



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Searchity: una aplicación móvil para la ayuda en la toma de
decisión de escoger una carrera universitaria

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Renau López, Juan Pablo

Tutor/a: Andrés Martínez, David de

CURSO ACADÉMICO: 2021/2022

Resumen

Una de las primeras decisiones importantes a las que se enfrentan las personas en su vida es qué van a estudiar en la universidad o, más importante aún, qué camino profesional quieren tomar. Durante la educación secundaria y el bachillerato los estudiantes caminan con pies de plomo, han ido tomando decisiones, pero han sido más pequeñas, sin mucho impacto, como qué rama escoger (si ciencias o letras), qué optativas te gustan más, etc. Además, en estas decisiones si se presentaba alguna duda se tenía la suerte de contar con profesores que te podían explicar cómo iban a ser las asignaturas de su rama. Sin embargo, cuando se termina el bachillerato no se sabe muy bien por dónde continuar. Hay gente que es afortunada y sabe qué estudiar, ya que lo ha tenido siempre claro, pero para la mayoría es una decisión con muchas incógnitas y pocas respuestas. Asimismo, hay personas que tienen suerte y conocen a alguien que ha estudiado lo mismo que ellos quieren estudiar, pero son casos puntuales. Por otra parte, existen demasiados grados y dobles títulos en distintas universidades y la única información disponible es el listado de asignaturas que tienen en sus respectivos sitios web.

Ante este problema nace Searchity, una red social que conecta a los preuniversitarios, jóvenes que están buscando una carrera, con los universitarios para que puedan resolver todas sus dudas acerca de la elección del grado universitario. La aplicación cuenta con dos funcionalidades diferentes pero complementarias para cumplir con esta misión. La primera y principal es que permite a los preuniversitarios establecer un chat con un universitario del grado y de la universidad en los que tengan interés. En este chat el preuniversitario podrá tener una conversación con el universitario, quien le responderá a todas sus dudas. La segunda es un foro dividido en universidades y grados en el que se pueden ver opiniones de otros usuarios acerca de las asignaturas, los profesores, los programas Erasmus y las prácticas entre otros. De esta manera, Searchity ofrece una fuente de información muy útil que no está disponible en ningún otro lugar para que ayude a los preuniversitarios en la toma de su primera gran decisión.

Palabras clave: Aplicación Móvil, Android, Java, Firebase, Universidad, Carrera, Grado, Asignaturas, Profesores, Prácticas, Erasmus



Abstract

One of the first important decisions people face in their life is what they are going to study at university or, more importantly, what career path they want to take. During secondary education and high school, students walk safely, they have been making decisions, without much impact, such as which branch to choose (if sciences or literature), which electives do you like more, etc. In addition, in these decisions, if there was any doubt, you were lucky to have professors who could explain to you what the courses of their branch were going to be like. However, when you finish high school, you don't really know where to continue. On the one hand, some lucky people know what to study, since it has always been clear to them, but for most of the students it is a decision with many unknowns and few answers. Also, some other people know someone who has studied the same thing they want to study, but these are isolated cases. On the other hand, there are too many degrees and double degrees in different universities and the only information available is the list of courses they have on their respective websites.

Searchity was born to face this problem, a social network that connects pre-university students, young people who are looking for a career, with university students so that they can solve all their doubts about choosing a university degree. The application has two different but complementary functionalities to fulfill this mission. The first and main one is that it allows pre-university students to establish a chat with a university student of the degree and university in which they are interested. In this chat, the pre-university student will be able to have a conversation with the university student, who will answer all her questions. The second is a forum divided into universities and degrees in which you can see the opinions of other users about the courses, teachers, Erasmus programmes and internships, among others. In this way, Searchity offers a very useful source of information not available anywhere else to help pre-college students make their first big decision.

