenta con mecanismos para acceder a la información y actualizar su estado.

Listado de modelos:

- auxpublicacion.dart
- comentario.dart
- conrato.dart
- filtropage.dart
- habitacion.dart
- imagen.dart
- publicacion.dart
- servicioBasico.dart
- ubicacion.dart
- usuario.dart

## **Vista**

Es la capa que contiene el código responsable de generar las interfaces de usuario que deberán contener los elementos de interacción, permitiendo al usuario enviar información e invocarlas en el sistema.

Listado de vistas:

- filtro\_pagina.dart
- listado\_publicaciones.dart
- mapaDetalle.dart
- misFvaortios.dart
- reconstruccion\_publicacion.dart

- editar\_publicacion.dart
- lista\_publicacion.dart
- atepper\_page.dart
- borrar.dart
- hime\_pagina.dart
- login\_pagina.dart
- register\_pagina.dart
- splash\_screen.dart

### **Controlador**

Es la capa intermediaria entre el usuario y el sistema, captura las acciones de este sobre la vista, además de realizar tareas de transformación de datos para que los componentes de la Vista y el modelo se entienda.

Listado de controladores:

- auth.controller.dart
- PdfStorageController.dart
- Publicación.controller.dart
- User.controller.dart

### **Dependencias**

Las dependencias agregadas en Flutter fueron realizadas con el fin de agregar funcionales y características adicionales a la aplicación de manera eficiente y confiable.

```
11
12 dependencies:
13   flutter:
14     sdk: flutter
15   #Guardar datos en firestore
16   firebase_core: ^2.13.0
17   cloud_firestore: ^4.7.1
18   firebase_auth: ^4.6.1
19   google_sign_in: ^6.1.2
20   flutter_native_splash: ^2.3.0
21   logger: ^1.1.0
22   flutter_pdfview: ^1.3.1
23   firebase_storage: ^11.2.2
24   file_picker: ^5.3.2
25   flutter_map_marker_popup: ^4.0.3
26   geolocator: ^9.0.2
27   cached_network_image: ^3.2.0
28   flutter_map: ^3.1.0
29   latlong2: ^0.8.2
30   image_picker: ^0.8.7+5
31   flutter_polyline_points: ^1.0.0
32
33   # The following adds the Cupertino Icons font to your application.
34   # Use with the CupertinoIcons class for iOS style icons.
35   cupertino_icons: ^1.0.2
36   provider: ^6.0.5
37   fluttertoast: ^8.2.2
38   url_launcher: ^6.1.11
39   path_provider: ^2.0.15
40   open_file: ^3.3.2
41   flutter_inappwebview: ^5.7.2+3
42   path: ^1.8.3
43   flutter_image_compress: ^2.0.3
44   carousel_slider: ^4.2.1
45   http: ^0.13.4
46   google_maps_flutter: ^2.3.1
```

**Figura 47.** Dependencias usadas en Flutter.

**Elaborado por:** La investigadora

## **B) Storycards**

Las Storycards es una herramienta descriptiva utilizada para representar y gestionar las funcionalidades o requisitos de un proyecto de software con el fin de mantener un seguimiento de las tareas que se deben cumplir mediante el desarrollo de código para satisfacer los requisitos de la historia de usuario.

Se utilizan para estimar prioridades, tiempo de realización, describir la dificultad, así como el esfuerzo que implica cada actividad y añadir comentarios relevantes de las historias.

## Iteración 2: Implementación de la interfaz de inicio o Splash Screen.

### StoryCard 1

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
1	Nuevo	3	3	3	3	Alta
<b>Descripción:</b> Pantalla principal de inicio						
<b>Fecha</b>		<b>Estado</b>		<b>Comentario</b>		
20/05/2023		Definido		Sin comentario		
22/05/2023		Implementado		Sin comentario		
22/05/2023		Ejecutado		Sin comentario		
23/05/2023		Verificado		Sin comentario		

**Tabla 43.** Storycard - Pantalla principal

**Elaborado por:** La investigadora.

En la pantalla de Splash Screen se visualiza brevemente al iniciar la aplicación, antes de cargar su contenido principal, con el objetivo de transmitir al usuario la sensación de que la aplicación se encuentra en proceso de carga activa. Ver Anexo 10.

## Iteración 3: Implementación de la función de inicio de sesión.

### StoryCard 2

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
2	Nuevo	4	4	4	4	Alta
<b>Descripción:</b> Módulo de inicio de sesión						
<b>Fecha</b>		<b>Estado</b>		<b>Comentario</b>		
20/05/2023		Definido		Sin comentario		
22/05/2023		Implementado		Sin comentario		
24/05/2023		Ejecutado		Sin comentario		
25/05/2023		Verificado		Sin comentario		

**Tabla 44.** Registro e inicio de sesión de usuarios.

**Elaborado por:** La investigadora.

El método `_signIn` se usa para manejar la lógica de inicio de sesión de un usuario. En la cual `_formKey.currentState` para asegurar que se hayan cumplido todas las validaciones definidas en el formulario, continuado por la línea de código

“formKey.currentState!.save()”, la cual guarda los valores ingresados en el formulario. En el intento de inicio de sesión: Se llama al método signIn() de la clase authService como se muestra en el fragmento de código. Ver Anexo 8.

En el método signIn de la clase authService maneja la autenticación de usuarios y por medio de la llamada a signInWithEmailAndPassword del objeto \_firebaseAuth en la cual confirma la existencia del usuario en la colección llamada “usuarios”, verifica la existencia de los datos mediante el mapeo en un objeto Usuario y finalmente se establece el objeto Usuario en un proveedor de usuario, devolviendo el objeto userCredential como inicio de sesión exitoso. Ver Anexo 9.

Los campos de email y contraseña se encuentran validados como se muestra en el código del Anexo 10, y si el usuario intenta ingresar sin una contraseña o usuario previo se retornan mensajes de solicitud

El campo email posee un método llamado \_isValidEmail el cual, valida el correo electrónico como el símbolo de arroba, uno o más segmentos de dominio separados por puntos y una extensión de dominio válida de al menos dos caracteres como se detalla en el fragmento de código del Anexo 11.

Una vez que el usuario haya ingresado correctamente los campos como el email y contraseña se limpian como se muestra en el fragmento de código del Anexo 12.

#### **Iteración 4: Implementación de la función de registro de usuario.**

##### **StoryCard 3**

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
3	Nuevo	4	4	4	4	Alta
<b>Descripción:</b> Módulo de registro de usuario						
<b>Fecha</b>		<b>Estado</b>		<b>Comentario</b>		
20/05/2023		Definido		Sin comentario		
27/05/2023		Implementado		Sin comentario		
27/05/2023		Ejecutado		Sin comentario		
29/05/2023		Verificado		Sin comentario		

**Tabla 45.** Módulo de registro de usuario.

**Elaborado por:** La investigadora.

Se importó la librería o paquete de Firebase llamado “firebase\_auth” para usar el servicio “authService” y utilizar métodos para el registro de nuevos usuarios. Luego que el usuario de clic en el botón de registrar se llama al método “registerUserWithEmail” como se muestra en el fragmento de código del Anexo 13.

Luego al redirigirse a la nueva página llamada “auth.service.dart”, donde los datos ingresados por el usuario como el rol el mismo que posee dos opciones, la de estudiante que hace referencia al arrendatario y la de arrendador, continuado por sus nombres, apellidos, correo, celular, contraseña y una confirmación de contraseña ingresada anteriormente. Estos datos se guardan en la colección “usuarios” del Firestore Database como se detalla en el Anexo 16.

Previamente a ser guardados los datos ingresados por el usuario, los mismo son validados con el fin de conservar la integridad de los datos. En el Anexo 17 se detalla el código en el cual se validan los campos: rol, nombre y apellido, así como el Anexo 18 se evidencia la validación del campo celular para permitir el ingreso de 9 número y restringir el uso de letras.

El método \_validateEmail posee un patrón de expresión regular que valida si la cadena cumple con el formato de dirección de correo electrónico, verificando la existencia de caracteres alfanuméricos, guiones, puntos. Mientras que el método \_validatePassword posee un patrón que valida que una cadena tenga al menos una letra mayúscula, una letra minúscula, un dígito numérico y una longitud mínima de 8 caracteres lo cual se detalla en el fragmento de código en el Anexo19.

## Iteración 5: Implementación del módulo de mapas.

### StoryCard 4

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
4	Nuevo	4	4	4	4	Alta
<b>Descripción:</b> Implementación del módulo de mapas.						
<b>Fecha</b>		<b>Estado</b>		<b>Comentario</b>		
20/05/2023		Definido		Sin comentario		
31/05/2023		Implementado		Sin comentario		
03/06/2023		Ejecutado		Sin comentario		
04/06/2023		Verificado		Sin comentario		

**Tabla 46.** Módulo de mapas.

**Elaborado por:** La investigadora.

Para cargar la lista de publicaciones se debe cumplir la condición en la que si el estado de conexión es `ConnectionState.waiting`, se muestra un widget `Center` con un `CircularProgressIndicator` en el centro de la pantalla, caso contrario se muestra un texto indicando que hubo un error al cargar las publicaciones. Ver Anexo 20.

En el fragmento de código que se muestra en el Anexo 21, posee una lista de objetos en `UbicacionModel`, que se utiliza para mostrar marcadores en el mapa y `MisPublicaciones`, que se utiliza para mostrar información relacionada con las publicaciones en el mapa. Con el fin de mostrar el mapa de Google y controlar la visualización de marcadores y polilíneas en el mapa.

## Iteración 6: Implementación del módulo de perfil de usuario.

### StoryCard 5

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
5	Nuevo	4	4	4	4	Alta
<b>Descripción:</b> Implementación del módulo de perfil de usuario.						
<b>Fecha</b>		<b>Estado</b>		<b>Comentario</b>		
20/05/2023		Definido		Sin comentario		
06/06/2023		Implementado		Sin comentario		
07/06/2023		Ejecutado		Sin comentario		
08/06/2023		Verificado		Sin comentario		

**Tabla 47.** Módulo de perfil de usuario.

**Elaborado por:** La investigadora.

En la clase miCuentaSettings se mapean los datos del usuario mediante el valor del usuario del objeto userProvider como: el nombre, apellido, teléfono, correo y rol. De los cuales se le permite editar los campos como: el nombre, apellido y teléfono. Ver Anexo 22.

Los datos previamente a ser actualizados se encuentran controlados mediante validaciones para conservar la integridad de los datos. Una vez que el usuario cambie sus datos y los actualice se llama al método guardarCambios. Ver Anexo 23.

El método guardarCambios actualiza los datos del usuario en el UserProvider y guarda los cambios en el backend que es el Firestore Database mediante el método updateUser que se encuentra en la clase UserService. Ver Anexo 24.

El método updateUser se utiliza para actualizar los datos de un usuario en una base de datos Firestore, asegurándose de que haya un usuario autenticado antes de realizar la actualización mediante el uid. Ver Anexo 25.



## Iteración 7: Implementación de la función publicar arriendos.

### StoryCard 6

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
6	Nuevo	4	4	4	4	Alta
<b>Descripción:</b> Implementación de la función de publicar arriendos.						
<b>Fecha</b>		<b>Estado</b>		<b>Comentario</b>		
20/05/2023		Definido		Sin comentario		
09/06/2023		Implementado		Sin comentario		
11/06/2023		Ejecutado		Sin comentario		
14/06/2023		Verificado		Sin comentario		

**Tabla 48.** Función publicar arriendos.

**Elaborado por:** La investigadora.

Cada uno de los archivos dart que posee la carpeta publicar almacena el paso 1, el paso 2 y el paso 3, las cuales son características del cuarto de arriendo a publicar, divididas en un stepper. Ver Anexo 26.

### Paso 1

En el paso 1 el arrendador puede brindar detalles del cuarto de arriendo como: tamaño, precio, selección de ubicación, si es amoblado o no, con cocina compartida o individual, luego adjuntar el PDF de contrato de arriendo y adjuntar imágenes, incluidas sus validaciones. Ver Anexo 27.

Continuando en el paso 1 en donde le permite añadir la ubicación del cuarto de arriendo llamando a la clase MapaPublicar el cual contiene el mapa leaflet y es necesario la importación de la librería latlong2. En la variable `_selectedLocation.latitude` y `_selectedLocation.longitude` guarda la ubicación actual que el usuario seleccione en el mapa. Ver Anexo 28.

En la clase MapaPublicar es necesario importar los paquetes `flutter_map` y `latlong2`, para hacer uso de la clase `MapOptions` que permiten controlar el comportamiento y la apariencia del mapa. En la variable `_markerLocation` se inicializa con valores estáticos

que representan una ubicación de referencia relacionada con la UTA, los cuales pueden ser editados mediante la selección de la ubicación en el mapa o el arrastre del marcador. Ver Anexo 29.

En la función Navigator.pop se envía los datos de ubicación seleccionados, por medio de la variable selectedLocation. Ver Anexo 30.

El arrendador puede habilitar por medio del CheckboxListTile si su habitación es amoblada o no y por medio de un RadioListTile para seleccionar si la cocina es individual o compartida. Ver Anexo 31.

Para permitir al usuario seleccionar un archivo PDF es necesario el paquete "file\_picker". En el archivo result.files.single.path retorna la referencia del archivo, mientras que result.files.single.name obtiene el nombre del resultado. Luego en setState se actualiza la referencia y nombre del archivo a publicacionProvider.publicacion.contrato. Esto permite el almacenamiento de los datos del archivo que será posteriormente utilizados en la aplicación. Ver Anexo 32.

La clase "Contrato", almacena la URL del contrato si está disponible, el archivo del contrato y nombre del archivo. También incluye método para convertir la instancia de "Contrato" en JSON. Ver Anexo 33.

Para permitir al usuario seleccionar imágenes de la galería es necesario el paquete "image\_picker". Se verifica se ya se han adjuntado 4 imágenes. Si es así se muestra un dialogo indicando que se ha alcanzado el límite de imágenes. La imagen seleccionada se almacena en la variable image de tipo XFile. Una vez seleccionado la imagen se actualiza el estado setState para agregar al listado images y se asigna el listado actualizado a publicacionProvider.publicacion.imagenes. Ver Anexo 34.

El método guardarDatosPaso1, toma un objeto "PublicacionProvider" como parámetro. Con el fin de guardar los datos del primer paso de la publicación en el objeto Publicacion y luego los actualiza en el publication Provider. Ver Anexo 35.

## **Paso 2**

Se crea una clase llamada `PublicacionProvider` que extiende `ChangeNotifier`. Esta clase se utiliza como un proveedor de datos para la gestión de una publicación, es decir esta clase puede notificar a los widgets que dependen de ella cuando los datos cambian. Se define un método `actualizarPublicacion` el cual actualiza la variable `_publicacion` con la nueva instancia recibida y notifica a los listeners que los datos han cambiado. Ver Anexo 36.

Se instancia publicación de `publicacionProvider.publicacion`. Luego se crean los Servicios básicos como: internet, luz, agua y parqueadero dentro de varios `CheckboxListTile`, con valores booleanos (`value`) que indica si el servicio esta seleccionado o no, y se actualiza mediante la función `onChanged`. Ver Anexo 37.

Para representar los meses de garantía del cuarto de arriendo se usa varios `RadioListTile` con un valor (`value`) que representa el valor de opción y `groupValue` que indica la opción seleccionada actualmente. El estado de selección se maneja mediante la variable de estado `mesesGarantia` y se actualiza mediante la función `onChanged`. Ver Anexo 38.

En la clase Paso 2 se encuentra el método estático `guardarDatosPaso2` en la clase `PublicacionProvider`. Este método se utiliza para guardar los datos correspondientes al segundo paso de la creación de una publicación. Ver Anexo 39.

## **Paso 3**

El fragmento de código en la Figura 74, define la clase `Paso3`. Dentro del `Paso3State`, se definen varias variables estáticas, incluyendo, `preferenciaGenero`, se usa tres `RadioListTile` que representan las opciones de preferencia de género: "Masculino", "Femenino" y "Sin preferencia de género". Cada uno tiene un título y un valor asociado. Ver Anexo 40.

El valor de la casilla de verificación se establece en la variable `permitirMascotas`. Cuando se produce un cambio en el estado de la casilla de verificación, se llama al

método `onChanged`. Luego en el `TextField` representa un campo de texto editable. El parámetro `maxLines` se establece en `null` para permitir múltiples líneas de texto en el campo. Ver Anexo 41.

El método `guardarDatosPaso3` guarda los datos del tercer paso de la creación de una publicación en la instancia de `PublicacionesModelo` y actualiza esta instancia en `publicacionProvider` utilizando el servicio de publicaciones. Ver Anexo 42.

### **Implementación de los 3 pasos para publicar el arriendo.**

Se importan las clases y paquetes necesarios para el funcionamiento de la página, como modelos, proveedores y servicios. Ver Anexo 43.

En la clase `PasosPage` se declaran las variables `_currentStep` para realizar un seguimiento del paso actual en el que se encuentra el usuario, mientras que `isStepComplete` indica si el paso actual se ha completado correctamente. Ver Anexo 44.

El widget `Consumer<PublicacionProvider>`, que permite acceder y consumir el estado proporcionado por `PublicacionProvider`. Se utiliza el widget `Stepper` para mostrar los pasos en la página. Cada paso se representa como un objeto `Step`, que contiene un título, un estado (activo o no), y un contenido asociado (en este caso, los widgets `Paso1`, `Paso2` y `Paso3`). Ver Anexo 45.

Dentro de los manejadores de eventos de los botones, se realizan diversas acciones, como guardar datos ingresados en los pasos, llamar a servicios para subir imágenes y PDFs, crear una publicación, limpiar los campos después de una publicación exitosa y mostrar un mensaje de éxito.

El botón `Guardar` maneja el evento `onPressed` obtiene el `uid` del usuario actual, guardar los datos del paso 3, llamar a servicios para subir imágenes y PDF, crear una publicación, limpiar los campos y mostrar un mensaje de éxito. Ver Anexo 46.

La función `_limpiarCampos()` se encarga de restablecer o limpiar los campos y variables utilizados en los diferentes pasos del formulario. Ver Anexo 47.

La función `_mostrarRegistroExitoso()` muestra un mensaje de éxito a través de un `SnackBar` y luego navega a la página de inicio (`HomePage`) después de que el mensaje desaparezca. Esto brinda retroalimentación al usuario sobre el registro exitoso y redirige al usuario a la página principal. Ver Anexo 48.

**Iteración 8: Implementación del módulo para obtener listado de arriendos publicados opción propietarios.**

**StoryCard 7**

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
7	Nuevo	4	4	4	4	Alta
<b>Descripción:</b> Implementación del módulo para obtener el listado de arriendos publicados opción propietarios.						
<b>Fecha</b>		<b>Estado</b>		<b>Comentario</b>		
20/05/2023		Definido		Sin comentario		
15/06/2023		Implementado		Sin comentario		
15/06/2023		Ejecutado		Sin comentario		
15/06/2023		Verificado		Sin comentario		

**Tabla 49.** Listado de arriendos publicados opción propietarios.

**Elaborado por:** La investigadora.

En el fragmento de código del Anexo 49, se obtiene el ID del usuario actual, mediante el método `getPublicacionesUsuario` se obtiene las publicaciones asociadas al usuario enviado como argumento y actualiza el estado de la aplicación con las publicaciones obtenidas.

Por medio del argumento `uidUsuario` se accede a la colección donde se almacenan las publicaciones. Mapea los datos obtenidos a objetos `MisPublicaciones` y retorna una lista de dichas publicaciones. Ver Anexo 50.

**Iteración 9: Implementación del módulo borrar y editar arriendos de la lista del arrendador.**

**StoryCard 8**

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
8	Nuevo	4	4	4	4	Alta
<b>Descripción:</b> Implementación del módulo borrar y editar arriendos de la lista del arrendador.						
<b>Fecha</b>		<b>Estado</b>		<b>Comentario</b>		
20/05/2023		Definido		Sin comentario		
16/06/2023		Implementado		Sin comentario		
17/06/2023		Ejecutado		Sin comentario		
21/06/2023		Verificado		Sin comentario		

**Tabla 50.** Módulo borrar y editar arriendos.

**Elaborado por:** La investigadora.

El código del Anexo 51 detalla un diálogo de confirmación con dos botones: "Cancelar" y "Eliminar". Si el usuario elige "Eliminar", se llama al servicio `userService.deletePublicacionesUsuario(idPublicacion)`.