Keywords: Mobile app, Android, Java, Firebase, University, College Career, Degree, Subjects, Teachers, Internships, Erasmus

Índice de contenidos

1. Introducción	11
1.1 Motivación	11
1.2 Objetivos.....	13
1.3 Impacto esperado	13
1.4 Metodología.....	14
1.5 Estructura	15
2. Estado del arte.....	17
2.1 Contexto tecnológico	17
2.2 Proyectos similares.....	17
2.2.1 Unigow.....	18
2.2.2 Uniscopio	19
2.2.3 Yaq	20
2.2.4 Elegircarrera	20
2.3 Crítica al estado del arte	21
2.4 Propuesta.....	22
3. Análisis del problema	24
3.1 Requisitos funcionales.....	24
3.2 Diagramas de los casos de uso.....	25
3.3 Especificación de los casos de uso	27
3.4 Diagrama de clases.....	34
4. Diseño de la solución	35
4.1 Esquema de componentes	35
4.2 Arquitectura de software	36
4.2.1 Importancia de la arquitectura de software	36
4.2.2 Patrón MVVM.....	37
4.3 Diseño de las interfaces.....	39
4.3.1 Diagrama de flujo	39



4.3.2	Prototipos de las interfaces	40
4.4	Diseño de la base de datos	48
5.	Tecnología utilizada	51
5.1	Android	51
5.2	Android Studio	52
5.3	Java	52
5.4	Firebase	53
5.5	GitHub.....	54
5.6	Lucidchart	54
5.7	Moqups.....	55
6.	Implementación	56
6.1	Presentación	56
6.2	Inicio de sesión.....	57
6.3	Registro.....	58
6.4	Menú	59
6.4.1	Home	60
6.4.2	Chats	63
6.4.3	Perfil.....	66
6.4.4	Ajustes	67
7.	Pruebas.....	68
7.1	Pruebas de usuario.....	68
7.2	Pruebas de calidad	71
7.2.1	Experiencia visual	72
7.2.2	Funcionalidad	73
7.2.3	Rendimiento y estabilidad	73
7.2.4	Privacidad y seguridad	74
8.	Conclusiones.....	75
8.1	Relación del trabajo desarrollado con los estudios cursados	76

9. Trabajos futuros	77
10. Referencias	79
11. Anexo	82

Índice de figuras

Figura 1 – Captura de pantalla de Unigow	18
Figura 2 – Captura de pantalla de Uniscopio.....	19
Figura 3 – Captura de pantalla de yaq	20
Figura 4 – Captura de pantalla del test de orientación de Elegircarrera	21
Figura 5 – Diagrama de caso de uso de un usuario sin autenticar	26
Figura 6 –Diagrama de caso de uso de un usuario autenticado.....	26
Figura 7 – Diagrama de clases de Searchcity	34
Figura 8 – Esquema de los componentes de Searchcity.....	35
Figura 9 – Esquema del patrón MVVM	38
Figura 10 – Diagrama de flujo de las interfaces de usuario	39
Figura 11 – Prototipo de registrarse 1	40
Figura 12 – Prototipo de iniciar sesión.....	40
Figura 13 – Prototipo de registrarse 2	41
Figura 14 – Prototipo de menú.....	41
Figura 15 – Prototipo de home.....	42
Figura 16 – Prototipo de lista de grados	42
Figura 17 – Prototipo de preguntas frecuentes.....	43
Figura 18 – Prototipo de foros.....	43
Figura 19 – Prototipo de crear opinión	44
Figura 20 – Prototipo de mostrar opiniones	44
Figura 21 – Prototipo de perfil.....	45
Figura 22 – Prototipo de ajustes.....	45
Figura 23 – Prototipo de chat	46
Figura 24 – Prototipo de mostrar chats	46
Figura 25 – Menú flotante de mostrar chats.....	46
Figura 26 – Prototipo de buscar universitario	47
Figura 27 – Prototipo de puntuar ayuda	47
Figura 28 – Colecciones de datos que utiliza Searchcity	48
Figura 29 – Colección de blocked	48
Figura 30 – Colección de chats	49
Figura 31 – Colección de completed_helps.....	49
Figura 32 – Colección de messages.....	49
Figura 33 – Colección de users	50
Figura 34 – Colección de universities.....	50
Figura 35 – Colección de colleges	50
Figura 36 – Logo de Android.....	51
Figura 37 – Cuota de mercado de los sistemas operativos en España en marzo de 2022.....	51
Figura 38 – Logo de Android Studio.....	52
Figura 39 – Logo de Java	52
Figura 40 – Logo de Firebase	53
Figura 41 – Logo de GitHub	54