El método `deletePublicacionesUsuario` elimina una publicación de un usuario específico. Mediante una referencia a la colección `publicaciones` en Firestore utilizando `Firestore.instance.collection('publicaciones')`. Se obtiene la referencia del documento de la publicación específica, luego verifica si el documento existe, se procede a eliminar la publicación. Ver Anexo 52.

El usuario administrador tiene la funcionalidad de editar las publicaciones realizadas, tales como el tamaño, precio, seleccionar la ubicación en el mapa, la disponibilidad de cuarto amoblado, cocina individual o compartida, actualizar el contrato de arriendo en formato PDF, adjuntar hasta 4 imágenes, habilitar o deshabilitar los servicios básicos como el internet, agua, luz, parqueadero, los meses de garantía, la preferencia de género, el permiso de mascotas y una descripción del cuarto de arriendo.

En el Anexo 53 se indica los métodos que se crearon para la edición de los campos como la actualización de imágenes y PDF.

El código construye una pantalla de edición de publicación con campos para el tamaño, precio y actualizar un archivo PDF mediante el método `abrirPDF` y `_actualizarPDF`. Ver Anexo 54.

En la función `abrirPDF` verifica si se ha actualizado el PDF y si hay una URL válida para el archivo PDF. Si `urlPDF` está vacío, se muestra un cuadro de diálogo (`AlertDialog`) en la interfaz de usuario que indica que no se puede abrir el PDF. Ver Anexo 55.

Este código define una función asincrónica llamada `_actualizarPDF()`, que se utiliza para actualizar el archivo PDF que se muestra en la pantalla y guardar la información relacionada con el nuevo archivo PDF seleccionado. Ver Anexo 56.

Otros campos como si es amoblado, o si posee cocina individual o compartida, además se puede actualizar la ubicación del cuarto de arriendo. Ver Anexo 57.

Para editar las imágenes del cuarto de arriendo se crea el método `_agregarImagen`, con una instancia de `ImagePicker`, que es una clase utilizada para seleccionar imágenes de diferentes fuentes, como la galería de fotos o la cámara del dispositivo. Además, verifica si se ha alcanzado el límite de 4 imágenes adjuntas y muestra un cuadro de diálogo si se ha alcanzado el límite.

Si la imagen ha sido agregada se llama al método `comprimirImagen` de `publicacionService`, pasando como argumento el archivo de la imagen seleccionada. Ver Anexo 58.

El método `comprimirImagen` toma una imagen como entrada, la comprime utilizando la biblioteca `flutter_image_compress`, y devuelve un objeto `XFile` que representa la imagen comprimida. Ver Anexo 59.

**Iteración 10: Implementación de la funcionalidad de visualización rápida de detalles del arriendo en el mapa.**

**StoryCard 9**

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
9	Nuevo	4	4	4	4	Alta
<b>Descripción:</b> Implementación de la funcionalidad de visualización rápida de detalles del arriendo en el mapa.						
<b>Fecha</b>		<b>Estado</b>		<b>Comentario</b>		
20/05/2023		Definido		Sin comentario		
22/06/2023		Implementado		Sin comentario		
22/06/2023		Ejecutado		Sin comentario		
25/06/2023		Verificado		Sin comentario		

**Tabla 51.** Visualización rápida de detalles del arriendo en el mapa.

**Elaborado por:** La investigadora.

El contenido del diálogo es un AlertDialog, que muestra información de una publicación. El contenido incluye una imagen del cuarto de arriendos, una descripción, el precio y tamaño de la publicación. Ver Anexo 60.

Dentro del ShowDialog se encuentran tres acciones: un icono de envío, un icono de información y un botón para cerrar el dialogo actual. El icono de envío, realiza una acción asincrónica para obtener las coordenadas de ubicación actual y las coordenadas de ubicación seleccionadas en la publicación, el icono de información, cierra el diálogo actual y navega a otra pantalla llamada "reconstruccion\_publicacion", la cual describe los detalles del departamento. El botón de "Cerrar", cierra el diálogo actual y restablece una variable de estado llamada "selectedMarker". Ver Anexo 61.



**Iteración 11: Implementación de la funcionalidad de arriendos favoritos y filtros de usuario.**

**StoryCard 10**

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
10	Nuevo	4	4	4	4	Alta
<b>Descripción:</b> Implementación de la funcionalidad de arriendos favoritos y filtros de usuario.						
<b>Fecha</b>		<b>Estado</b>		<b>Comentario</b>		
20/05/2023		Definido		Sin comentario		
26/06/2023		Implementado		Sin comentario		
27/06/2023		Ejecutado		Sin comentario		
29/06/2023		Verificado		Sin comentario		

**Tabla 52.** Funcionalidad de arriendos favoritos y filtros.

**Elaborado por:** La investigadora.

La función `obtenerMisFavoritos` obtiene la lista de publicaciones favoritas de un usuario específico y actualiza el estado del widget con los resultados obtenidos. Ver Anexo 62.

La función `eliminarDeMisFavoritos` busca un documento con el ID especificado en la colección "misFavoritos" dentro de cada documento en la colección "usuarios". Si encuentra un documento coincidente, lo elimina y devuelve true. Si no encuentra ningún documento coincidente o si ocurre un error, devuelve false. Ver Anexo 63.

La clase `FilterModel` mantiene el estado de los filtros seleccionados y notifica a los observadores cuando cambian los filtros. Esto permite que otros componentes de la aplicación se actualicen y respondan a los cambios en los filtros. Ver Anexo 64.

## Iteración 12: Implementación de la función de geolocalización.

### StoryCard 11

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
11	Nuevo	4	4	4	4	Alta
<b>Descripción:</b> Implementación de la función de geolocalización.						
<b>Fecha</b>		<b>Estado</b>		<b>Comentario</b>		
20/05/2023		Definido		Sin comentario		
30/06/2023		Implementado		Sin comentario		
01/07/2023		Ejecutado		Sin comentario		
04/07/2023		Verificado		Sin comentario		

**Tabla 53.** Implementación de la función de geolocalización.

**Elaborado por:** La investigadora.

La función llamada `getDirections` utiliza la API de Google Maps Directions para obtener las coordenadas de una ruta entre un origen y un destino. Las coordenadas se devuelven en forma de una lista de objetos `LatLng`. Ver Anexo 65.

La función `obtenerUbicacionActual` utiliza la biblioteca `geolocator` para obtener la ubicación actual del dispositivo. Si el servicio de ubicación está activo, se devuelve un objeto `LatLng` con las coordenadas de la ubicación actual. Si el servicio de ubicación no está activo, se devuelve un objeto `LatLng` con coordenadas nulas (0, 0). Ver Anexo 66.

La función `agregarMarcadores` recibe una lista de objetos de tipo `MisPublicaciones` y agrega marcadores en un mapa utilizando la biblioteca `google_maps_flutter`. Se verifica si cada publicación tiene una ubicación válida y, en caso afirmativo, se crea un marcador en esa ubicación con un identificador único y una función de `onTap` asociada. Los marcadores se almacenan en una lista `_markers` para su uso posterior en el mapa. Ver Anexo 67.

### 3.2.4 Fase 4: Estabilización

Esta fase se enfoca en corregir errores, ajustar el diseño y la experiencia de usuario para que la aplicación sea confiable.

Modificaciones en el diseño:

- Se reemplazó el mapa Leaflet por el mapa de Google Maps debido a la falta de precisión al trazar rutas utilizando el API de Google en Leaflet.

Modificaciones en la base de datos:

- Incorporación del campo booleano habilitado en las publicaciones, para que el arrendador habilite o deshabilite la visualización del arriendo cuando este disponible.
- Incorporación del campo comentarios en la lista de publicaciones que se muestran al arrendatario.

Modificaciones en el código:

- Se agregó un método para minimizar la calidad de las imágenes agregadas por el arrendador, con el fin de minimizar el consumo de recursos en el Storage de Firebase.

### **3.2.5 Fase 5: Pruebas del sistema**

Esta fase se enfoca en realizar pruebas a la aplicación móvil con el fin de optimizar el rendimiento, ajustar el diseño y la experiencia de usuario, brindando así una versión estable y funcional.

#### **A) Pruebas de funcionalidad**

Esta prueba es fundamental para garantizar que la aplicación móvil cumpla con los requisitos funcionales establecidos.

- **Pantalla principal o Splash Screen**

El Splash Screen se muestra brevemente al iniciar la aplicación antes de cargar el contenido principal, con el objetivo de brindar una experiencia visual agradable, como se indica en la **Figura 48**.



**Figura 48.** Splash Screen.

**Elaborado por:** La investigadora.

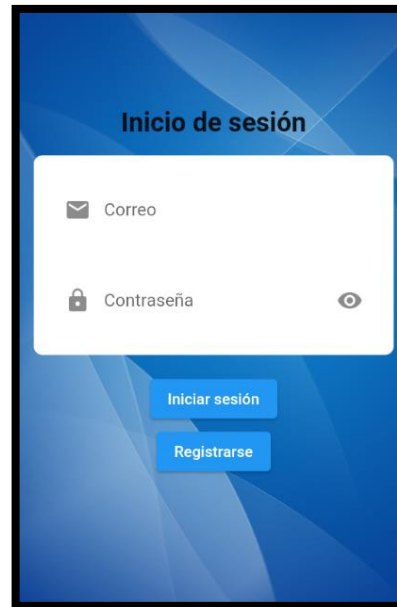
- **Registro y login de usuarios**

En esta sección, los usuarios pueden crear una cuenta en la aplicación móvil proporcionando la información necesaria, como el rol (arrendador, estudiante), el nombre, apellido, celular, correo, contraseña, confirmación de contraseña como se detalla en la **Figura 49**.

**Figura 49.** Pantalla para registro de usuario.

**Elaborado por:** La investigadora.

Una vez que los usuarios ya tienen una cuenta pueden ingresar a la aplicación móvil utilizando su correo electrónico y contraseña, como se muestra en la **Figura 50**.



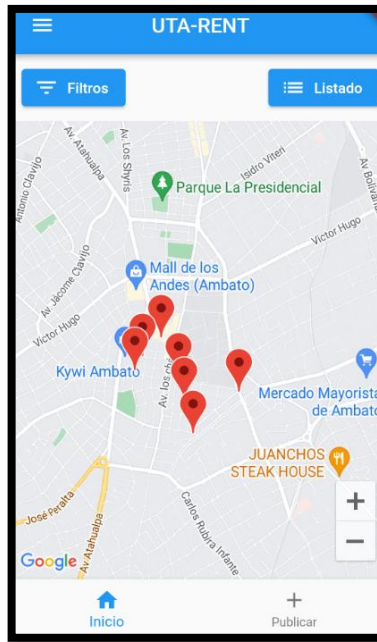
**Figura 50.** Pantalla para inicio de sesión.

**Elaborado por:** La investigadora.

### **Usuario Arrendador**

- **Ubicación y detalle de la lista de arriendos en el mapa**

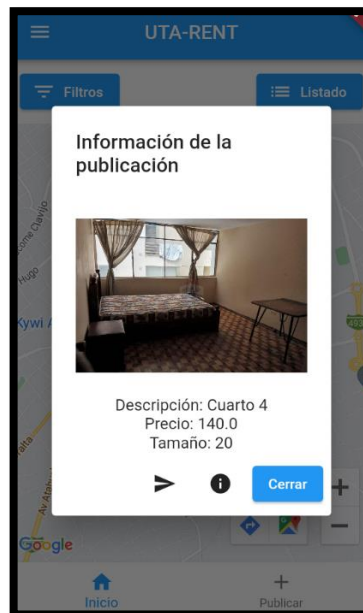
Esta funcionalidad despliega el mapa, con marcadores que indican las ubicaciones de las propiedades disponibles para arrendar, como se indica en la **Figura 51**.



**Figura 51.** Marcadores en la lista de arriendos.

**Elaborado por:** La investigadora.

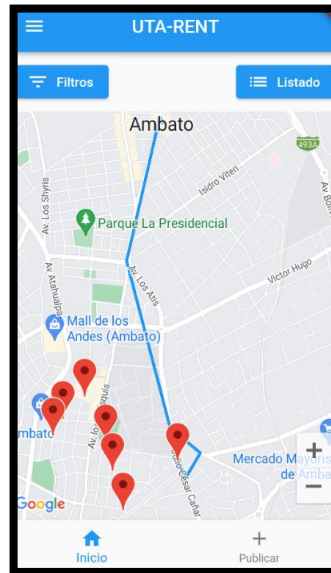
Al hacer clic en cada marcador, redirige al arrendador a la publicación detallada del arrendador correspondiente a esa propiedad en particular, como se indica en la **Figura 52**.



**Figura 52.** Detalle de la publicación.

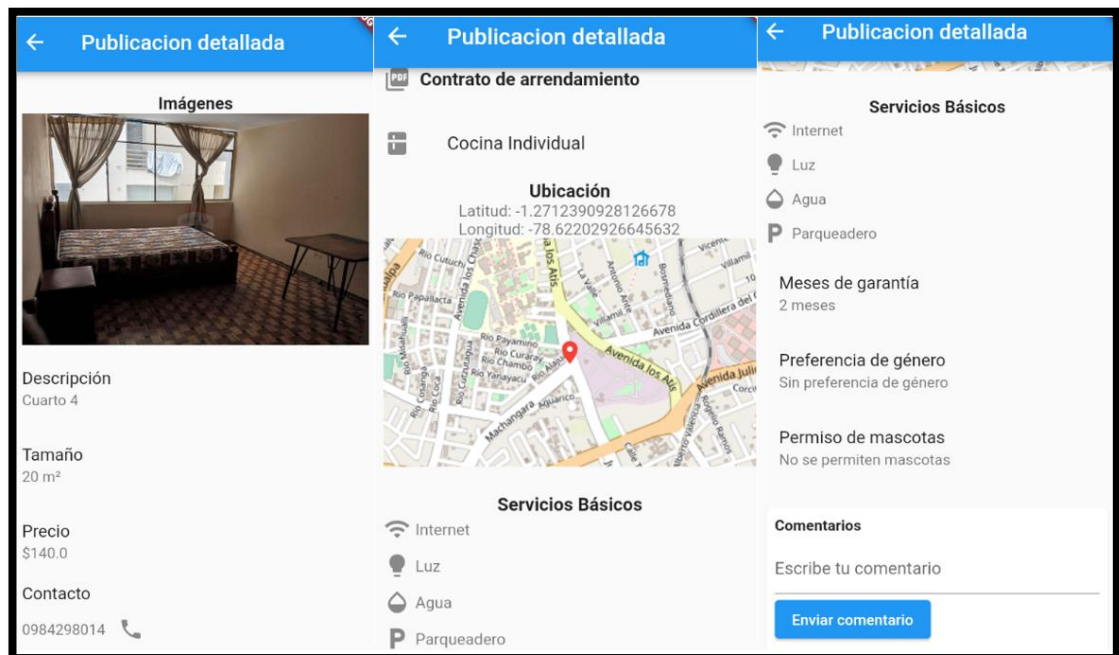
**Elaborado por:** La investigadora.

Además, brinda al usuario la opción de activar la ruta hacia el arriendo desde su ubicación actual como indica la **Figura 53**, u obtener una pantalla de información detallada, como se indica en el **Figura 55**.



**Figura 53.** Trazo de la ruta.

**Elaborado por:** La investigadora.

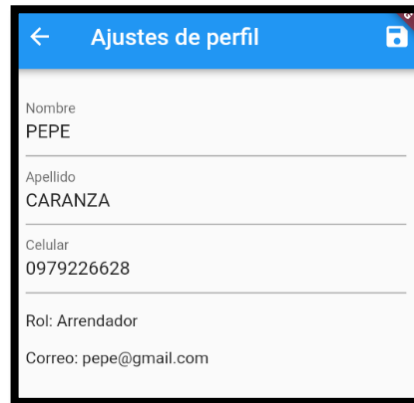


**Figura 54.** Publicación detallada.

**Elaborado por:** La investigadora.

- **Perfil de usuario**

El usuario administrador puede editar su información personal, como nombres, apellidos y número de celular. Esto le permite mantener actualizados sus datos de contacto, lo cual presentará información precisa y confiable para los demás usuarios, como se indica en la **Figura 55**.



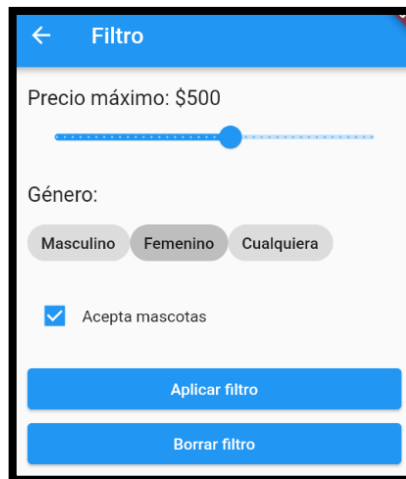
**Figura 55.** Ajustes de perfil.

**Elaborado por:** La investigadora.

- **Filtros para la lista de arriendos**

La posibilidad de agregar filtros permite a los usuarios refinar su búsqueda de propiedades de alquiler. Algunos de los filtros disponibles incluyen: precio máximo, preferencia de género, y permiso para la presencia de mascotas, como se indica en la **Figura 56**.





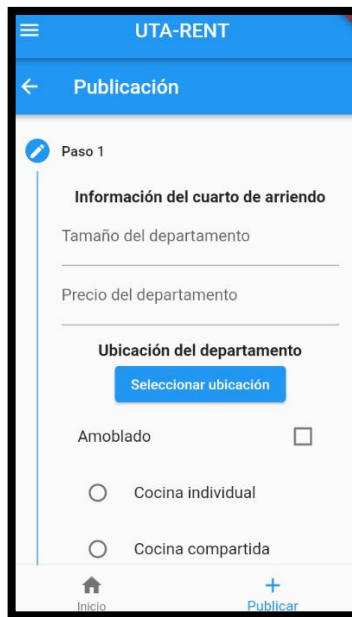
**Figura 56.** Filtros de búsqueda.

**Elaborado por:** La investigadora.

- **Publicación de arriendos**

Los usuarios tienen la posibilidad de publicar sus propios anuncios de alquiler detallando las características específicas de la propiedad. Al crear una publicación, los usuarios pueden incluir la siguiente información:

- **Tamaño:** se puede proporcionar el tamaño del cuarto en metros cuadrados, lo que ayuda a los arrendatarios a tener una idea clara del espacio disponible.
- **Precio:** se puede especificar el precio mensual de alquiler, para que los interesados coticen de acuerdo con su presupuesto.
- **Ubicación en el mapa de Google maps:** los arrendadores pueden agregar la ubicación exacta de la propiedad, ofreciendo a los posibles arrendatarios una referencia geográfica precisa.
- **Amoblada:** indica si la propiedad posee bienes para uso del arrendatario.
- **Cocina:** los arrendadores pueden especificar si la cocina es individual o compartida con otros inquilinos.



**Figura 57.** Paso 1, precio, tamaño, ubicación, amoblado, cocina.

**Elaborado por:** La investigadora.

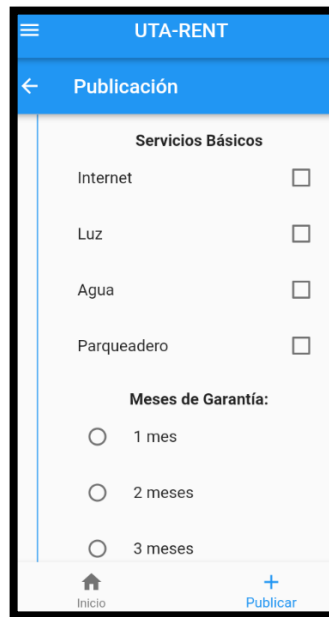
- Contrato de arriendo: se puede agregar el contrato para que los interesados puedan revisar las normas y reglas que se debe cumplir.
- Imágenes: se puede incluir hasta 4 imágenes para ofrecer una idea visual del espacio de arriendo.