Figura 42 – Logo de Lucidchart	54
Figura 43 – Logo de Moqups	55
Figura 44 – Interfaz de presentación	56
Figura 45 – Interfaz de inicio de sesión	57
Figura 46 – Interfaz de registrarse 2	58
Figura 47 – Interfaz de registrarse 1	58
Figura 48 – Interfaz de menú.....	59
Figura 49 – Interfaz de mostrar grados	60
Figura 50 – Interfaz de home	60
Figura 51 – Interfaz de preguntas frecuentes	61
Figura 52 – Interfaz de foros	61
Figura 53 – Interfaz de crear opinión.....	62
Figura 54 – Interfaz de mostrar opiniones	62
Figura 55 – Interfaz de chat.....	63
Figura 56 – Interfaz de chats	63
Figura 57 – Interfaz de puntuar ayuda.....	64
Figura 58 – Interfaz de menú flotante	64
Figura 59 – Interfaz de crear chat.....	65
Figura 60 – Interfaz de perfil	66
Figura 61 – Interfaz de ajustes.....	67
Figura 62 – Primera pregunta de las pruebas de usuario	68
Figura 63 – Segunda pregunta de las pruebas de usuario	69
Figura 64 – Tercera pregunta de las pruebas de usuario	69
Figura 65 – Cuarta pregunta de las pruebas de usuario	70
Figura 66 – Quinta pregunta de las pruebas de usuario	70
Figura 67 – Sexta pregunta de las pruebas de usuario	71



Índice de tablas

Tabla 1 – Planificación.....	14
Tabla 2 – Comparación de características.....	21
Tabla 3 – Caso de uso iniciar sesión.....	27
Tabla 4 – Caso de uso registrarse	27
Tabla 5 – Caso de uso cambiar contraseña	28
Tabla 6 – Caso de uso cerrar sesión.....	28
Tabla 7 – Caso de uso borrar cuenta	29
Tabla 8 – Caso de uso ver perfil.....	29
Tabla 9 – Caso de uso cambiar foto de perfil	29
Tabla 10 – Caso de uso mostrar chats	30
Tabla 11 – Caso de uso buscar universitario	30
Tabla 12 – Caso de uso enviar mensaje	31
Tabla 13 – Caso de uso bloquear usuario.....	31
Tabla 14 – Caso de uso puntuar ayuda	31
Tabla 15 – Caso de uso mostrar universidades	32
Tabla 16 – Caso de uso mostrar grados	32
Tabla 17 – Caso de uso mostrar preguntas frecuentes.....	33
Tabla 18 – Caso de uso mostrar opiniones.....	33
Tabla 19 – Caso de uso escribir opinión.....	33
Tabla 20 – Comparación de las características de Cloud Firestore y Realtime Database	53
Tabla 21 – Pruebas de la experiencia visual de Google	72
Tabla 22 – Pruebas de la funcionalidad de Google.....	73
Tabla 23 – Pruebas del rendimiento y la estabilidad de Google	74
Tabla 24 – Pruebas de la privacidad y seguridad de Google	74

1. Introducción

Uno de los principales problemas a los que se enfrentan las universidades es la tasa de abandono de sus estudiantes, reflejo de un clima de insatisfacción con el mundo académico. La fundación BBVA condujo un metaanálisis para dar una explicación sobre las causas de porqué un 33% de los alumnos españoles deja sin finalizar el grado en el que se matriculó, un 21% para abandonar la universidad sin obtener un título y el 12% restante para cambiar de estudios. La principal causa que hallaron fue la falta de interés en la carrera elegida debido principalmente a la discordancia entre la idea preconcebida que tenían los estudiantes acerca de cómo sería el grado y la realidad. Por ello, como primera iniciativa recomiendan “mejorar la orientación de los estudiantes a la hora de elegir grado, aprovechando la información disponible” (BBVA, 2019).

Por un lado, en los estudiantes y futuros estudiantes de hoy en día existe una necesidad de acceder a una mejor información sobre el mundo académico y laboral, así como poder expresar sus opiniones al respecto e incluso recibir asesoramiento para decidir mejor. Por otro lado, los estudiantes, al ser jóvenes de entre 18 y 25 años, forman parte de una generación que interactúan con la tecnología sin esfuerzo. Están acostumbrados a buscar información y a hablar con otras personas usando sus dispositivos móviles (Dorsey, 2021).