**Figura 58.** Paso 1, adjuntar pdf e imágenes.

**Elaborado por:** La investigadora.

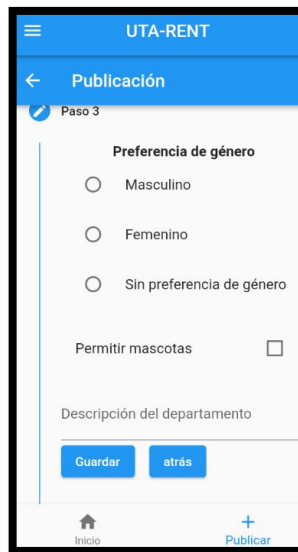
- Servicios básicos: se especifican los servicios básicos que están incluidos, como Internet, agua, luz y si hay disponibilidad de parqueadero.
- Meses de garantía: Se puede especificar la cantidad de meses de garantía que se requieren al momento de alquilar la propiedad.



**Figura 59.** Paso 2, servicios básicos y garantía.

**Elaborado por:** La investigadora.

- Género: los arrendadores pueden especificar si tienen preferencia de género para arrendar el cuarto.
- Descripción: Se puede proporcionar una breve descripción del cuarto, destacando características como la iluminación, la dirección o más.

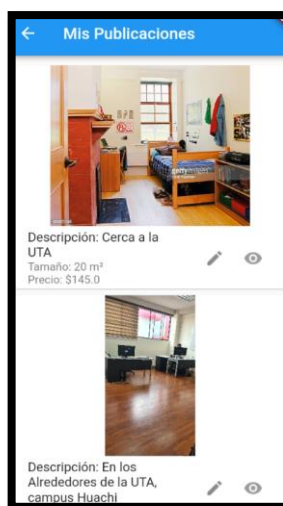


**Figura 60.** Paso 3, género, permiso mascotas, descripción.

**Elaborado por:** La investigadora.

- **Edición y habilitación de arriendos**

Las publicaciones realizadas por el arrendador se añaden a una lista. Estas publicaciones pueden ser editadas y habilitadas según las necesidades del arrendador. Una vez habilitadas, estas publicaciones se mostrarán visualmente en el mapa, lo que permitirá a los usuarios tener una visión general de todas las propiedades disponibles en la zona.



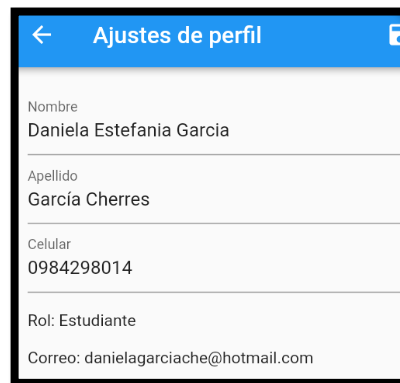
**Figura 61.** Edición y habilitación de arriendos.

**Elaborado por:** La investigadora.

## Usuario estudiante o arrendatario

- **Perfil de usuario**

El usuario arrendatario (estudiante), puede editar su información personal, como nombres, apellidos y número de celular. Esto le permite mantener actualizados sus datos de contacto, lo cual presentará información precisa y confiable para los demás usuarios, como se indica en la **Figura 62**.



**Figura 62.** Ajustes perfil arrendatario.

**Elaborado por:** La investigadora.

- **Lista de Favoritos**

El usuario estudiante tiene la opción de crear una lista de arriendos favoritos, lo que le brinda facilidad de acceso y organización, así como eliminarlos de la lista, como se indica en la **Figura 63**.

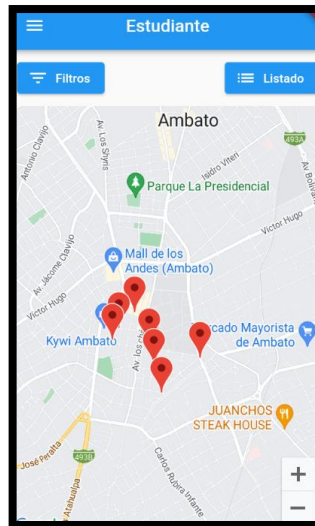


**Figura 63.** Lista favoritos.

**Elaborado por:** La investigadora.

- **Ubicación y detalle de la lista de arriendos en el mapa**

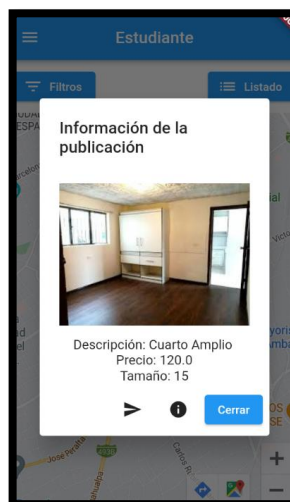
Esta funcionalidad despliega el mapa, con marcadores que indican las ubicaciones de las propiedades disponibles para arrendar, como se indica en la **Figura 64**.



**Figura 64.** Lista de arriendos en el mapa.

**Elaborado por:** La investigadora.

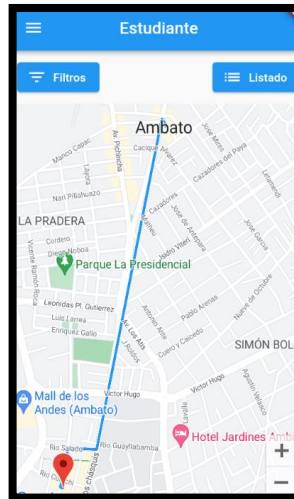
Al hacer clic en cada marcador, redirige al arrendatario (estudiante) a la publicación detallada del arrendador correspondiente a esa propiedad en particular, como se indica en la **Figura 65**.



**Figura 65.** Detalle de la publicación.

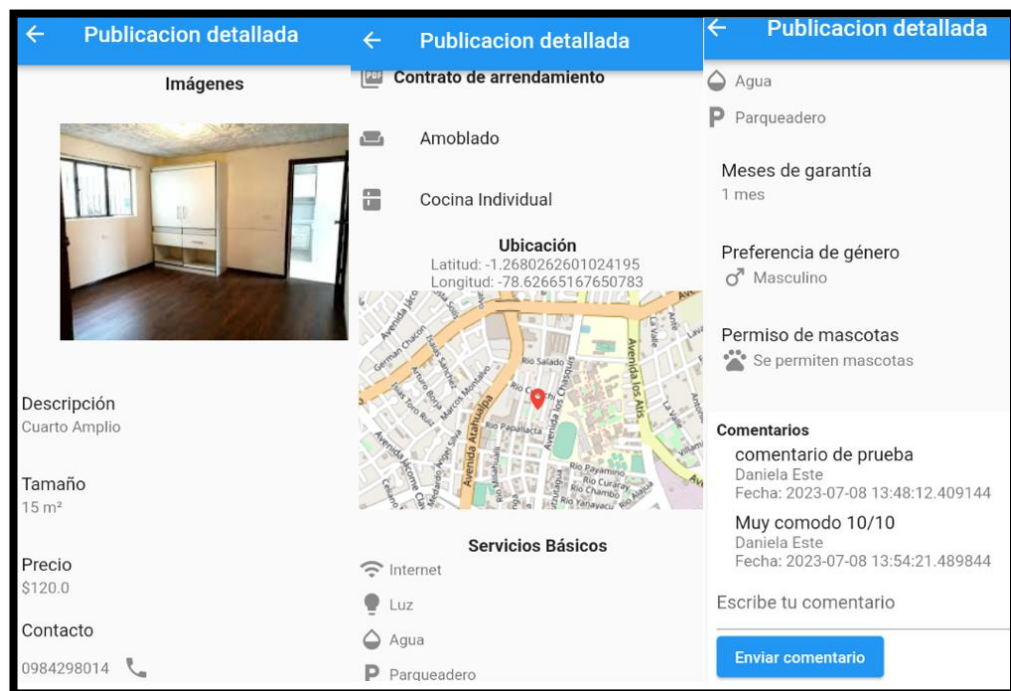
**Elaborado por:** La investigadora.

Además, brinda al estudiante la opción de activar la ruta hacia el arriendo desde su ubicación actual como indica la **Figura 66**, u obtener una pantalla de información detallada, como se indica en el **Figura 67**, en la cual el estudiante puede añadir comentarios relacionados con la experiencia que haya vivido en el arriendo.



**Figura 66.** Trazo de la ruta hacia los arriendos.

**Elaborado por:** La investigadora.

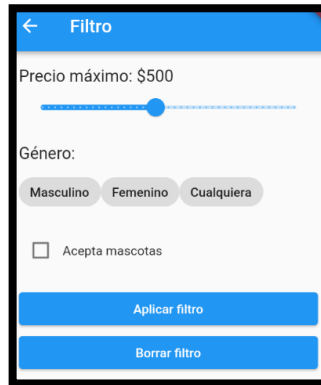


**Figura 67.** Publicación detallada del arriendo.

**Elaborado por:** La investigadora.

- **Filtros para la lista de arriendos**

El estudiante tiene la posibilidad de agregar filtros para la búsqueda de propiedades de alquiler. Algunos de los filtros disponibles incluyen: precio máximo, preferencia de género, y permiso para la presencia de mascotas, como se indica en la **Figura 68**.

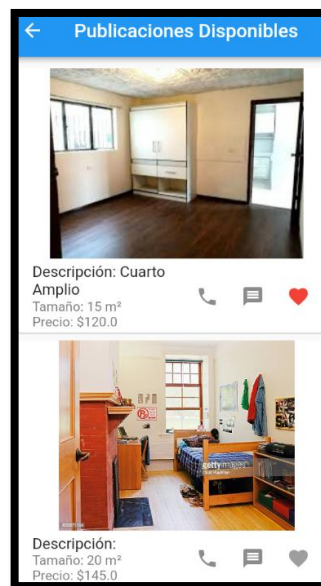


**Figura 68.** Filtro de la publicación.

**Elaborado por:** La investigadora.

- **Visualización de la lista de arriendos**

Muestra todas las publicaciones realizadas por los arrendadores, al dar clic sobre las imágenes se muestra a detalle la publicación, como se indica la **Figura 69**.

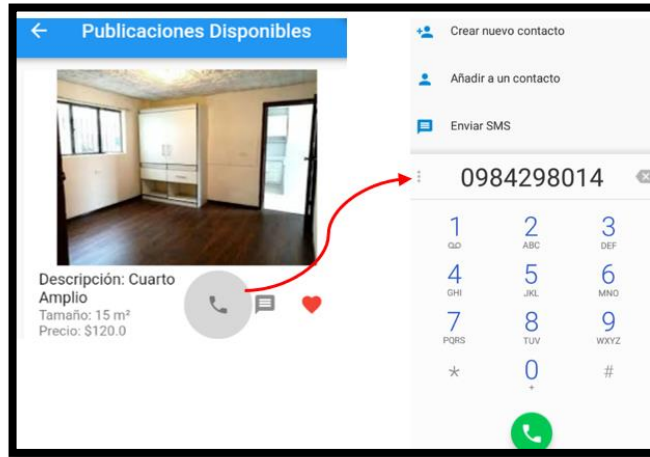


**Figura 69.** Lista de arriendos disponibles.

**Elaborado por:** La investigadora.



Además, se proporciona los detalles de contacto del arrendador, como su número de teléfono o cuenta de WhatsApp para solicitar información de interés, como se muestra en la **Figura 70**.



**Figura 70.** Detalles del contacto del arrendador.

**Elaborado por:** La investigadora.

## **CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES**

### **4.1 Conclusiones**

- Durante el proceso de recolección de la información existieron un número de usuarios que presentaron cierta incertidumbre al momento de entregar sus datos. Esto se debe a diversos fenómenos sociales relacionados con la seguridad a los que el país se enfrenta en la actualidad, así como al tipo de extorsiones. Bajo esta perspectiva el periodo de recopilación de datos para la presente investigación se amplió en un promedio de 1 semana adicional.
- Mediante la generación de un mapa de procesos de las actividades tanto de los arrendadores para la difusión de arriendos como de los arrendatarios para la búsqueda, se obtuvo una visión amplia y clara para establecer funcionalidades y características que la aplicación móvil proporcionaría, ajustándose a las necesidades de la sociedad.
- Entre las diferentes tecnologías multiplataforma encontradas durante la investigación, se seleccionó a Flutter como la más adecuada para el desarrollo de la aplicación móvil, en base a características como la baja curva de aprendizaje y la amplia gama de widgets personalizables, lo cual aporta en gran medida a la experiencia de usuario facilitando la creación de interfaces, sin dejar de lado la eficiencia del rendimiento de la aplicación en los dispositivos.
- La metodología Mobile-D aporta de manera significativa el desarrollo de la aplicación móvil, ya que ha permitido un seguimiento y control de las tareas mediante la división del proyecto en iteraciones, con entregas funcionales en el proceso de desarrollo, logrando una gestión eficiente con la entrega oportuna de la aplicación.
- La aplicación móvil se desempeña como una plataforma de acceso a los servicios requeridos por estudiantes universitarios para encontrar un departamento o cuarto de alquiler conforme a sus necesidades.

## 4.2 Recomendaciones

- Buscar espacios de retroalimentación para informar a las personas sobre las ventajas de las plataformas digitales y como pueden apoyar el proceso de búsqueda y difusión de arriendos.
- Dentro del proceso de mejoramiento continuo de las herramientas de software, se recomienda una reingeniería de los procesos con el objetivo de optimizar tiempos entre los flujos de trabajo e interacción de actividades entre arrendador y arrendatario.
- Mantener las dependencias actualizadas en el archivo pubspec.yaml del proyecto Flutter, con el fin de evitar posibles conflictos y problemas de compatibilidad entre los distintos paquetes utilizados
- Para proyectos futuros de desarrollo móvil, se recomienda utilizar la metodología Mobile-D debido a su efectividad en términos de productividad, calidad del producto final y capacidad para mantener un orden de desarrollo mediante iteraciones con tareas claramente detalladas.
- Para garantizar un rendimiento óptimo y gestión adecuada de los recursos de Firebase, es importante analizar los planes de suscripción y restricciones asociadas para actualizar a un plan de pago que se ajuste a las necesidades y requerimientos del proyecto.

## Bibliografía

- [1] C. Román, “El uso del celular y su influencia en las actividades académicas y familiares de los estudiantes de primer año de bachillerato de la Unidad Educativa Sagrados Corazones de Rumipamba de la ciudad de Quito,” 2017.
- [2] Miguel Zuniga, “Universidad Técnica De Babahoyo Examen Complexivo De Grado O De Fin De Carrera Prueba Práctica Ingeniería En Sistemas Previo a La Obtención Del Título De Ingeniero (a) En Sistemas,” 2022.
- [3] E. F. Flores Romero, “COVID-19 y comunicación de salud: Análisis de medios digitales ecuatorianos,” *Rev. Sci.*, vol. 6, no. 19, pp. 122–141, 2021, doi: 10.29394/scientific.issn.2542-2987.2021.6.19.6.122-141.
- [4] J. E. Torres Perdigón, “Homing: aplicación móvil para servicios de arrendamiento a través de un nuevo modelo para la búsqueda y administración de arriendos en Bogotá,” Bogotá, 2019.
- [5] P. Contreras, “Qué buscan los estudiantes al arrendar una vivienda – Publimetro Chile,” 2017. <https://www.publimetro.cl/cl/noticias/2017/03/20/buscan->

estudiantes-arrendar-vivienda.html (accessed Nov. 05, 2022).

- [6] P. J. Alcázar Ponce, “Ecuador Estado Digital,” *Mentimno – Innov. Lifetime Value Partners*, p. 37, 2021.
- [7] INEC, “Uso de tecnología en Ecuador,” 2011. [Online]. Available: <http://www.inec.gob.ec/estadisticas/>
- [8] M. Arias Pérez, M. Bastidas Ramos, and C. Salazar Mejía, “Estudio sobre la deserción estudiantil universitaria y sus implicaciones académicas, económicas y sociales,” *Bolentín Coyunt.*, vol. 1, no. 19, p. 9, 2018, doi: 10.31164/bcoyu.19.2018.677.
- [9] F. D. E. C. Administrativas *et al.*, “Mobile marketing como estrategia de comunicación en la política 2.0,” Universidad Técnica de Ambato, Ambato, 2022.
- [10] D. Reinoso and M. Quisimalín, “Plataformas Digitales: Diagnóstico de Aceptación del Comercio Electrónico en Pymes - Zona Urbana De Ambato,” *Dominio las Ciencias*, vol. 8, no. 1, pp. 734–749, 2022, [Online]. Available: <http://dx.doi.org/10.23857/dc.v8i41.2520>
- [11] M. E. Burbano Santamaría, “Aplicación móvil multiplataforma para la gestión de servicios de publicidad y delivery de la empresa FM multiservicios,” vol. 3, no. 2, p. 6, 2021.
- [12] W. G. M. Guajan, “Desarrollo de una aplicación móvil para el control de acceso a conjuntos residenciales mediante bluetooth utilizando framework Xamarin Forms,” Universidad Técnica del Norte, 2022.
- [13] “Ingeniería de Software - Guillermo Pantaleo, Ludmila Rinaudo - Google Libros.” <https://books.google.es/books?hl=es&lr=&id=rjxyEAAAQBAJ&oi=fnd&pg=PR3&dq=ingenieria+de+software&ots=5kIDqXbF-O&sig=IS4aZkaWCDdg360R2pIDc0gD484#v=onepage&q&f=false> (accessed Nov. 23, 2022).
- [14] “Introducción a la ingeniería del software - Fernando Alonso Amo, Loïc A. Martínez Normand, Francisco Javier Segovia Pérez - Google Libros.” <https://books.google.es/books?hl=es&lr=&id=rXU-WS4UatYC&oi=fnd&pg=PA1&dq=ingenieria+de+software&ots=vxvMz83pXW&sig=rR3uU2smMBslrpzLIVkwLBCSt8I&pli=1#v=onepage&q=ingenieria+de+software&f=false> (accessed Nov. 23, 2022).
- [15] L. Nahuel Delia, “Desarrollo de aplicaciones híbridas,” 2017. Accessed: May 04, 2023. [Online]. Available: <https://www.oracle.com/technetwork/es/articles/adf/integracion-continua-adf-parte3-4107412-esa.html%0Ahttp://blog.aplicacionesmovil.com/aplicaciones-celular/desarrollo-de-aplicaciones-hibridas>
- [16] “¿Qué es Xamarin? - Xamarin | Microsoft Learn.” <https://learn.microsoft.com/es-es/xamarin/get-started/what-is-xamarin> (accessed May 16, 2023).

- [17] V. Rodríguez, “Desarrollo de aplicaciones móviles multiplataforma con Flutter,” *Univ. Almer.*, p. 72, 2019, [Online]. Available: [http://repositorio.ual.es/bitstream/handle/10835/8010/TFG\\_VAZQUEZ RODRIGUEZ, VICTOR.pdf?sequence=1](http://repositorio.ual.es/bitstream/handle/10835/8010/TFG_VAZQUEZ RODRIGUEZ, VICTOR.pdf?sequence=1)
- [18] “Ventajas y desventajas de apps desarrolladas en Xamarin.” <https://inmediatum.com/blog/ingenieria/ventajas-y-desventajas-de-apps-desarrolladas-en-xamarin/> (accessed Jun. 05, 2023).
- [19] T. Paratsii, “Flutter vs. React Native vs. Xamarin | Leobit,” 2022. <https://leobit.com/blog/flutter-vs-react-native-vs-xamarin-what-to-choose-for-your-project/> (accessed Jun. 05, 2023).
- [20] Google, “Flutter architectural overview | Flutter,” 12/03/2023, 2023. <https://docs.flutter.dev/resources/architectural-overview> (accessed May 13, 2023).
- [21] Flutter-Dev, “Introducción a los Widgets - Flutter,” *Flutter Docs*, 2022. <https://esflutter.dev/docs/development/ui/widgets-intro> (accessed May 13, 2023).
- [22] J. R. Molina Ríos, J. A. Honores Tapia, N. Pedreira-Souto, and H. P. Pardo León, “Estado del arte: metodologías de desarrollo de aplicaciones móviles,” *3C Technol. innovación Apl. a la pyme*, vol. 10, no. 2, pp. 17–45, 2021, doi: 10.17993/3ctecno/2021.v10n2e38.17-45.
- [23] J. Dianella, T. Guerra, I. Alejandro, J. Moreno, O. J. Linkolk, and L. Gonzales, “Metodologías ágiles Scrum, XP, SLeSS, Scrumban, HME, Mobile-D y MASAN empleadas en la industria de dispositivos móviles: Un contraste en favor de la industria del desarrollo móvil,” UNIVERSIDAD PERUANA UNIÓN, 2019.
- [24] J. E. Salvay, “Kanban y Scrumban orientados a Proyectos de Tecnología de la Información,” 2015.
- [25] J. H. Canós, P. Letelier, and C. Penadés, “Riesgos en el desarrollo de software,” p. 8. Accessed: May 03, 2023. [Online]. Available: [www.agileuniverse.com](http://www.agileuniverse.com).
- [26] Y. D. Amaya Balaguera, “Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual,” *Rev. Technol.*, vol. 12, no. 2, 2015, doi: 10.18270/rt.v12i2.1291.
- [27] E. W. Aldas, “Desarrollo de una aplicación para reconocimiento y muestra de información de obras de arte: RecoArt,” 2018.
- [28] C. Antonio Muñoz Muñoz and I. Pamela Buñay Mgs, “Universidad Nacional de Chimborazo Facultad de Ingeniería una app móvil para gestionar citas médicas del centro JEL Robamba”.
- [29] J. G. Enriquez and S. I. Casas, “Usabilidad en aplicaciones móviles, vista de usabilidad en aplicaciones móviles,” *Itc*, p. 23, 2013, Accessed: Nov. 24, 2022. [Online]. Available: <https://publicaciones.unpa.edu.ar/index.php/ICTUNPA/article/view/581/560>

- [30] P. Thomas, N. Galdamez, L. Delia, L. Corbalán, and P. Pesado, “Dispositivos Móviles: Desarrollo de Aplicaciones Multiplataforma,” 2015.
- [31] J. Cañizares, “Aplicación con tecnología móvil para monitoreo, geolocalización en tiempo real, y control de las solicitudes y entregas de pedidos para la empresa motoencomiendas el flaco de la ciudad de Latacunga,” 2022. [Online]. Available: <https://repositorio.uta.edu.ec/jspui/handle/123456789/34815>
- [32] A. Haro, “Universidad Nacional De Chimborazo Facultad De Ciencias Políticas Y Administrativas Carrera De Contabilidad Y Auditoría,” *Univ. Nac. Chimborazo*, pp. 1–2, 2017, Accessed: Nov. 24, 2022. [Online]. Available: <http://dspace.unach.edu.ec/handle/51000/4301>
- [33] Congreso Nacional Comisión de Legislación y Codificación, “Codificación de la ley de inquilinato,” *Codif. la ley inquilinato*, vol. (Registro, pp. 1–7, 2015, [Online]. Available: [https://www.registrocivil.gob.ec/wp-content/uploads/downloads/2017/05/Ley\\_de\\_Inquilinato.pdf](https://www.registrocivil.gob.ec/wp-content/uploads/downloads/2017/05/Ley_de_Inquilinato.pdf)
- [34] D. Ocampo Medina, “Facultad de Ciencias Sociales y Gestión mención en dirección de cuentas tema: Propuesta de Factibilidad para la Creación de una Solución Tecnológica Integrada para Comunicación de la Información en las Empresas de Publicidad y AUTOR : Daniel Ocampo Medin,” 2015.
- [35] S. Carralón, “Publicidad digital: qué es y cómo llevarla a cabo ★ The Social Media Family,” *The Social Media Family*, 2018. [https://thesocialmediafamily.com/publicidad-digital/#Que\\_es\\_la\\_publicidad\\_digital](https://thesocialmediafamily.com/publicidad-digital/#Que_es_la_publicidad_digital) (accessed Nov. 24, 2022).
- [36] S. Lavandeira Rojo, “Publicidad digital,” *Janus*, vol. 1, no. 1, pp. 257–262, 2014, Accessed: Nov. 24, 2022. [Online]. Available: <http://dialnet.unirioja.es/descarga/articulo/5181034.pdf>
- [37] M. L. García Guardia, J. I. Niño, and P. Núñez Gómez, “Nuevas perspectivas de la publicidad en el teléfono móvil. Relación de los usuarios y la tecnología,” *Área Abierta*, no. 29, 2011, doi: 10.5209/rev\_arab.2011.n29.36200.
- [38] Universidad de Málaga, “Grado En Publicidad Y Relaciones Públicas - Competencias Publicidad Y Rr.Pp,” pp. 1–50, 2021, [Online]. Available: <https://www.uma.es/grado-en-publicidad-y-relaciones-publicas/info/8379/competencias-publicidad-y-rrpp/>
- [39] F. D. Cajacuri Zacarias, C. A. Medina Rivera, W. C. J. Ordoñez Alcantara, and R. I. Rosales Perez, “Propuesta De Negocio Para La Intermediación De Alquileres De Habitaciones Para Estudiantes a Través De Una Plataforma Virtual,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1–301, 2018.
- [40] Z. Mat, “Flutter vs. React Native in 2022,” 2022. <https://www.amplifyre.com/articles/flutter-vs-react-native-vs-xamarin-best-in-2022> (accessed Jun. 05, 2023).
- [41] “Estructura tu base de datos | Firebase Realtime Database.” <https://firebase.google.com/docs/database/web/structure-data?hl=es> (accessed Jun. 15, 2023).

## ANEXOS

### Anexo 1. Ley del inquilinato

#### Título VI

##### DE LOS CONTRATOS DE ARRENDAMIENTO Y SUS EFECTOS

Art. 27.- Formas de los contratos. - El contrato de arrendamiento podrá ser verbal o escrito.

Art. 28.- Plazo del contrato escrito. - El plazo estipulado en el contrato escrito será obligatorio para arrendador y arrendatario. Sin embargo, en todo contrato de arrendamiento tendrá derecho el arrendatario a una duración mínima de dos años, excepto en los siguientes casos:

- a) De habitaciones en hoteles, casas de pensión o posadas;
- b) De arrendamiento de locales a individuos o familias que, teniendo su residencia habitual en un lugar, van a otros transitoriamente; y,
- c) De arrendamiento de locales para exhibiciones, espectáculos y otros fines, que por su propia naturaleza, tengan corta duración.

#### Título VII

##### DE LA TERMINACIÓN DEL CONTRATO DE ARRENDAMIENTO

Art. 30.- Causales de terminación.- El arrendador podrá dar por terminado el arrendamiento y, por consiguiente, exigir la desocupación y entrega del local arrendado antes de vencido el plazo legal o convencional, sólo por una de las siguientes causas:

- a) Cuando la falta de pago de las dos pensiones locativas mensuales se hubiere mantenido hasta la fecha en que se produjo la citación de la demanda al inquilino;
- b) Peligro de destrucción o ruina del edificio en la parte que comprende el local arrendado y que haga necesaria la reparación;

- c) Algazaras o reyertas ocasionadas por el inquilino;
- d) Destino del local arrendado a un objeto ilícito o distinto del convenido;
- e) Daños causados por el inquilino en el edificio, sus instalaciones o dependencias, conforme a lo previsto en el Art. 7;
- f) Subarriendo o traspaso de sus derechos, realizados por el inquilino, sin tener autorización escrita para ello;
- g) Ejecución por el inquilino en el local arrendado de obras no autorizadas por el arrendador;
- h) Resolución del arrendador de demoler el local para nueva edificación. En ese caso, deberá citarse legalmente al inquilino con la solicitud de desahucio, con tres meses de anticipación por lo menos, a la fecha fijada, para la demolición, la que sólo podrá ser tramitada cuando se acompañen los planos aprobados y el permiso de la Municipalidad respectiva para iniciar la obra.
- i) Decisión del propietario de ocupar el inmueble arrendado, siempre y cuando justifique legalmente la necesidad de hacerlo, porque es arrendatario y no tiene otro inmueble que ocupar [33].



## Anexo 2. Estadísticas de número de estudiantes.



### Anexo 3. Kuder Richardson arrendadores.

Individuos	Preguntas		
	P1	P8	
1	1	1	2
2	0	0	0
3	0	0	0
4	0	0	0
5	1	0	1
6	1	1	2
7	1	0	1
8	0	0	0
9	0	0	0
10	0	0	0
11	0	0	0
12	1	0	1
13	0	0	0
14	0	0	0
15	0	0	0
<b>Totales</b>	5	2	
p	0,33	0,13	
q	0,67	0,87	
p*q	0,22	0,12	
$\Sigma(p*q)$	0,34		
$\sigma^2$	0,55		
K	2		

$$\left( \frac{k}{k-1} \right) \rightarrow 2,00$$

$$\left( 1 - \frac{\Sigma pq}{\sigma^2} \right) \rightarrow 0,39$$

<b>KR-20</b>	<b>0,78</b>
--------------	-------------



### Anexo 5. Kuder Richardson arrendatarios.

Individuos	Preguntas	
	P1	P10
1	0	1
2	0	1
3	0	1
4	0	1
5	0	1
6	0	1
7	0	1
8	0	1
9	1	1
10	0	1
11	0	1
12	0	1
13	0	1
14	0	1
15	0	1
16	0	1
17	0	1
18	0	1
19	0	1
20	0	1
21	0	1
22	0	1
23	1	1
24	0	1
25	1	1
26	0	1
27	1	1
28	1	1
29	0	0
30	0	0
<b>Totales</b>	5	28
p	0,17	0,93
q	0,83	0,07
p*q	0,14	0,06
$\Sigma(p*q)$	0,14	
$\sigma^2$	0,23	
K	2	

$\left(\frac{k}{k-1}\right)$	>	2,00		
$\left(1 - \frac{\Sigma pq}{\sigma^2}\right)$	>	0,40		
			<b>KR-20</b>	<b>0,80</b>

**Anexo 6. Alfa de Cronbach arrendatarios.**

	Preguntas				
OBJETO DE ESTUDIO	4	7			
1	4	4	8		
2	5	5	10		
3	4	4	8		
4	5	5	10		
5	4	4	8		
6	4	5	9		
7	4	4	8		
8	5	4	9		
9	3	3	6		
10	5	5	10		
11	5	4	9		
12	5	5	10		
13	5	4	9		
14	5	5	10		
15	5	5	10		
16	4	4	8		
17	4	4	8		
18	5	3	8		
19	5	4	9		
20	3	3	6		
21	4	4	8		
22	4	3	7		
23	5	4	9		
24	5	5	10		
25	5	4	9		
26	4	4	8		
27	2	2	4		
28	4	4	8		
29	4	4	8		
30	4	4	8		
	0,556	0,529			

$\alpha$ (ALFA) =	0,821
K (NUMERO DE ITEMS) =	2
$\sum Vi$ (VARIANZA DE CADA ITEM)=	1,084
Vt (VARIANZA TOTAL) =	1,84

## Anexo 7. Pantalla de inicio o Splash Screen.

```
lib > paginas > splash_screen.dart > _SplashScreenState > initState
1 import 'dart:async';
2 import 'package:arriendos_uta/paginas/login_pagina.dart';
3 import 'package:flutter/material.dart';
4
5 class SplashScreen extends StatefulWidget {
6   @override
7   _SplashScreenState createState() => _SplashScreenState();
8 }
9
10 class _SplashScreenState extends State<SplashScreen> {
11   @override
12   void initState() {
13     super.initState();
14
15     Timer(Duration(seconds: 3), () {
16       // Navegar a la siguiente pantalla después de un tiempo determinado
17       Navigator.of(context).pushReplacement(
18         MaterialPageRoute(builder: (context) => LoginPage()),
19       );
20     }); // Timer
21   }
22
23   @override
24   Widget build(BuildContext context) {
25     return Scaffold(
26       body: Center(
27         child: Image.asset('assets/rent.jpg'),
28       ), // Center
29     ); // Scaffold
30   }
31 }
```

## Anexo 8. Método \_signIn.

```
login_pagina.dart 3, M X
lib > paginas > login_pagina.dart > _LoginPageState > _signIn

34 void _signIn() async {
35   if (_formKey.currentState != null && _formKey.currentState!.validate()) {
36     setState(() {
37       _errorMessage = '';
38       _isLoading = true;
39     });
40
41     _formKey.currentState!.save(); //Guarda los valores ingresados en el formulario
42
43     try {
44       UserCredential? userCredential = await authService.signIn(
45         context,
46         _email,
47         _password,
48       );
49
50       if (userCredential != null) {
51         Navigator.of(context).pushReplacement(
52           MaterialPageRoute(
53             builder: (context) => HomePage(),
54           ), // MaterialPageRoute
55         );
56       } else {
57         showDialog(
58           context: context,
59           builder: (context) => AlertDialog(
60             title: Text('Error de inicio de sesión'),
61             content: Text('Usuario o contraseña incorrectos'),
62             actions: [
63               TextButton(
64                 onPressed: () => Navigator.pop(context),
65                 child: Text('Cerrar'),
66               ), // TextButton
67             ],
68           ), // AlertDialog
69         );
70       }
71     } catch (error) {
72       setState(() {
73         _errorMessage = error.toString();
74       });
75     } finally {
76       setState(() {
77         _isLoading = false;
78       });
79     }
80   }
81 }
```

## Anexo 9. Inicio de sesión utilizando Firebase Authentication.

```
login_pagina.dart 3, M  auth.service.dart 2, M X
lib > servicios > auth.service.dart > AuthService > signIn
19
20 //MÉTODO PARA INGRESAR POR EMAIL Y CONTRASEÑA
21 Future<UserCredential?> signIn(
22   BuildContext context, String email, String password) async {
23   try {
24     UserCredential userCredential =
25       await _firebaseAuth.signInWithEmailAndPassword(
26         email: email,
27         password: password,
28       );
29
30     // Obtener el documento del usuario utilizando Firestore
31     DocumentSnapshot userDocument = await FirebaseFirestore.instance
32       .collection('usuarios')
33       .doc(userCredential.user!.uid)
34       .get();
35
36     // Acceso a los datos del documento del usuario
37     if (userDocument.exists) {
38       Map<String, dynamic> userData =
39         userDocument.data() as Map<String, dynamic>;
40       // Mapear los datos del documento en un objeto Usuario
41       Usuario usuario = Usuario(
42         nombre: userData['nombre'] ?? '',
43         apellido: userData['apellido'] ?? '',
44         telefono: userData['phone'] ?? '',
45         uid: userData['uid'] ?? '',
46         email: userData['email'] ?? '',
47         rol: userData['rol'] ?? '',
48       );
49
50       final UserProvider userProvider =
51         Provider.of<UserProvider>(context, listen: false);
52       userProvider.setUser(usuario);
53       print('Datos del usuario: $userData');
54       return userCredential;
55     } else {
56       print('El documento del usuario no existe');
57       return null;
58     }
59   } catch (e) {
60     print('Error durante el inicio de sesión: $e');
61     return null;
62   }
63 }
```



## Anexo 10. Control de campos: email y contraseña en el método `_signIn`.

```
login_pagina.dart 3, M X
lib > paginas > login_pagina.dart > _LoginPageState > build
134     child: Padding(
135       padding: const EdgeInsets.all(16.0),
136       child: Form(
137         key: _formKey,
138         child: Column(
139           children: [
140             TextFormField(
141               controller: _emailController,
142               decoration: InputDecoration(
143                 labelText: 'Correo',
144                 prefixIcon: Icon(Icons.email),
145                 border: InputBorder.none,
146                 contentPadding: EdgeInsets.all(16.0),
147               ), // InputDecoration
148               validator: (value) {
149                 if (value!.isEmpty) {
150                   return 'Por favor, ingresa un correo electrónico';
151                 }
152                 if (!_isValidEmail(value)) {
153                   return 'Por favor, ingresa un correo electrónico válido';
154                 }
155                 return null;
156               },
157               onSave: (value) => _email = value!,
158             ), // TextFormField
159             SizedBox(height: 16.0),
160             TextFormField(
161               controller: _passwordController,
162               decoration: InputDecoration(
163                 labelText: 'Contraseña',
164                 prefixIcon: Icon(Icons.lock),
165                 border: InputBorder.none,
166                 contentPadding: EdgeInsets.all(16.0),
167               ), // InputDecoration
168               validator: (value) {
169                 if (value!.isEmpty) {
170                   return 'Por favor, ingresa una contraseña';
171                 }
172                 return null;
173               },
174               onSave: (value) => _password = value!,
175               obscureText: true,
176             ), // TextFormField
177           ],
178         ), // Column
179       ), // Form
180     ), // Padding
181   ), // Container
182   const SizedBox(height: 16.0),
183   ElevatedButton(
184     onPressed: _signIn,
185     child: const Text('Iniciar sesión'),
```

## Anexo 11. Validación email de \_signIn.

```
//METODO PARA VALIDAR CORREO DEL LOGIN
bool _isValidEmail(String value) {
    // Validación básica del formato del correo electrónico.
    final emailRegex =
        r'^[\w-]+(\.[\w-]+)*@[a-zA-Z0-9-]+(\.[a-zA-Z0-9-]+)*(\.[a-zA-Z]{2,})$';
    final regExp = RegExp(emailRegex);
    return regExp.hasMatch(value);
}
```

## Anexo 12. Limpieza campos email y contraseña.

```
//METODO PARA LIMPAIR CAMPOS DE CORREO Y LOGIN
void _limpiarCampos() {
    _emailController.clear();
    _passwordController.clear();
}
```

## Anexo 13. Uso del servicio AuthService.

```
register_pagina.dart M X
lib > paginas > register_pagina.dart > ...
170 ElevatedButton(
171   child: Text('Registrar'),
172   onPressed: () async {
173     String firstName = _firstNameController.text;
174     String lastName = _lastNameController.text;
175     String phone = _phoneController.text;
176     String password = _passwordController.text;
177     String email = _emailController.text;
178     String confirmPassword = _confirmPasswordController.text;
179
180     // Validación de datos
181     if (firstName.isNotEmpty &&
182         lastName.isNotEmpty &&
183         phone.isNotEmpty &&
184         password.isNotEmpty) {
185       if (password == confirmPassword) {
186         AuthService authService =
187           AuthService(); //instancio el servicio
188         UserCredential? userCredential =
189           await authService.registerUserWithEmail(
190             email,
191             password,
192             firstName,
193             lastName,
194             phone,
195             _selectedRole); //envio datos
196         if (userCredential != null) {
197           ScaffoldMessenger.of(context).showSnackBar(
198             SnackBar(
199               content: Text('Registro exitoso'),
200               duration: Duration(seconds: 3),
201             ), // SnackBar
202           );
203
204           await Future.delayed(
205             Duration(seconds: 3)); // Esperar 3 segundos // Future.delayed
206           Navigator.pop(context); // Regresar a la pantalla anterior
207         }
208       } else {
209         ScaffoldMessenger.of(context).showSnackBar(
210           SnackBar(
211             content: Text('Las contraseñas no coinciden.'),
```

## Anexo 14. Llamado al método mapaSolo a la pantalla home.

```
lib > paginas > inicio > mapa_MostrarLista.dart > MostrarMapaState > build
4 import 'package:flutter/material.dart';
5 import 'package:latlong2/latlong.dart';
6
7 class MostrarMapa extends StatefulWidget {
8   @override
9   MostrarMapaState createState() => _MostrarMapaState();
10 }
11
12 class _MostrarMapaState extends State<MostrarMapa> {
13   LatLng _markerLocation = LatLng(-1.2687, -78.62434);
14
15   @override
16   Widget build(BuildContext context) {
17     return Column(
18       children: [
19         Padding(
20           padding: const EdgeInsets.all(8.0),
21           child: Row(
22             mainAxisAlignment: MainAxisAlignment.spaceBetween,
23             children: [
24               ElevatedButton.icon(
25                 onPressed: () {
26                   Navigator.push(
27                     context,
28                     MaterialPageRoute(builder: (context) => FiltroPage()),
29                   );
30               },
31               icon: Icon(Icons.filter_list),
32               label: Text('Filtros'),
33             ), // ElevatedButton.icon
34             ElevatedButton.icon(
35               onPressed: () {
36                 Navigator.push(
37                   context,
38                   MaterialPageRoute(builder: (context) => listaPage()),
39                 );
40             },
41             icon: Icon(Icons.list),
42             label: Text('Listado'),
43           ), // ElevatedButton.icon
44         ],
45       ), // Row
46     ), // Padding
47     //TRAER EL MAP
48     Expanded(
49       child: mapaSolo(
50         markerLocation: _markerLocation,
51         onMarkerLocationChanged: (LatLng location) {
52           setState(() {
53             _markerLocation = location;
54           });
55         },
56     ),
```