Dado este contexto, nace la idea de Searchity. Una aplicación móvil que cubre estas necesidades de los estudiantes para ayudarlos a tomar una mejor decisión sobre la carrera universitaria que quieren estudiar, y de esta manera, aumentar su satisfacción en el mundo académico y reducir la tasa de abandono en las universidades.

1.1 Motivación

La aplicación de Searchity surge por dos motivos principales. Uno personal y otro profesional.

Por un lado, nunca tuve claro qué carrera quería estudiar. No he sido una de esas personas que lo ha tenido siempre claro. De hecho, tomé la decisión de estudiar el

doble grado de informática y ADE una semana antes de que finalizara el periodo de inscripción.

Durante mi etapa en el colegio, escogí la rama de ciencias porque me gustaban bastante las matemáticas y la biología por lo que cuando llegué a bachillerato lo único que tenía claro era que iba a estudiar una carrera de ciencias. A lo largo de segundo de bachillerato me tomé en serio la elección de la carrera por lo que empecé a hacer lo único que se puede hacer, ir mirando en los sitios webs de las distintas universidades qué grados había y sus respectivas asignaturas. Me acuerdo de que cuando leía el nombre de las asignaturas me sentía un poco perdido, por ejemplo, cuando leía la asignatura de GCA, gestión y configuración de la arquitectura de los sistemas de información, no sabía muy bien qué era exactamente y qué me iba a encontrar. Solo contaba con un nombre y una breve descripción. Por ello, decidí solicitar una tutoría con el tutor que ofrecía el colegio. El tutor me preguntó cuáles eran las carreras que más me gustaban, le contesté, y su respuesta fue imprimir el listado de sus asignaturas con la nota de corte, escrita por él, en la parte de arriba de las hojas. Al terminar la tutoría, no me podía creer que la ayuda que te ofrecía el centro era algo que podías hacer perfectamente en tu casa en 5 minutos por lo que no volví a solicitar una tutoría y empecé a buscar otras formas que me ayudaran con la decisión. Pregunté a familiares y a amigos si conocían a alguien que hubiera estudiado algo parecido, pero no hubo suerte.

Al final, decidí el doble grado de informática y ADE porque la carrera de informática era la que más me llamaba la atención y porque así, tendría 2 títulos universitarios estudiando solo un año más. Cuando empezaron las clases de la universidad tuve mucha suerte porque las asignaturas me gustaban, sobre todo las de programación, pero podría haberse dado el caso contrario, ya que la decisión final fue más una corazonada que una decisión formada y evaluada debido a la poca información de la que disponía. Sin embargo, muchos amigos y compañeros no acertaron con su corazonada, por el mismo problema de falta de información, por lo que algunos cambiaron de carrera y otros se quedaron porque ya estaban en segundo de carrera y preferían terminarla a empezar otra de nuevo y volver a equivocarse. Ante este contexto, me di cuenta de que existía un problema y, por tanto, una oportunidad.

Por otro lado, me gusta mucho programar. Para mí es como pintar, tienes un lienzo en blanco y puedes crear algo bonito y único, crear algo de la nada. Por este motivo, siempre he estado interesado en hacer una aplicación móvil. Además, programar una aplicación desde cero es un antes y un después en la carrera profesional de un programador ya que se aprenden nuevos conocimientos, se obtiene experiencia y se desarrollan competencias transversales como la capacidad de resolver problemas de manera independiente, habilidades que se tienen muy en cuenta en las empresas.

Por estas razones, comencé este proyecto, pensando en darle solución a un problema que existe en la toma de decisión de una carrera universitaria y en mi futuro laboral como programador.

1.2 Objetivos

El objetivo principal de este proyecto es el desarrollo de una aplicación móvil para Android que proporcione ayuda a los preuniversitarios en la decisión de escoger una carrera universitaria. Con esto en mente, se han establecido los siguientes objetivos:

- Facilitar a los usuarios el registro en la aplicación para poder mantener su información privada.
- Desarrollar un chat que permita establecer conversaciones entre los preuniversitarios y los universitarios en tiempo real.
- Integrar un algoritmo de búsqueda que encuentre al universitario que más se adecue a las necesidades del preuniversitario.
- Disponer de un conjunto de foros donde los usuarios puedan publicar opiniones.
- Permitir a los usuarios valorar las publicaciones.
- Establecer una conexión segura con una base de datos en la nube.
- Disponer de una interfaz gráfica de usuario adaptada a las características del dispositivo y que su uso y navegación sea fácil y atractiva para los usuarios.
- Publicar Searchity en la tienda de aplicaciones para facilitar su descarga e instalación por parte de los usuarios

1.3 Impacto esperado

En este apartado se identificará a los usuarios y se explicarán las mejoras que proporciona Searchity.