## Anexo 15. Mapa iterativo con marcador.

```
lib > paginas > inicio > mapaSolo.dart > mapaSolo > mapaSolo
9  mapaSolo({
10     required this.markerLocation,
11     required this.onMarkerLocationChanged,
12 });
13
14 @override
15 _MapaPruebaState createState() => _MapaPruebaState();
16 }
17
18 class _MapaPruebaState extends State<mapaSolo> {
19   late LatLng _markerLocation;
20
21   @override
22   void initState() {
23     super.initState();
24     _markerLocation = widget.markerLocation;
25   }
26
27   @override
28   Widget build(BuildContext context) {
29     return FlutterMap(
30       options: MapOptions(
31         center: _markerLocation,
32         zoom: 15.0,
33         onTap: (tapPosition, LatLng tapLatLng) {
34           setState(() {
35             _markerLocation = tapLatLng;
36             widget.onMarkerLocationChanged(tapLatLng);
37           });
38         },
39       ), // MapOptions
40       children: [
41         TileLayer(
42           urlTemplate: 'https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',
43           subdomains: ['a', 'b', 'c'],
44         ), // TileLayer
45         MarkerLayer(
46           markers: [
47             Marker(
48               width: 40.0,
49               height: 40.0,
50               point: _markerLocation,
51               builder: (ctx) => GestureDetector(
52                 onTap: (details) {
53                   setState(() {
54                     _markerLocation = LatLng(
55                       _markerLocation.latitude + details.delta.dy,
56                       _markerLocation.longitude + details.delta.dx,
57                     ); // LatLng
58                     widget.onMarkerLocationChanged(_markerLocation);
59                   });
60                 },
61               ),
62             ],
63         ),
64       ],
65     );
66   }
67 }
```

## Anexo 16. Guardar los datos de usuarios en Firestore Database.

```
auth.service.dart 2, M x
lib > servicios > auth.service.dart > AuthService > registerUserWithEmail
89
90 //MÉTODO PARA REGISTRAR UN USUARIO POR EMAIL Y CONTRASEÑA
91
92 Future<UserCredential?> registerUserWithEmail(String email, String password,
93 String firstName, String lastName, String phone, String rol) async {
94   try {
95     UserCredential userCredential =
96       await _firebaseAuth.createUserWithEmailAndPassword(
97         email: email,
98         password: password,
99       );
100
101     // Asignar nombre de usuario
102     if (userCredential.user != null) {
103       await userCredential.user!
104         .updateDisplayName(firstName + ' ' + lastName);
105     }
106
107     // Almacenar los datos en la colección "usuarios" de Firestore
108     if (userCredential.user != null) {
109       await _firestore
110         .collection('usuarios')
111         .doc(userCredential.user!.uid)
112         .set({
113           'uid': userCredential.user!.uid,
114           'nombre': firstName,
115           'apellido': lastName,
116           'email': email,
117           'contraseña': password,
118           'rol': rol,
119           'phone': phone,
120           'fecha_creacion': DateTime.now(),
121         });
122     }
123
124     return userCredential;
125   } catch (e) {
126     // Manejo de errores
127     _errorMessage = e.toString(); // Capturar el mensaje de error
128     return null;
129   }
130 }
```

## Anexo 17. Validación campos: rol, nombre, apellido.

```
register_pagina.dart M x
lib > paginas > register_pagina.dart > _paginaRegistroState > build
66   DropdownButtonFormField(
67     value: _selectedRole,
68     items: _roles.map((role) {
69       return DropdownMenuItem(
70         value: role,
71         child: Text(role),
72       ); // DropdownMenuItem
73     }).toList(),
74     hint: Text('Seleccione un rol'),
75     onChanged: (value) {
76       setState(() {
77         _selectedRole = value!;
78       });
79     },
80     validator: (value) {
81       if (value == null) {
82         return 'Por favor, seleccione un rol';
83       }
84       return null;
85     },
86   ), // DropdownButtonFormField
87
88   TextField(
89     inputFormatters: [
90       FilteringTextInputFormatter.allow(RegExp(r'[a-zA-Z\s]')),
91     ],
92     controller: _firstNameController,
93     decoration: InputDecoration(
94       labelText: 'Nombre',
95     ), // InputDecoration
96   ), // TextField
97   SizedBox(height: 10),
98   TextField(
99     inputFormatters: [
100     FilteringTextInputFormatter.allow(RegExp(r'[a-zA-Z\s]')),
101   ],
102   controller: _lastNameController,
103   decoration: InputDecoration(
104     labelText: 'Apellido',
105   ), // InputDecoration
106 ), // TextField
```



## Anexo 18. Validación campo: celular.

```
register_pagina.dart M x
lib > paginas > register_pagina.dart > _paginaRegistroState > build
108   TextField(
109     inputFormatters: [
110       FilteringTextInputFormatter.allow(RegExp(r'[0-9]')),
111       LengthLimitingTextInputFormatter(10),
112     ],
113     controller: _phoneController,
114     decoration: InputDecoration(
115       labelText: 'Celular',
116     ), // InputDecoration
117   ), // TextField
```

## Anexo 19. Validación campos: email y contraseña.

```
//VALIDACION EMAIL
bool _validateEmail(String value) {
    String pattern = r'^[\w-]+(\.[\w-]+)*@([a-zA-Z0-9-]+\.)+[a-zA-Z]{2,7}$';
    RegExp regex = RegExp(pattern);
    return regex.hasMatch(value);
}

void _validateInput(String value) {
    setState(() {
        _isEmailValid = _validateEmail(value);
    });
}

//VALIDACION PASSWORD
bool _validatePassword(String value) {
    // Mínimo 8 caracteres, al menos una letra mayúscula, una letra minúscula y un número
    String pattern = r'^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9]).{8,}$';
    RegExp regex = RegExp(pattern);
    return regex.hasMatch(value);
}

void _validateInputPassword(String value) {
    setState(() {
        _isPasswordValid = _validatePassword(value);
    });
}
```

## Anexo 20. Listado de publicaciones en el mapa.

```
lib > paginas > inicio > mapa_MostrarLista.dart > _MostrarMapaState > build
176     ],
177     ), // Row
178   ), // Padding
179   //TRAE EL MAP
180   Expanded(
181     child: FutureBuilder<List<MisPublicaciones>>(
182       future: _futurePublicaciones,
183       builder: (BuildContext context,
184         AsyncSnapshot<List<MisPublicaciones>> snapshot) {
185         if (snapshot.connectionState == ConnectionState.waiting) {
186           return Center(child: CircularProgressIndicator());
187         } else if (snapshot.hasError) {
188           return Center(child: Text('Error al cargar las publicaciones'));
189         } else {
190           final List<MisPublicaciones> publicaciones = snapshot.data!;
191           return MapSample(
192             ubicaciones: listaUbicaciones,
193             publicaciones: publicacionesFiltradas.isNotEmpty
194               ? publicacionesFiltradas
195               : publicaciones,
196             zoomLevel: widget.zoomLevel,
197           ); // MapSample
198         }
199       },
200     ), // FutureBuilder
201   ), // Expanded
```

## Anexo 21. Mostrar mapa de Google con el listado de publicaciones.

```
mapa_MostrarLista.dart 1, M  MapSample.dart 1 x
ib > paginas > inicio > MapSample.dart > MapSampleState > actualizarMarcadores
6 import 'package:flutter_polyline_points/flutter_polyline_points.dart';
7 import 'package:geolocator/geolocator.dart';
8 import 'package:google_maps_flutter/google_maps_flutter.dart';
9 import 'package:flutter/material.dart';
10 import 'package:http/http.dart' as http;
11
12 class MapSample extends StatefulWidget {
13   final List<UbicacionModel> ubicaciones;
14   final List<MisPublicaciones> publicaciones;
15   final double zoomLevel;
16   const MapSample({
17     required this.ubicaciones,
18     required this.publicaciones,
19     this.zoomLevel = 10.0,
20   });
21
22   @override
23   State<MapSample> createState() => MapSampleState();
24 }
25
26 class MapSampleState extends State<MapSample> {
27   final Completer<GoogleMapController> _controller =
28     Completer<GoogleMapController>();
29   List<Marker> _markers = [];
30   PolylinePoints polylinePoints = PolylinePoints(); // Add this line
31   List<Polyline> polylines = [];
32   late LatLng currentLocation = LatLng(0.0, 0.0);
33   Marker? currentLocationMarker;
34
35   void actualizarMarcadores(List<MisPublicaciones> publicaciones) {
36     setState(() {
37       _markers = [];
38       agregarMarcadores(publicaciones);
39     });
40   }
}
```

## Anexo 22. Edición del nombre, apellido y teléfono del usuario.

```
miCuentaAjustes.dart X user.service.dart 6, M
lib > paginas > Ajustes > miCuentaAjustes.dart > _miCuentaSettingsState > _guardarCambios
1 import 'package:arriendos_uta/modelos/usuario.dart';
2 import 'package:arriendos_uta/providers/user.provider.dart';
3 import 'package:arriendos_uta/servicios/user.service.dart';
4 import 'package:flutter/material.dart';
5 import 'package:flutter/services.dart';
6 import 'package:provider/provider.dart';
7
8 class miCuentaSettings extends StatefulWidget {
9   @override
10  _miCuentaSettingsState createState() => _miCuentaSettingsState();
11 }
12
13 class _miCuentaSettingsState extends State<miCuentaSettings> {
14   TextEditingController nombreController = TextEditingController();
15   TextEditingController apellidoController = TextEditingController();
16   TextEditingController telefonoController = TextEditingController();
17   //TextEditingController _emailController = TextEditingController();
18
19   @override
20   void initState() {
21     super.initState();
22     // Inicializar los controladores con los valores actuales del usuario
23     final userProvider = Provider.of<UserProvider>(context, listen: false);
24     Usuario? usuario = userProvider.user;
25     if (usuario != null) {
26       _nombreController.text = usuario.nombre;
27       _apellidoController.text = usuario.apellido;
28       _telefonoController.text = usuario.telefono;
29       //_emailController.text = usuario.email;
30     }
31   }
32 }
```

## Anexo 23. Método para guardar Cambios.

```
miCuentaAjustes.dart X user.service.dart 6, M
lib > paginas > Ajustes > miCuentaAjustes.dart > _miCuentaSettingsState > build
43 @override
44 Widget build(BuildContext context) {
45   final userProvider = Provider.of<UserProvider>(context);
46
47   Usuario? usuario = userProvider.user;
48
49   return Scaffold(
50     appBar: AppBar(
51       title: Text('Ajustes de perfil'),
52       actions: [
53         IconButton(
54           icon: Icon(Icons.save),
55           onPressed: () {
56             _guardarCambios(userProvider);
57           },
58         ), // IconButton
59       ],
60     ), // AppBar
61     body: usuario != null
62       ? Padding(
63         padding: EdgeInsets.all(16.0),
64         child: Column(
65           crossAxisAlignment: CrossAxisAlignment.start,
66           children: [
67             TextField(
68               inputFormatters: [
69                 FilteringTextInputFormatter.allow(RegExp(r'[a-zA-Z\s]')),
70               ],
71               controller: _nombreController,
72               decoration: InputDecoration(
73                 labelText: 'Nombre',
74               ), // InputDecoration
75             ), // TextField
76             TextField(
77               inputFormatters: [
78                 FilteringTextInputFormatter.allow(RegExp(r'[a-zA-Z\s]')),
79               ],
80               controller: _apellidoController,
81               decoration: InputDecoration(
82                 labelText: 'Apellido',
83               ), // InputDecoration
84             ), // TextField
85             TextField(
86               inputFormatters: [
87                 FilteringTextInputFormatter.allow(RegExp(r'[0-9]')),
88                 LengthLimitingTextInputFormatter(10),
89               ],
90               controller: _telefonoController,
91               decoration: InputDecoration(
92                 labelText: 'Celular',
93               ), // InputDecoration
94             ), // TextField
```

## Anexo 24. Actualización de datos del usuario mediante UserProvider.

```
Future<void> _guardarCambios(UserProvider userProvider) async {
  String nombre = _nombreController.text;
  String apellido = _apellidoController.text;
  String telefono = _telefonoController.text;
  String uid = userProvider.user?.uid ?? '';
  String email = userProvider.user?.email ?? '';
  String rol = userProvider.user?.rol ?? '';

  // Actualizar los datos del usuario en el UserProvider
  Usuario? usuario = userProvider.user;
  if (usuario != null) {
    Usuario usuarioActualizado = Usuario(
      uid: uid,
      nombre: nombre,
      apellido: apellido,
      telefono: telefono,
      email: email,
      rol: rol);
    userProvider.setUser(usuarioActualizado);
  }

  // Guardar los cambios en el backend o en la base de datos
  UserService userService = UserService();
  bool exitoso =
    await userService.updateUser(uid, nombre, apellido, telefono);

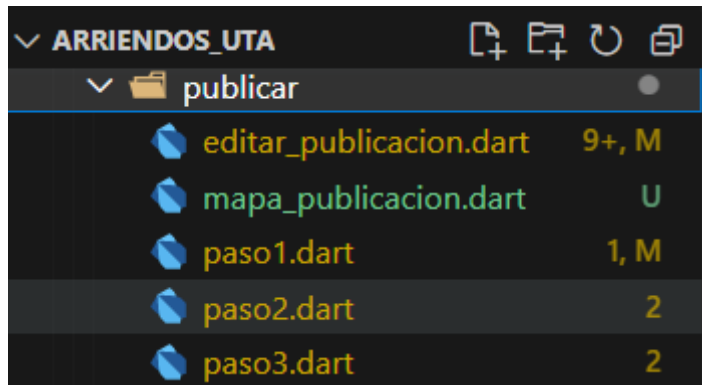
  // Mostrar un mensaje de éxito o cualquier otra acción que desees realizar
  if (exitoso) {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: Text('Cambios guardados correctamente'),
        duration: Duration(seconds: 2),
      ), // SnackBar
    );
  } else {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: Text('No se pudieron guardar los cambios'),
        duration: Duration(seconds: 2),
      ), // SnackBar
    );
  }
}
```

## Anexo 25. Actualizar los datos en Firebase.

```
miCuentaAjustes.dart user.service.dart 6, M X
lib > servicios > user.service.dart > UserService > updateUser
40 // Actualizar un usuario existente
41 Future<bool> updateUser(
42   String? uid, String nombre, String apellido, String telefono) async {
43   try {
44     User? user = FirebaseAuth.instance.currentUser;
45     if (user != null) {
46       //await user.updateEmail(correo);
47
48       await _usersCollection.doc(uid).update({
49         'nombre': nombre,
50         'apellido': apellido,
51         'phone': telefono,
52       });
53     }
54     return true; // Indicar éxito en la actualización
55   } catch (error) {
56     print('Error al actualizar el usuario: $error');
57     return false;
58   }
59 }
```



**Anexo 26. Contenido carpeta publicar.**



## Anexo 27. Validaciones del tamaño y precio.

```
lib > paginas > publicar > paso1.dart > Paso1State > build
79 TextFormField(
80   controller: tamañoController, //almacena la informacion
81   onChanged: (value) {
82     setState() {
83       departamentoTamañoValido = value.isNotEmpty;
84     };
85   },
86   keyboardType: TextInputType.number,
87   inputFormatters: <TextInputFormatter>[
88     FilteringTextInputFormatter.allow(RegExp(r'^\d*\.\?\d{0,2}$'))
89   ], // <TextInputFormatter>[]
90   decoration: InputDecoration(
91     labelText: 'Tamaño del departamento',
92     errorText:
93       departamentoTamañoValido ? null : 'Ingrese un tamaño válido',
94     suffixText: 'm²',
95   ), // InputDecoration
96 ), // TextFormField
97 TextFormField(
98   controller: precioController,
99   onChanged: (value) {
100     departamentoPrecio = value;
101   },
102   keyboardType: TextInputType.number,
103   inputFormatters: <TextInputFormatter>[
104     FilteringTextInputFormatter.allow(
105       RegExp(r'^\d*\.\?\d{0,2}$')) // Cantidades con 2 decimales // Fi
106   ], // <TextInputFormatter>[]
107   decoration: InputDecoration(
108     labelText: 'Precio del departamento',
109     prefixText: '\$',
110   ), // InputDecoration
111 ), // TextFormField
```

## Anexo 28. Añadir ubicación del cuarto de arriendo.

```
lib > paginas > publicar > paso1.dart > Paso1State > build
113 Text('Ubicación del departamento',
114 style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold)), // Text
115 ElevatedButton(
116   onPressed: () async {
117     final LatLng? location = await Navigator.push<LatLng?>(
118       context,
119       MaterialPageRoute(builder: (context) => MapaPublicar()),
120     );
121     if (location != null) {
122       setState(() {
123         _selectedLocation = location;
124       });
125       print(
126         'Ubicación seleccionada: ${location.latitude}, ${location.longitude}');
127       guardarDatosPaso1(publicacionProvider);
128     }
129   },
130   child: Text('Seleccionar ubicación'),
131 ), // ElevatedButton
132 if (_selectedLocation != null)
133 Text(
134   'Ubicación seleccionada: $_selectedLocation!.latitude, $_selectedLocation!.longitude',
135   style: TextStyle(fontSize: 16),
136   textAlign: TextAlign.center,
137 ), // Text
```

## Anexo 29. Clase MapaPublicar para seleccionar ubicación.

```
lib > paginas > mapas > mapa_publicar.dart > _MapaPublicarState > build
1 import 'package:flutter/material.dart';
2 import 'package:flutter_map/flutter_map.dart';
3 import 'package:latlong2/latlong.dart';
4
5 class MapaPublicar extends StatefulWidget {
6   @override
7   _MapaPublicarState createState() => _MapaPublicarState();
8 }
9
10 class _MapaPublicarState extends State<MapaPublicar> {
11   LatLng _markerLocation = LatLng(-1.2687, -78.62434);
12   LatLng? selectedLocation; //Guardar lat y long
13
14   @override
15   Widget build(BuildContext context) {
16     return Scaffold(
17       appBar: AppBar(
18         title: Text('Map Page'),
19       ), // AppBar
20       body: FlutterMap(
21         options: MapOptions(
22           center: _markerLocation,
23           zoom: 15.0,
24           onTap: (tapPosition, LatLng tapLatLng) {
25             setState(() {
26               _markerLocation = tapLatLng; //Permite mover el marker
27               selectedLocation =
28                 tapLatLng; //Actualiza la variable con la latitud y longitud seleccionadas
29             });
30           },
31         ), // MapOptions
32         children: [
33           TileLayer(
34             urlTemplate: 'https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',
35             subdomains: ['a', 'b', 'c'],
36           ), // TileLayer
37           MarkerLayer(
38             markers: [
39               Marker(
40                 width: 40.0,
41                 height: 40.0,
42                 point: _markerLocation,
43                 builder: (ctx) => GestureDetector(
44                   onTap: (details) {
45                     setState(() {
46                       _markerLocation = LatLng(
47                         _markerLocation.latitude +
48                           details.delta.dy, //Se guarda la latitud
49                         _markerLocation.longitude +
50                           details.delta.dx, //Se guarda la longitud
51                       ); // LatLng
52                     });
53                   },
54                 ),
55             ],
56           ), // MarkerLayer
57         ],
58       ), // FlutterMap
59     ); // Scaffold
60   }
61 }
```

### Anexo 30. Envío de ubicación en el mapa.

```
floatingActionButton: FloatingActionButton(  
  onPressed: () {  
    if (selectedLocation != null) {  
      Navigator.pop(context,  
        selectedLocation); //se cierra la página y envíe los datos de ubicación seleccionados  
    }  
  },  
  child: Icon(Icons.send),  
), // FloatingActionButton
```

## Anexo 31. Habilitar habitación amoblada, individual o compartida.

```
editar_publicacion.dart 9+, M  paso1.dart 1, M X
lib > paginas > publicar > paso1.dart > Paso1State > build
138  CheckboxListTile(
139    title: Text('Amoblado'),
140    value: amoblado,
141    onChanged: (value) {
142      setState(() {
143        amoblado = value!;
144      });
145    },
146  ), // CheckboxListTile
147  RadiolistTile(
148    title: Text('Cocina individual'),
149    value: 'Cocina individual',
150    groupValue: preferenciaCocina,
151    onChanged: (value) {
152      setState(() {
153        preferenciaCocina = value!;
154      });
155    },
156  ), // RadiolistTile
157  RadiolistTile(
158    title: Text('Cocina compartida'),
159    value: 'Cocina compartida',
160    groupValue: preferenciaCocina,
161    onChanged: (value) {
162      setState(() {
163        preferenciaCocina = value!;
164      });
165    },
166  ), // RadiolistTile
```

## Anexo 32. Funcionalidad de seleccionar y adjuntar un archivo PDF.

```
editar_publicacion.dart 9+, M  paso1.dart M x  contrato.dart
lib > paginas > publicar > paso1.dart > Paso1State > build
166 Column(
167   children: [
168     ElevatedButton(
169       onPressed: () async {
170         // Seleccionar un archivo PDF
171         final FilePickerResult? result =
172           await FilePicker.platform.pickFiles(
173             type: FileType.custom,
174             allowedExtensions: ['pdf'],
175           );
176
177         if (result != null) {
178           final File file = File(result.files.single.path!);
179           final fileName = result.files.single.name;
180           setState(() {
181             publicacionProvider.publicacion.contrato = Contrato(
182               file: file,
183               fileName: fileName); //extrae el nombre del fie // Contrato
184           });
185         }
186       },
187       style: ElevatedButton.styleFrom(
188         backgroundColor: Colors.red, // Cambia el color del botón a rojo
189       ),
190       child: Row(
191         mainAxisAlignment: MainAxisAlignment.min,
192         children: [
193           SizedBox(width: 8),
194           Text('Adjuntar contrato de arriendo PDF'),
195         ],
196       ), // Row
197     ), // ElevatedButton
198     if (publicacionProvider.publicacion.contrato != null)
199       Column(
200         children: [
201           Icon(Icons.picture_as_pdf, color: Colors.red),
202           SizedBox(height: 8),
203           Text(
204             publicacionProvider.publicacion.contrato!.fileName ?? ''), // Text
205         ],
206       ), // Column
```

### Anexo 33. Clase Contrato con URL y nombre del a archivo.

```
lib > modelos > contrato.dart > Contrato

1  import 'dart:io';
2
3  class Contrato {
4      String? url;
5      File? file;
6      String? fileName;
7
8      Contrato({this.url, this.file, this.fileName });
9
10     Map<String, dynamic> toJson() {
11         return {
12             'url': url,
13             'fileName': fileName,
14         };
15     }
16     factory Contrato.fromJson(Map<String, dynamic> json) {
17         return Contrato(
18             url: json['url'],
19             fileName: json['fileName'],
20             file: null,
21         );
22     }
23 }
```



## Anexo 34. Selección de imágenes desde la galería.

```
lib > paginas > publicar > paso1.dart > Paso1State > build
208 ElevatedButton(
209   onPressed: () async {
210     final ImagePicker picker = ImagePicker();
211     if (images.length >= 4) {
212       showDialog(
213         context: context,
214         builder: (BuildContext context) {
215           return AlertDialog(
216             title: Text('Limite alcanzado'),
217             content:
218               Text('Ya has adjuntado el máximo de 4 imágenes.'),
219             actions: [
220               TextButton(
221                 child: Text('Aceptar'),
222                 onPressed: () {
223                   Navigator.of(context).pop();
224                 },
225               ), // TextButton
226             ],
227           ); // AlertDialog
228         },
229       );
230       return;
231     }
232     final XFile? image =
233       await _picker.pickImage(source: ImageSource.gallery);
234     if (image != null) {
235       setState(() {
236         images.add(File(image.path));
237       });
238       publicacionProvider.publicacion.imagenes = images;
239     }
240   },
241   child: Text('Adjuntar imágenes'),
242 ), // ElevatedButton
```

## Anexo 35. Método estático guardarDatosPaso1.

```
lib > paginas > publicar > paso1.dart > Paso1State > guardarDatosPaso1
296 void guardarDatosPaso1(PublicacionProvider publicacionProvider) async {
297   final publicacion = publicacionProvider.publicacion;
298   publicacion.habitacion.tamano = tamanoController.text;
299   publicacion.habitacion.precio = double.tryParse(precioController.text) ?? 0;
300   publicacion.habitacion.amoblado = amoblado;
301   publicacion.habitacion.compartida = cocinaCompartida;
302   publicacion.habitacion.individual = cocinaIndividual;
303
304   if (_selectedLocation != null) {
305     publicacion.ubicacion.latitud = _selectedLocation!.latitud.toString();
306     publicacion.ubicacion.longitud = _selectedLocation!.longitud.toString();
307   }
308
309   publicacionProvider.actualizarPublicacion(publicacion);
310 }
311 }
```

## Anexo 36. Clase PublicacionProvider.

```
publicacion.provider.dart M X
lib > providers > publicacion.provider.dart > ...
1 import 'package:arriendos_uta/modelos/contrato.dart';
2 import 'package:flutter/foundation.dart';
3 import 'package:arriendos_uta/modelos/habitacion.dart';
4 import 'package:arriendos_uta/modelos/ubicacion.dart';
5 import 'package:arriendos_uta/modelos/publicacion.dart';
6
7 class PublicacionProvider extends ChangeNotifier {
8   PublicacionesModelo _publicacion = PublicacionesModelo(
9     id: '',
10    contrato: Contrato(),
11    nombre: '',
12    uidUsuario: '',
13    imagenes: [],
14    ubicacion: UbicacionModel(latitud: '', longitud: ''),
15    habitacion: HabitacionModelo(
16      tamaño: '',
17      amoblado: false,
18      individual: false,
19      compartida: false,
20      precio: 0.0,
21      serviciosBasicos: [],
22      mesesGarantia: '',
23      mascotas: false,
24      generoPreferencia: '',
25    ), // HabitacionModelo
26    descripcion: '',
27  ); // PublicacionesModelo
28
29  //Define un getter llamado publicacion en la clase PublicacionesModelo.
30  //Este getter devuelve el valor de la variable privada _publicacion.
31  PublicacionesModelo get publicacion => _publicacion;
32
33  // Metodo que toma una instancia de PublicacionesModelo como argumento
34  void actualizarPublicacion(PublicacionesModelo publicacion) {
35    _publicacion = publicacion; // actualiza la propiedad _publicacion
36    notifyListeners();
37  }
38
39  void resetPublicacion() {
40    _publicacion = PublicacionesModelo(
41      id: '',
42      contrato: Contrato(),
43      nombre: '',
44      uidUsuario: '',
45      imagenes: [],
46      ubicacion: UbicacionModel(latitud: '', longitud: ''),
47      habitacion: HabitacionModelo(
48        tamaño: '',
49        amoblado: false,
50        individual: false,
51        compartida: false,
52        precio: 0.0,
```

## Anexo 37. Servicios básicos como: internet, luz, agua y parqueadero.

```
lib > paginas > publicar > paso2.dart > Paso2State > build
39  @override
40  Widget build(BuildContext context) {
41    final publicacionProvider = Provider.of<PublicacionProvider>(context);
42    final publicacion = publicacionProvider.publicacion;
43    return Column(
44      children: [
45        Text(
46          'Servicios Básicos',
47          style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
48        ), // Text
49        CheckboxListTile(
50          title: Text('Internet'),
51          value: internet,
52          onChanged: (value) {
53            setState(() {
54              internet = value!;
55            });
56          },
57        ), // CheckboxListTile
58        CheckboxListTile(
59          title: Text('Luz'),
60          value: luz,
61          onChanged: (value) {
62            setState(() {
63              luz = value!;
64            });
65          },
66        ), // CheckboxListTile
67        CheckboxListTile(
68          title: Text('Agua'),
69          value: agua,
70          onChanged: (value) {
71            setState(() {
72              agua = value!;
73            });
74          },
75        ), // CheckboxListTile
76        CheckboxListTile(
77          title: Text('Parqueadero'),
78          value: parqueadero,
79          onChanged: (value) {
80            setState(() {
81              parqueadero = value!;
82            });
83          },
84        ), // CheckboxListTile
```

## Anexo 38. Meses de garantía en RadioListTile.

```
lib > paginas > publicar > paso2.dart > Paso2State > build
86   Text(
87     'Meses de Garantía:',
88     style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
89   ), // Text
90   RadioListTile(
91     title: Text('1 mes'),
92     value: '1 mes',
93     groupValue: mesesGarantia,
94     onChanged: (value) {
95       setState(() {
96         mesesGarantia = value!;
97       });
98     },
99   ), // RadioListTile
100  RadioListTile(
101    title: Text('2 meses'),
102    value: '2 meses',
103    groupValue: mesesGarantia,
104    onChanged: (value) {
105      setState(() {
106        mesesGarantia = value!;
107      });
108    },
109  ), // RadioListTile
110  RadioListTile(
111    title: Text('3 meses'),
112    value: '3 meses',
113    groupValue: mesesGarantia,
114    onChanged: (value) {
115      setState(() {
116        mesesGarantia = value!;
117      });
118    },
119  ), // RadioListTile
```

### Anexo 39. Método estático guardarDatosPaso2.

```
static void guardarDatosPaso2(PublicacionProvider publicacionProvider) {  
    final publicacion = publicacionProvider.publicacion;  
    List<ServiciosBasicosModel> listaServiciosBasicos = [];  
    listaServiciosBasicos  
        .add(ServiciosBasicosModel(flag: internet, nombre: 'Internet'));  
    listaServiciosBasicos.add(ServiciosBasicosModel(flag: luz, nombre: 'Luz'));  
    listaServiciosBasicos  
        .add(ServiciosBasicosModel(flag: agua, nombre: 'Agua'));  
    listaServiciosBasicos  
        .add(ServiciosBasicosModel(flag: parqueadero, nombre: 'Parqueadero'));  
    publicacion.habitacion.serviciosBasicos = listaServiciosBasicos;  
    publicacion.habitacion.mesesGarantia = mesesGarantia;  
  
    publicacionProvider.actualizarPublicacion(publicacion);  
}
```

## Anexo 40. Preferencia de género en RadioListTile.

```
paso3.dart 1, M x
lib > paginas > publicar > paso3.dart > Paso3State > build
@override
40   Widget build(BuildContext context) {
41     return Column(
42       children: [
43         Text(
44           'Preferencia de género',
45           style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
46         ), // Text
47         RadioListTile(
48           title: Text('Masculino'),
49           value: 'Masculino',
50           groupValue: preferenciaGenero,
51           onChanged: (value) {
52             setState(() {
53               preferenciaGenero = value!;
54             });
55           },
56         ), // RadioListTile
57         RadioListTile(
58           title: Text('Femenino'),
59           value: 'Femenino',
60           groupValue: preferenciaGenero,
61           onChanged: (value) {
62             setState(() {
63               preferenciaGenero = value!;
64             });
65           },
66         ), // RadioListTile
67         RadioListTile(
68           title: Text('Sin preferencia de género'),
69           value: 'Sin preferencia de género',
70           groupValue: preferenciaGenero,
71           onChanged: (value) {
72             setState(() {
73               preferenciaGenero = value!;
74             });
75           },
76         ), // RadioListTile
```

#### Anexo 41. Permiso de mascotas y descripción del cuarto de arriendo.

```
78     CheckboxListTile(  
79       title: Text('Permitir mascotas'),  
80       value: permitirMascotas,  
81       onChanged: (value) {  
82         setState(() {  
83           permitirMascotas = value!  
84         });  
85       },  
86     ), // CheckboxListTile  
87     SizedBox(height: 16),  
88     TextField(  
89       controller: descripcionController,  
90       decoration: InputDecoration(  
91         labelText: 'Descripción del departamento',  
92       ), // InputDecoration  
93       maxLines: null,  
94     ), // TextField
```



### Anexo 42. Método estático guardarDatosPaso3.

```
99  static void guardarDatosPaso3(PublicacionProvider publicacionProvider) {
100      PublicacionService publicacionService = PublicacionService();
101
102      final publicacion = publicacionProvider.publicacion;
103      publicacion.habitacion.mascotas = permitirMascotas;
104      publicacion.habitacion.generoPreferencia = preferenciaGenero;
105      publicacion.descripcion = descripcionController.text;
106      publicacionProvider.actualizarPublicacion(publicacion);
107  }
```

### Anexo 43. Implementación de paquetes en la clase PasosPage.

```
stepper_page.dart X
lib > paginas > publicar > stepper_page.dart > ...
1  import 'package:arriendos_uta/modelos/imagen.dart';
2  import 'package:arriendos_uta/paginas/home_pagina.dart';
3  import 'package:arriendos_uta/paginas/publicar/paso1.dart';
4  import 'package:arriendos_uta/paginas/publicar/paso2.dart';
5  import 'package:arriendos_uta/paginas/publicar/paso3.dart';
6  import 'package:arriendos_uta/providers/publicacion.provider.dart';
7  import 'package:arriendos_uta/providers/user.provider.dart';
8  import 'package:arriendos_uta/servicios/PdfStorageService%20.dart';
9  import 'package:arriendos_uta/servicios/publicacion.service.dart';
10 import 'package:flutter/material.dart';
11 import 'package:provider/provider.dart';
```

## Anexo 44. Clase PasosPage.

```
class PasosPage extends StatefulWidget {  
  @override  
  PasosPageState createState() => _PasosPageState();  
}  
  
class _PasosPageState extends State<PasosPage> {  
  int currentStep = 0;  
  bool isStepComplete = false;  
  
  final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();  
}
```

## Anexo 45. Widgets Paso1, Paso2 y Paso3.

```
body: Consumer<PublicacionProvider>(
  builder: (context, publicacionProvider, _) {
    isStepComplete = _currentStep ==
      0 && // _currentStep determina el paso en el que me encuentro
      Paso1State.validarDatosPaso1(publicacionProvider) ||
      _currentStep == 1 &&
      Paso2State.validarDatosPaso2(publicacionProvider) ||
      _currentStep == 2 &&
      Paso3State.validarDatosPaso3(publicacionProvider);

    return Stepper(
      currentStep: _currentStep,
      onStepContinue: () {
        setState(() {
          if (_currentStep < 2) {
            _currentStep += 1;
          }
        });
      },
      onStepCancel: () {
        setState(() {
          if (_currentStep > 0) {
            _currentStep -= 1;
          }
        });
      },
      steps: [
        Step(
          title: Text('Paso 1'),
          isActive: _currentStep == 0,
          state:
            _currentStep == 0 ? StepState.editing : StepState.indexed,
          content: Paso1(),
        ), // Step
        Step(
          title: Text('Paso 2'),
          isActive: _currentStep == 1,
          state:
            _currentStep == 1 ? StepState.editing : StepState.indexed,
          content: Paso2(),
        ), // Step
        Step(
          title: Text('Paso 3'),
          isActive: _currentStep == 2,
          state:
            _currentStep == 2 ? StepState.editing : StepState.indexed,
          content: Paso3(),
        ), // Step
      ],
    ),
  ),
),
```

## Anexo 46. Manejadores de eventos de los botones.

```
controlsBuilder:
  (BuildContext context, ControlsDetails controlsDetails) {
    if (_currentStep == 2) {
      return Row(
        children: [
          ElevatedButton(
            onPressed: () async {
              final usuarioProvider =
                Provider.of<UserProvider>(context, listen: false);
              publicacionProvider.publicacion.uidUsuario =
                usuarioProvider.user!.uid!;
              Paso3State.guardarDatosPaso3(publicacionProvider);
              PublicacionService publicacionService =
                PublicacionService();
              final pdfService = PdfStorageService();

              List<ImagenModelo> imagenes = await publicacionService
                .subirImagenesAFirestorage(
                  publicacionProvider.publicacion.imagenes);
              final pdfUrl =
                await pdfService.uploadPdf(publicacionProvider.publicacion.contrato!);
              bool correcto = await publicacionService
                .crearPublicacion(publicacionProvider, imagenes, pdfUrl!);
              if (correcto) {
                limpiarCampos();
                publicacionProvider.resetPublicacion();
                _mostrarRegistroExitoso(
                  'Registro guardado exitosamente');
              } else {
                print('error en la insercion');
              }
            },
            child: Text('Guardar'),
          ), // ElevatedButton
          SizedBox(width: 16),
          ElevatedButton(
            onPressed: () {
              controlsDetails.onStepCancel!();
            },
            child: Text('atrás'),
          ), // ElevatedButton
        ],
      ); // Row
    }
  }
```

#### Anexo 47. Función para restablecer o limpiar los campos.

```
void _limpiarCampos() {
    Paso1State.amoblado = false;
    Paso1State.tamanoController.text = '';
    Paso1State.precioController.text = '';
    Paso1State.cocinaCompartida = false;
    Paso1State.cocinaIndividual = false;
    Paso2State.agua = false;
    Paso2State.luz = false;
    Paso2State.internet = false;
    Paso2State.parqueadero = false;
    Paso2State.mesesGarantia = '';
    Paso3State.permitirMascotas = false;
    Paso3State.preferenciaGenero = '';
    Paso3State.descripcionController.text = '';
}
```

#### Anexo 48. Función para mostrar el registro exitoso.

```
ScaffoldMessenger.of(context).showSnackBar(snackBar).closed.then((_) {  
  Navigator.pushReplacement(  
    context,  
    MaterialPageRoute(builder: (context) => HomePage()),  
  );  
});
```

## Anexo 49. Recorrer publicaciones del arrendador.

```
mapa_MostrarLista.dart  lista_pagina.dart 1, M X
o > paginas > lista_pagina.dart > _listaPageState > initState
13 class listaPageState extends State<listaPage> {
14   List<MisPublicaciones> publicaciones = [];
15   UserService userService = UserService();
16
17   @override
18   void initState() {
19     super.initState();
20     recorrerPublicacionesUsuario();
21   }
22
23 > Future<void> confirmarEliminacion(String idPublicacion) async {...
71
72 Future<void> recorrerPublicacionesUsuario() async {
73   try {
74     UserProvider userProvider =
75     | | Provider.of<UserProvider>(context, listen: false);
76     String uidUsuario = userProvider.user?.uid ?? '';
77
78     List<MisPublicaciones> result =
79     | | await userService.getPublicacionesUsuario(uidUsuario);
80     setState(() {
81     | | publicaciones = result;
82     | | });
83   } catch (error) {
```



## Anexo 50. Método para obtener las publicaciones del usuario.

```
lib > servicios > user.service.dart > UserService > getPublicacionesUsuario
97 // MÉTODO PARA OBTENER LAS PUBLICACIONES DEL USUARIO
98
99 Future<List<MisPublicaciones>> getPublicacionesUsuario(
100     String uidUsuario) async {
101     try {
102         // Obtener la referencia a la colección "publicaciones" en Firebase
103         final collectionRef =
104             FirebaseFirestore.instance.collection('publicaciones');
105
106         // Consulta para obtener las publicaciones de un usuario específico
107         final querySnapshot =
108             await collectionRef.where('uidUsuario', isEqualTo: uidUsuario).get();
109
110         final publicaciones = querySnapshot.docs.map((doc) {
111             final data = doc.data();
112
113             return MisPublicaciones(
114                 id: data['id'],
115                 uidUsuario: data['uidUsuario'],
116                 urlPDFContrato: data['contrato'],
117                 imagenes: List<ImagenesMiPublicacion>.from(data['imagenes']
118                     .map((imagen) => ImagenesMiPublicacion.fromJson(imagen))), // List.from
119                 descripcion: data['descripcion'],
120                 habitacion: HabitacionModelo.fromJson(data['habitacion']),
121                 ubicacion: UbicacionModel.fromJson(data['ubicacion']),
122             ); // MisPublicaciones
123         }).toList();
124
125         return publicaciones;
126     } catch (e) {
127         print('Error al obtener las publicaciones: $e');
128         return [];
129     }
130 }
```

## Anexo 51. Diálogo de confirmación.

```
lib > paginas > lista_pagina.dart > _listaPageState > confirmarEliminacion
22
23 Future<void> confirmarEliminacion(String idPublicacion) async {
24   return showDialog(
25     context: context,
26     builder: (BuildContext context) {
27       return AlertDialog(
28         title: Text('Confirmar eliminación'),
29         content: Text('¿Estás seguro de eliminar esta publicación?'),
30         actions: [
31           TextButton(
32             child: Text('Cancelar'),
33             onPressed: () {
34               Navigator.of(context).pop();
35             },
36           ), // TextButton
37           TextButton(
38             child: Text('Eliminar'),
39             onPressed: () async {
40               bool borrado =
41                 await userService.deletePublicacionesUsuario(idPublicacion);
42               setState(() {
43                 recorrerPublicacionesUsuario();
44               });
45               if (borrado)
46                 ScaffoldMessenger.of(context).showSnackBar(
47                   SnackBar(
48                     content: Text('Publicacion borrada con éxito.'),
49                     duration: Duration(seconds: 2), // Duración del SnackBar
50                   ), // SnackBar
51                 );
52               else {
53                 ScaffoldMessenger.of(context).showSnackBar(
54                   SnackBar(
55                     content:
56                       Text('Ocurrió un error al eliminar la publicación.'),
57                     duration: Duration(seconds: 2), // Duración del SnackBar
58                   ), // SnackBar
59                 );
60             }
61             Navigator.of(context).pop();
62           },

```

## Anexo 52. Método para borrar las publicaciones del usuario.

```
mapa_MostrarLista.dart  lista_pagina.dart 1, M  user.service.dart 6, M X
> servicios > user.service.dart > UserService > deletePublicacionesUsuario
32 // MÉTODO PARA BORRAR LAS PUBLICACIONES DEL USUARIO
33
34 Future<bool> deletePublicacionesUsuario(String publicacionId) async {
35   try {
36     // Obtiene la referencia a la colección "publicaciones" en Firebase
37     final collectionRef =
38       FirebaseFirestore.instance.collection('publicaciones');
39
40     // Obtiene la referencia al documento de la publicación específica
41     final docRef = collectionRef.doc(publicacionId.toString());
42
43     // Verifica si la publicación pertenece al usuario antes de eliminarla
44     final snapshot = await docRef.get();
45     if (snapshot.exists) {
46       // Elimina la publicación del usuario
47       await docRef.delete();
48       return true;
49     } else {
50       return false;
51     }
52   } catch (e) {
53     return false;
54   }
55 }
```

## Anexo 53. Métodos para editar la publicación del arrendador.

```
lib > paginas > publicar > editar_publicacion.dart > ...
44   bool _descripcionError = false;
45
46   ///////////////////////////////////////////////////
47
48   LatLng? _selectedLocation;
49
50   bool seActualizoPDF = false;
51   bool seActualizoUbicacion = false;
52   String urlPDFActualizado = '';
53   String urlPDF = '';
54   String nombrePDF = '';
55   double? _latitud;
56   double? _longitud;
57
58   List<File> images = [];
59   List<ImagenesMiPublicacion> misImagenesPublicacion = [];
60   List<File> updatedImages = [];
61   //METODO PARA ABRIR EL PDF EN UN NAVEGADOR
62
63 > void abrirPDF() async { ...
122
123   @override
124 > void initState() { ...
151
152 > void _agregarImagen() async { ...
192
193 > String obtenerNombreDocumento(String url) { ...
202
203 > Future<void> guardarCambios() async { ...
310
311 > Future<void> _actualizarPDF() async { ...
329
330   @override
331 > Widget build(BuildContext context) { ...
703
704
705 > Future<bool> _mostrarDialogoConfirmacionSalir(BuildContext context) async { ...
```

## Anexo 54. Visualizar y actualizar PDF.

```
editar_publicacion.dart 9, M x
lib > paginas > publicar > editar_publicacion.dart > _EditarPublicacionState > build
348 style: TextStyle(fontWeight: FontWeight.bold),
349 ), // Text
350 TextFormField(
351   controller: _tamañoController,
352   decoration: InputDecoration(
353     hintText: 'Ingrese tamaño',
354     errorText: _tamañoError ? 'Campo Obligatorio' : null,
355     errorStyle: TextStyle(color: Colors.red),
356   ), // InputDecoration
357 ), // TextFormField
358 Text(
359   'Precio:',
360   style: TextStyle(fontWeight: FontWeight.bold),
361 ), // Text
362 TextFormField(
363   controller: _precioController,
364   decoration: InputDecoration(
365     hintText: 'Ingrese precio ',
366     errorText: _precioError ? 'Campo Obligatorio' : null,
367     errorStyle: TextStyle(color: Colors.red),
368   ), // InputDecoration
369 ), // TextFormField
370 SizedBox(height: 16.0),
371 Center(
372   child: Column(
373     mainAxisAlignment: MainAxisAlignment.center,
374     children: [
375       ListTile(
376         leading: Image.asset('assets/icono_pdf.png'),
377         title: Text(nombrePDF),
378         onTap: abrirPDF,
379       ), // ListTile
380       ElevatedButton(
381         child: Text('Actualizar PDF'),
382         onPressed: _actualizarPDF,
```

## Anexo 55. Visualizar el PDF de contrato de arriendo.

```
editar_publicacion.dart 9, M x
lib > paginas > publicar > editar_publicacion.dart > _EditarPublicacionState > abrirPDF
62
63 void abrirPDF() async {
64   String urlPDF;
65   if (seActualizoPDF == true) {
66     urlPDF = urlPDFActualizado;
67
68     if (urlPDF.isNotEmpty) {
69       Navigator.push(
70         context,
71         MaterialPageRoute(
72           builder: (context) => PDFViewer(filePath: urlPDF),
73         ), // MaterialPageRoute
74       );
75     } else {
76       showDialog(
77         context: context,
78         builder: (BuildContext context) {
79           return AlertDialog(
80             title: Text('Error'),
81             content: Text('No se puede abrir el PDF.'),
82             actions: [
83               TextButton(
84                 child: Text('Aceptar'),
85                 onPressed: () {
86                   Navigator.of(context).pop();
87                 },
88               ), // TextButton
89             ],
90           ); // AlertDialog
91         },
92       );
93     }
94   } else {
95     urlPDF = widget.publicacion.urlPDFContrato;
96     final pdfUrl = Uri.encodeFull(urlPDF);
97     final webUrl = 'https://docs.google.com/viewer?url=$pdfUrl';
98   }
```

## Anexo 56. Actualizar el PDF de contrato de arriendo.

```
Future<void> _actualizarPDF() async {  
  // Seleccionar un archivo PDF  
  final FilePickerResult? result = await FilePicker.platform.pickFiles(  
    type: FileType.custom,  
    allowedExtensions: ['pdf'],  
  );  
  
  if (result != null) {  
    final File file = File(result.files.single.path!);  
    final fileName = result.files.single.name;  
    setState(() {  
      seActualizoPDF = true;  
      nombrePDF = fileName;  
      urlPDFActualizado = file.path;  
      urlPDFBorrar = urlPDF;  
    });  
  }  
}
```

## Anexo 57. Actualizar cuarto amoblado y cocina.

```
editar_publicacion.dart 9, M x
lib > paginas > publicar > editar_publicacion.dart > _EditarPublicacionState > _actualizarPDF
388 ), // Center
389 SizedBox(height: 16.0),
390 CheckboxListTile(
391   title: Text('Amoblado'),
392   value: _amoblado,
393   onChanged: (value) {
394     setState(() {
395       _amoblado = value!;
396     });
397   },
398 ), // CheckboxListTile
399 RadiolistTile(
400   title: Text('Cocina Individual'),
401   value: 'Cocina Individual',
402   groupValue: preferenciaCocina,
403   onChanged: (value) {
404     setState(() {
405       preferenciaCocina = value!;
406     });
407   },
408 ), // RadiolistTile
409 RadiolistTile(
410   title: Text('Cocina Compartida'),
411   value: 'Cocina Compartida',
412   groupValue: preferenciaCocina,
413   onChanged: (value) {
414     setState(() {
415       preferenciaCocina = value!;
416     });
417   },
418 ), // RadiolistTile
```



## Anexo 58. Adjuntar imágenes de cuarto de arriendo.

```
void _agregarImagen() async {
  final ImagePicker _picker = ImagePicker();
  if (misImagenesPublicacion.length > 4) {
    showDialog(
      context: context,
      builder: (BuildContext context) {
        return AlertDialog(
          title: Text('Limite alcanzado'),
          content: Text('Ya has adjuntado el máximo de 4 imágenes.'),
          actions: [
            TextButton(
              child: Text('Aceptar'),
              onPressed: () {
                Navigator.of(context).pop();
              },
            ), // TextButton
          ],
        ); // AlertDialog
      },
    );
  }
  return;
}

final publicacionService = PublicacionService();
final XFile? image = await _picker.pickImage(source: ImageSource.gallery);
if (image != null) {
  final File file = File(image.path);
  publicacionService.comprimirImagen(file).then((XFile imagenComprimida) {
    setState(() {
      images.add(File(imagenComprimida.path));
      updatedImages.add(File(imagenComprimida.path));
      misImagenesPublicacion.add(
        ImagenesMiPublicacion(
          imagenUrl: imagenComprimida.path, isLocal: true),
      );
    });
  });
}
```

## Anexo 59. Método para comprimir imágenes.

```
Future<XFile> comprimirImagen(imagen) async {
  final Directory directorioTemp = await getTemporaryDirectory();
  final String rutaTempImagen = "${directorioTemp.path}/imagen_temp.jpg";

  final result = await FlutterImageCompress.compressAndGetFile(
    imagen.path,
    rutaTempImagen,
    quality: 85,
    minWidth: 800,
    minHeight: 200,
  );

  return XFile(result!.path);
}
```

## Anexo 60. Detalle de la publicación en el mapa.

```
lib > paginas > inicio > MapSample.dart > MapSampleState > _mostrarDetallesPublicacion
155 void _mostrarDetallesPublicacion(MisPublicaciones publicacion) {
156   showDialog(
157     context: context,
158     builder: (BuildContext context) {
159       return AlertDialog(
160         title: Text('Información de la publicación'),
161         content: SingleChildScrollView(
162           child: Column(
163             mainAxisAlignment: MainAxisAlignment.min,
164             children: [
165               publicacion.imagenes.isNotEmpty
166                 ? Image.network(publicacion.imagenes[0].imagenUrl,
167                   errorBuilder: (context, error, stackTrace) {
168                     // Manejar el error 404 aquí
169                     return Icon(
170                       Icons
171                         .image_not_supported, // Icono de imagen no disponible
172                       size: 150,
173                       color: Colors.grey, // Color del icono
174                     ); // Icon
175                   }, fit: BoxFit.fitWidth, height: 200) // Image.network
176                 : Text(
177                   'Imagen no disponible',
178                   style: TextStyle(
179                     color: Colors.white,
180                     fontSize: 16,
181                   ), // TextStyle
182                 ), // Text
183               Text('Descripción: ${publicacion.descripcion}'),
184               Text('Precio: ${publicacion.habitacion.precio}'),
185               Text('Tamaño: ${publicacion.habitacion.tamano}'),
186             ],
187           ), // Column
188         ), // SingleChildScrollView
189         actions: <Widget>[
190           IconButton(
191             icon: Icon(Icons.send),
192             onPressed: () async {
193               LatLng currentLocation = await obtenerUbicacionActual();
194               LatLng selectedUbicacion = LatLng(
195                 double.parse(publicacion.ubicacion.latitud),
196                 double.parse(publicacion.ubicacion.longitud),
197               ); // LatLng
198               Navigator.of(context).pop();
199             },
200             List<LatLng> coordinates =
201               await getDirections(currentLocation, selectedUbicacion);
202             // ...
203           ), // ...

```

## Anexo 61. Funciones de los iconos envío, información y cerrar.

```
mapaSolo.dart 6, M X main.dart 2
ib > paginas > inicio > mapaSolo.dart > MapaPruebaState > build
214 actions: <Widget>[
215   IconButton(
216     icon: Icon(Icons.send),
217     onPressed: () async {
218       getCurrentLocation();
219       LatLng selectedUbicacion = LatLng(
220         double.parse(publicacion.ubicacion.latitud),
221         double.parse(publicacion.ubicacion.longitud),
222       ); // LatLng
223       Navigator.of(context).pop();
224
225       List<LatLng> coordinates = await getDirections(
226         currentLocation, selectedUbicacion);
227
228       setState(() {
229         polylines = [
230           Polyline(
231             points: coordinates,
232             color: Colors.blue,
233             strokeWidth: 3.0,
234           ), // Polyline
235         ];
236       });
237     },
238   ), // IconButton
239   IconButton(
240     icon: Icon(Icons.info),
241     onPressed: () {
242       Navigator.of(context).pop();
243       Navigator.push(
244         context,
245         MaterialPageRoute(
246           builder: (context) =>
247             reconstruccion_publicacion(
248               publicacion: publicacion,
249             ), // reconstruccion_publicacion
250         ), // MaterialPageRoute
251       );
252     },
253   ), // IconButton
254   ElevatedButton(
255     child: Text('Cerrar'),
256     onPressed: () {
257       Navigator.of(context).pop();
258       setState(() {
259         selectedMarker = null;
260       });
261     },

```

## Anexo 62. Obtener lista de arriendos favoritos.

```
void obtenerMisFavoritos() async {
  AuthService authService = AuthService();
  String uid = authService.getCurrentUser().uid;
  publicacionService = PublicacionService();
  misFavoritos = await publicacionService.obtenerMisFavoritos(uid);
  if (mounted)
    setState(() {
      misFavoritos = misFavoritos;
    });
}
```

### Anexo 63. Eliminar lista de arriendos de favoritos.

```
Future<bool> eliminarDeMisFavoritos(String id) async {
  try {
    final usuariosCollection =
      Firestore.instance.collection('usuarios');
    final querySnapshot = await usuariosCollection.get();

    for (final doc in querySnapshot.docs) {
      final misFavoritosCollection = doc.reference.collection('misFavoritos');
      final querySnapshot =
        await misFavoritosCollection.where('id', isEqualTo: id).get();

      if (querySnapshot.docs.isNotEmpty) {
        await querySnapshot.docs.first.reference.delete();
        return true;
      }
    }

    return false; // No se encontró el documento con el ID especificado
  } catch (e) {
    print('Error al eliminar de mis favoritos: $e');
    return false;
  }
}
```

## Anexo 64. Modelo para filtrar publicaciones.

```
misFavoritos.dart 1, M  filtro.model.dart M x  filtro_pagina.dart 3, M
lib > providers > filtro.model.dart > FilterModel
1  import 'package:flutter/foundation.dart';
2
3  class FilterModel extends ChangeNotifier {
4      double precioMaximo = 500;
5      String generoSeleccionado = '';
6      bool aceptaMascotas = false;
7      double distanciaMaxima = 10;
8
9      void actualizarPrecioMaximo(double value) {
10         precioMaximo = value;
11         notifyListeners();
12     }
13
14     void actualizarGeneroSeleccionado(String value) {
15         generoSeleccionado = value;
16         notifyListeners();
17     }
18
19     void actualizarAceptaMascotas(bool value) {
20         aceptaMascotas = value;
21         notifyListeners();
22     }
23
24     void restablecerFiltros() {
25         precioMaximo = 500;
26         generoSeleccionado = '';
27         aceptaMascotas = false;
28         notifyListeners();
29     }
30
31     bool tieneFiltrosSeleccionados() {
32         return precioMaximo != 500 ||
33             generoSeleccionado.isNotEmpty ||
34             aceptaMascotas;
35     }
36 }
```

## Anexo 65. Obtención de coordenada de la ruta.

```
Future<List<LatLng>> getDirections(LatLng origin, LatLng destination) async {
  final String apiKey =
    // Reemplaza con tu API Key de Google

  final String url =
    'https://maps.googleapis.com/maps/api/directions/json?origin=${origin.latitude},${origin.longitude}&destination=${destination.latitude},${destination.longitude}&key=$apiKey';

  final response = await http.get(Uri.parse(url));

  if (response.statusCode == 200) {
    final decoded = json.decode(response.body);

    List<LatLng> coordinates = [];
    List<dynamic> steps = decoded['routes'][0]['legs'][0]['steps'];

    coordinates.add(origin);

    for (var step in steps) {
      double startLat = step['start_location']['lat'];
      double startLng = step['start_location']['lng'];

      coordinates.add(LatLng(startLat, startLng));

      double endLat = step['end_location']['lat'];
      double endLng = step['end_location']['lng'];

      coordinates.add(LatLng(endLat, endLng));
    }

    coordinates.add(destination);

    return coordinates;
  } else {
    throw Exception('Failed to load directions');
  }
}
```



## Anexo 66. Obtener la ubicación actual del dispositivo.

```
Future<LatLng> obtenerUbicacionActual() async {  
  bool servicioActivo = await Geolocator.isLocationServiceEnabled();  
  if (servicioActivo) {  
    Position posicion = await Geolocator.getCurrentPosition(  
      desiredAccuracy: LocationAccuracy.high,  
    );  
    setState(() {  
      currentLocation = LatLng(posicion.latitude, posicion.longitude);  
    });  
    return LatLng(posicion.latitude, posicion.longitude);  
  } else {  
    return LatLng(0, 0);  
  }  
}
```

## Anexo 67. Agregar Marcadores a las publicaciones.

```
void agregarMarcadores(List<MisPublicaciones> publicaciones) {  
    for (var publicacion in publicaciones) {  
        if (publicacion.ubicacion != null) {  
            LatLng ubicacion = LatLng(  
                double.parse(publicacion.ubicacion.latitud),  
                double.parse(publicacion.ubicacion.longitud),  
            );  
            Marker marker = Marker(  
                markerId: MarkerId(ubicacion.toString()),  
                position: ubicacion,  
                onTap: () {  
                    _mostrarDetallesPublicacion(publicacion);  
                },  
            ); // Marker  
            _markers.add(marker);  
        }  
    }  
}
```

## **Anexo 67**

Manual de usuario

### **“APLICACIÓN MÓVIL USANDO TECNOLOGÍAS MULTIPLAFORMA PARA LA DIFUSIÓN Y BÚSQUEDA DE ARRIENDOS UNIVERSITARIOS EN LOS ALREDEDORES DE LA UNIVERSIDAD TÉCNICA DE AMBATO CAMPUS HUACHI”**

MANUAL DE USUARIO

El presente manual brinda una guía completa sobre el uso de la aplicación móvil destinada para la búsqueda y difusión de arriendos universitarios. Este manual tiene como objetivo proporcionar instrucciones detalladas para el uso correcto de la aplicación móvil.

## Registro e inicio de sesión del usuario

The screenshot shows a registration form with the following fields: 'Arrendador' (a dropdown menu), 'Nombre', 'Apellido', 'Celular', 'Correo', 'Contraseña', and 'Confirmar contraseña'. A blue 'Registrar' button is located at the bottom of the form.

### Roles:

- Arrendador.
- Estudiante (arrendatario)

El usuario necesita registrarse antes de iniciar sesión. El registro posee dos roles, tanto arrendador para publicar cuartos de arriendo, como arrendatarios para buscar alojamiento.

**Botón “Registrar”:** una vez completado los campos requeridos se puede registrar el nuevo usuario.

The screenshot shows a login screen with a white card containing two input fields: 'Correo' (with an envelope icon) and 'Contraseña' (with a lock icon and a visibility toggle). Below the card are two blue buttons: 'Iniciar sesión' and 'Registrarse'.

Con la cuenta creada se debe ingresar el correo y contraseña para iniciar sesión.

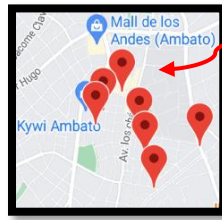
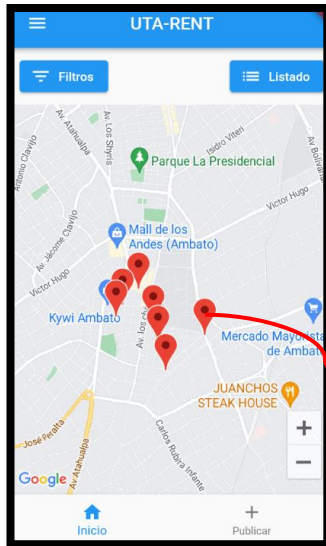
El usuario puede visualizar la contraseña si así lo desea, caso contrario la misma permanecerá protegida.

**Botón “Iniciar sesión”:** si el usuario y contraseña son correctos se iniciará la sesión.

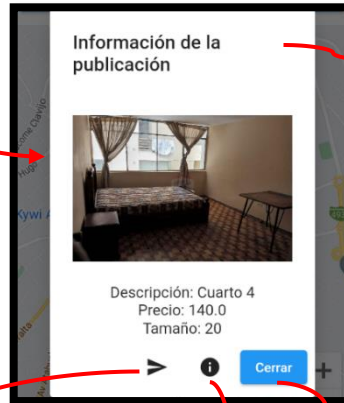
**Botón “Registrarse”:** para crear una cuenta nueva.

## Funcionalidad arrendadores/administradores

### Interacción con los marcadores del mapa

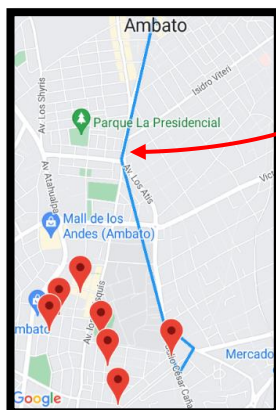


En el centro del mapa se muestran las habitaciones disponibles mediante marcadores.



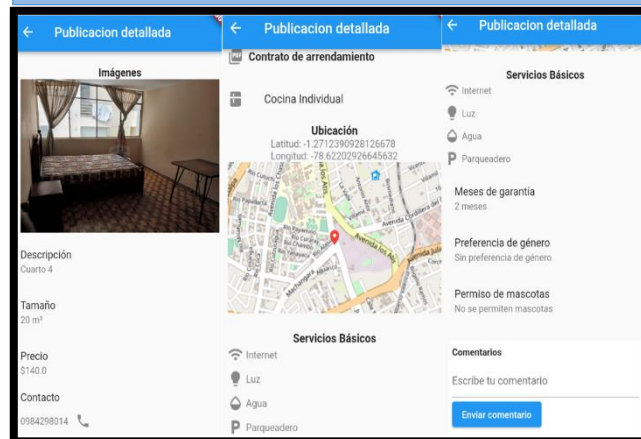
**Marcadores:** Al presionarlos se visualiza un detalle breve de la habitación con su descripción, precio y tamaño.

**Botón "Cerrar":** desaparece el detalle del cuarto de arriendo.

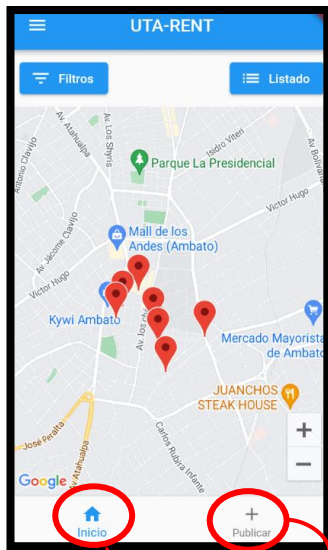


**Botón de ruta:** Al presionarlo, el usuario visualizará la ruta desde su posición actual hasta el marcador seleccionado.

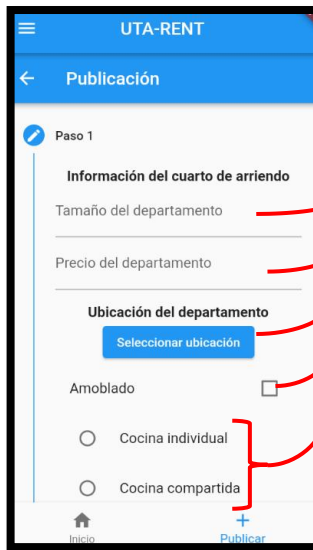
**Botón de información:** Al presionarlo se abre la descripción completa del cuarto de arriendo, incluido el contrato de arriendo, servicios básicos, contactos, y más.



## Publicación de un cuarto de arriendo



**Botón “Inicio”:**  
dirige al usuario al mapa con la lista arriendos identificados por



**Botón “Publicar”:**  
dirige al usuario a los pasos de publicación.

**Publicación (paso 1):** la información del cuarto de arriendo debe ser completada con los siguientes detalles:

- Tamaño
- Precio
- Ubicación
- Si es amoblado
- Si posee cocina individual o compartida.

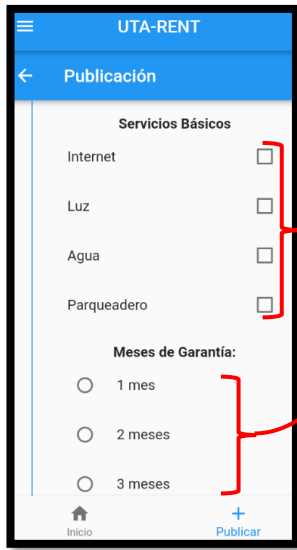


Se visualiza las imágenes agregadas.

**Publicación (paso 1):**

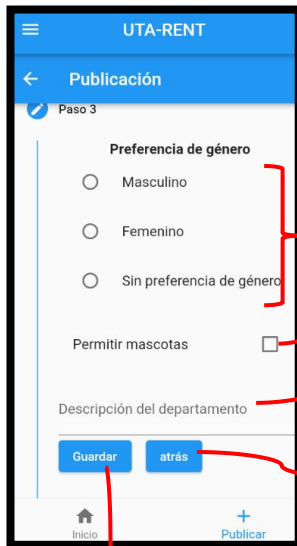
- Contrato de arriendo en formato PDF.
- Imágenes agregadas (máximo 4)

**Botón “Continuar”:**  
dirige al paso 2 de la publicación del arriendo.



**Publicación (paso 2):** se debe completar los siguientes detalles:

- Servicios básicos (internet, luz, agua, parqueadero).
- Meses de garantía (1, 2 o 3 meses).



**Publicación (paso 3):** se debe completar los siguientes detalles:

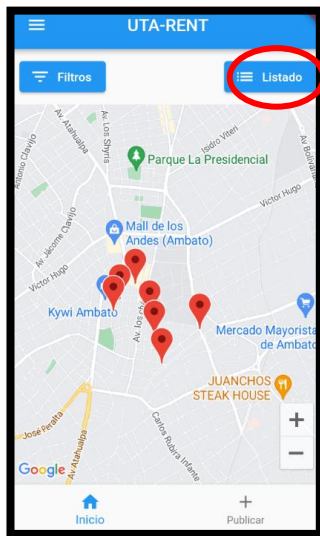
- Preferencia de género (masculino, femenino, sin preferencia de género).
- Permiso de mascotas.
- Descripción del departamento.

**Botón “Atrás”:**

regresa al paso 2 de la publicación del arriendo.

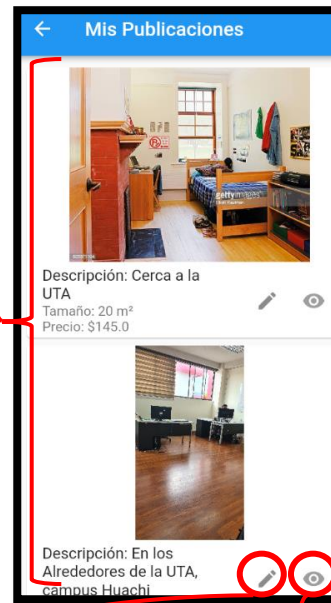
**Botón “Guardar”:** se guarda la publicación en la lista de publicaciones.

## Lista de arriendos que el usuario (arrendador) ha publicado.



**Botón listado:** muestra las publicaciones propias del usuario que ha realizado.

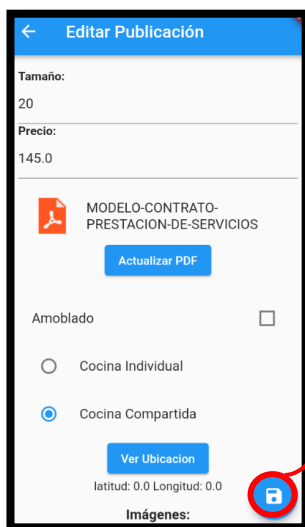
Si el usuario tiene publicaciones, se mostrarán en la lista.



Si el usuario no tiene publicaciones, se muestra el mensaje “No hay publicaciones disponibles”



**Botón habilitar:** El usuario puede habilitar o deshabilitar la publicación para que sea visible en el mapa.



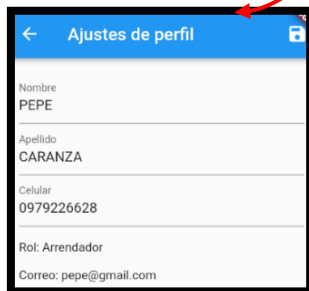
**Botón editar:** redirige a la publicación del usuario la cual puede ser editada en su totalidad.

**Botón guardar:** guarda los cambios editados de la publicación.

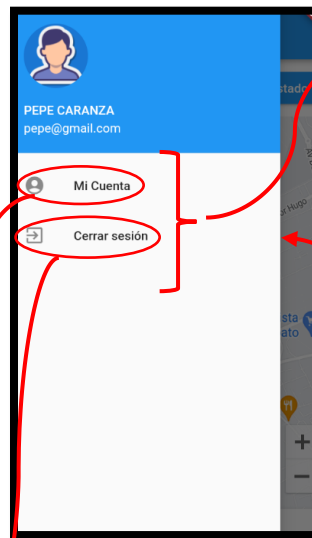


## Barra lateral con filtros y ajustes de perfil.

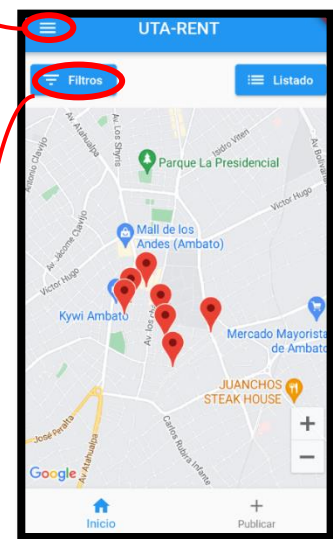
**Ajustes de perfil:** el usuario puede editar campos personales como el nombre, apellido y teléfono. Además, muestra el rol al que pertenece y el correo con el que se ha registrado.



**Botón cerrar sesión:** Cierra la sesión actual del usuario.

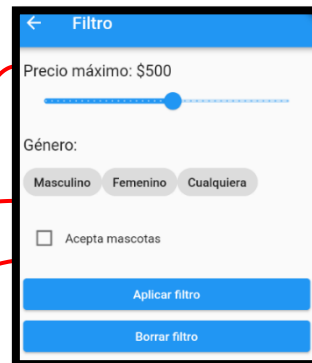


**Barra lateral:** Se despliega la cuenta del usuario y la opción para cerrar la sesión actual.



### Filtros por:

- Precio máximo.
- Género.
- Permiso de mascotas.



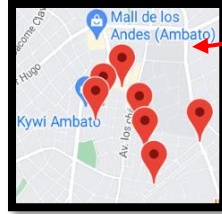
**Botón aplicar filtro:** aplica el filtro en la lista de arriendos del mapa.

**Filtro:** realiza la búsqueda de acuerdo con las necesidades del arrendatario.

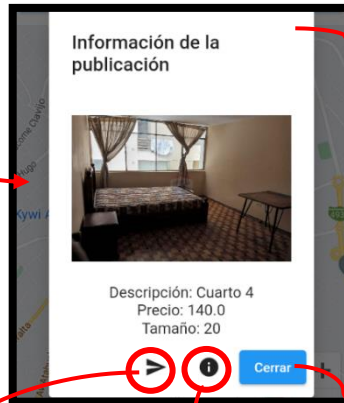
**Botón borrar filtro:** borra el filtro en la lista de arriendos del mapa.

## Funcionalidad arrendatarios/estudiantes

### Interacción con los marcadores del mapa

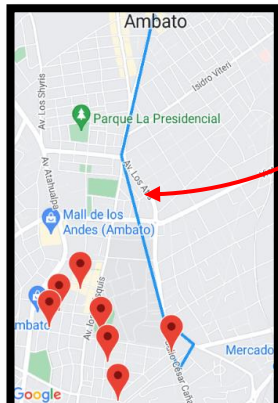


En el centro del mapa se muestran las habitaciones disponibles mediante **marcadores**.



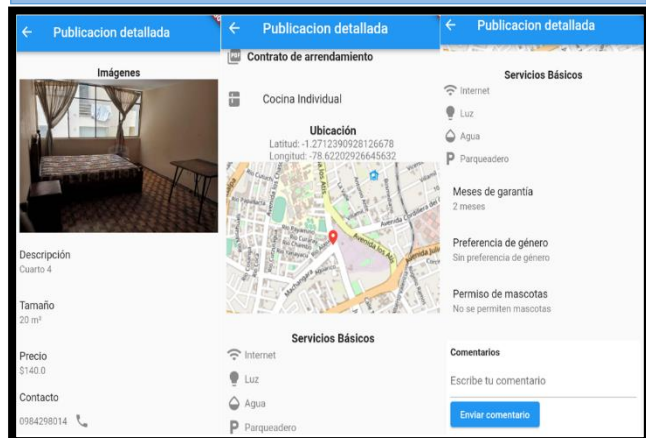
**Marcadores:** Al presionarlos se visualiza un detalle breve de la habitación con su descripción, precio y tamaño.

**Botón "Cerrar":** desaparece el detalle del cuarto de arriendo.



**Botón de ruta:** Al presionarlo, el usuario visualizará la ruta desde su posición actual hasta el marcador seleccionado.

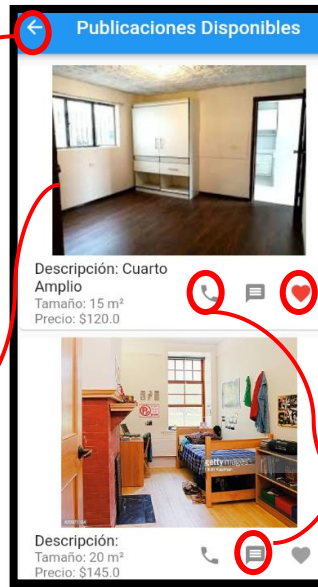
**Botón de información:** Al presionarlo se abre la descripción completa del cuarto de arriendo, incluido el contrato de arriendo, servicios básicos, contactos, y más.



## Lista de todos los cuartos de arriendo disponibles

**Botón atrás:** retorna al mapa con los marcadores en la lista de publicaciones.

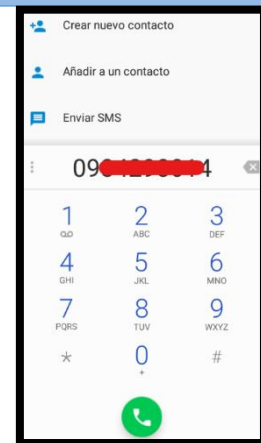
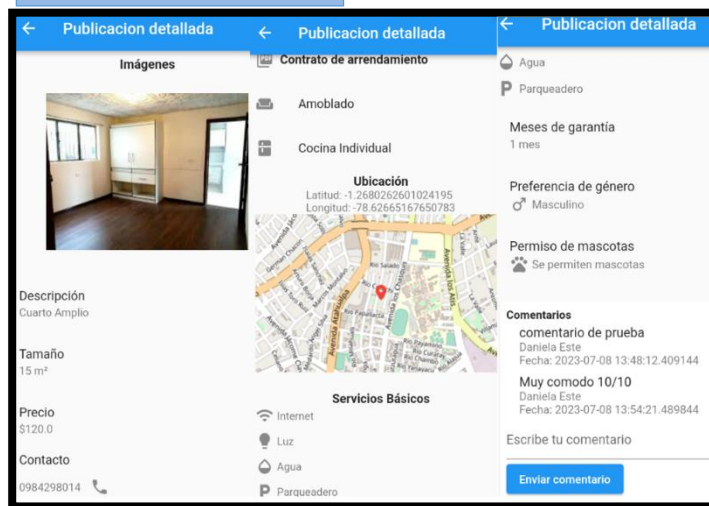
**Clic en la imagen:** Al presionar la imagen se abre la descripción completa del cuarto de arriendo, incluido el contrato de arriendo, servicios básicos, contactos, y más.



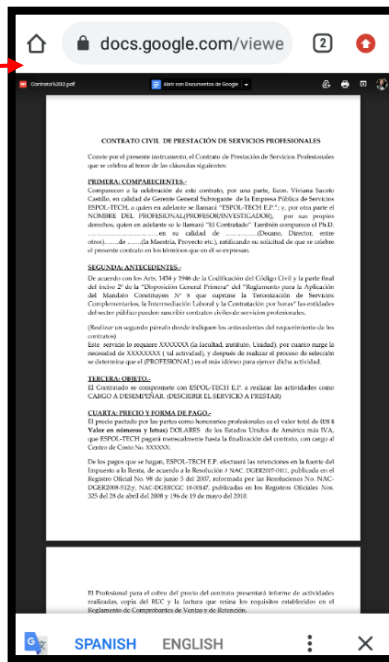
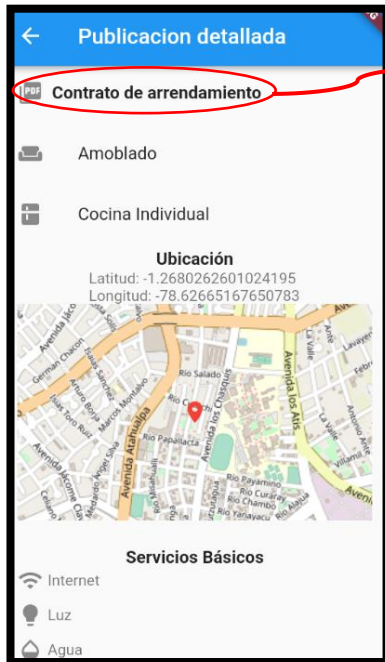
**Botón favorito:** el estudiante puede agregar el departamento a su lista de favoritos para un acceso rápido.

**Botón de contacto:** el estudiante puede contactarse con el arrendador vía whatsapp.

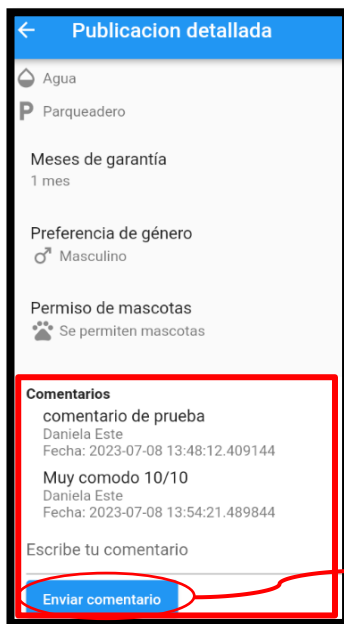
**Botón llamar:** el estudiante puede contactarse con el arrendador vía telefónica.



## Visualización de contrato de arriendo y agregar comentarios.



**Botón contrato:** el estudiante puede visualizar en el navegador el contrato de arriendo y descargarlo si así lo desea.



**Botón "Comentario":** el estudiante puede aportar con un comentario sobre el arriendo para referencia a otros estudiantes.

## Filtros para las publicaciones en el mapa

**Filtros por:**

- Precio máximo.
- Género.
- Permiso de mascotas.

**Filtro**

Precio máximo: \$500

Género:

Masculino Femenino Cualquiera

Acepta mascotas

Aplicar filtro

Borrar filtro

**Botón “Aplicar filtro”:** aplica el filtro en la lista de arriendos del mapa.

**Botón “Borrar filtro”:** borra el filtro en la lista de arriendos del mapa.

## Barra lateral con filtros y ajustes de perfil.

The diagram illustrates a mobile application interface with a side menu. The central element is a smartphone screen showing a user profile for Daniela Estefanía García Cherres. The side menu is open, displaying three options: 'Favoritos', 'Mi Cuenta', and 'Cerrar sesión'. Three callout boxes provide detailed explanations of these features:

- Botón favorito:** muestra las publicaciones que se han añadido como favoritos.
- Barra lateral:** Se despliega la lista de favoritos, la cuenta del usuario y la opción para cerrar la sesión actual.
- Botón cerrar sesión:** Cierra la sesión actual del estudiante.

Below the main interface, a second callout box explains the profile settings screen:

- Ajustes de perfil:** el estudiante puede editar campos personales como el nombre, apellido y teléfono. Además, muestra el rol al que pertenece y el correo con el que se ha registrado.

The 'Ajustes de perfil' screen displays the following information:

Nombre	Daniela Estefanía
Apellido	García Cherres
Celular	0984298014
Rol:	Estudiante
Correo:	danielagarciache@hotmail.com