En Searchity existen dos tipos de usuarios. Los usuarios que usan la aplicación para encontrar algún tipo de ayuda y los usuarios que se la proporcionan. El primer grupo de usuarios son los preuniversitarios y en el segundo grupo se encuentran los universitarios y los graduados, personas que acabaron la carrera hace menos de 10 años. Searchity divide en dos a este segundo grupo de usuarios para ofrecer una ayuda más completa, de esta manera, los preuniversitarios podrán tener una visión de la carrera desde dos puntos de vista distintos y complementarios: una visión más centrada en la carrera en sí y la otra más centrada en las salidas profesionales que esta ofrece.

Por un lado, los preuniversitarios tendrán la posibilidad de obtener información y aclarar sus dudas desde el aspecto académico y laboral de las carreras que más les interesen. Por ello, se espera que la calidad de sus decisiones mejore, aumentando la satisfacción de los universitarios y reduciendo la tasa de abandono de las universidades.

Por otro lado, los universitarios y los graduados tienen la oportunidad de ofrecer la ayuda que les hubiera gustado recibir en su momento por lo que se espera que aumente su estado de ánimo al ayudar a los demás.

En definitiva, este proyecto pretende apoyar y guiar a los estudiantes para que puedan tener una experiencia universitaria de mayor calidad. Este fin está relacionado sobre todo con el objetivo 4, garantizar una educación de calidad, de la ODS (Moran, 2021).

1.4 Metodología

Para realizar el proyecto es necesario elegir una metodología que se ajuste a sus características. En este caso, se ha seguido una metodología ágil, aquellas que facilitan la corrección de los problemas que fueran surgiendo, consiguiendo una mayor flexibilidad e inmediatez en el desarrollo del proyecto. En la Tabla 1 se pueden ver todos los sprints que ha tenido el proyecto en la que el desarrollo de una funcionalidad incluye el *frontend*, la interfaz gráfica de usuario, y el *backend*, la parte lógica de la aplicación.

Tabla 1 – Planificación

Esprint	Tareas	Fecha de Inicio	Fecha de Fin
Esprint 0	<ul style="list-style-type: none"> Definición de requisitos 	07/06/2021	13/06/2021
Esprint 1	<ul style="list-style-type: none"> Plan de diseño 	14/06/2021	20/06/2021
Esprint 2	<ul style="list-style-type: none"> Desarrollo de iniciar sesión y de registrarse Establecer conexión con la base de datos de Firebase 	21/06/2021	27/06/2021
Esprint 3	<ul style="list-style-type: none"> Desarrollo del <i>navigation view</i> 	28/07/2021	04/07/2021
Esprint 4	<ul style="list-style-type: none"> Desarrollo del perfil del usuario 	05/07/2021	11/07/2021
Esprint 5	<ul style="list-style-type: none"> Desarrollo del listado de chats 	12/07/2021	18/07/2021
Esprint 6	<ul style="list-style-type: none"> Desarrollo de mandar y recibir mensajes 	19/07/2021	25/07/2021
Esprint 7	<ul style="list-style-type: none"> Desarrollo del algoritmo de búsqueda de universitarios Desarrollo de la calificación de la ayuda del universitario al preuniversitario 	26/07/2021	01/08/2021

	<ul style="list-style-type: none"> • Desarrollo de bloquear usuario 		
Esprint 8	<ul style="list-style-type: none"> • Desarrollo del listado de universidades • Desarrollo del listado de grados 	02/08/2021	08/08/2021
Esprint 9	<ul style="list-style-type: none"> • Desarrollo de los foros 	09/08/2021	15/08/2021
Esprint 10	<ul style="list-style-type: none"> • Desarrollo de cerrar sesión 	16/08/2021	22/08/2021
Esprint 11	<ul style="list-style-type: none"> • Revisión 	23/08/2021	29/08/2021
Esprint 12	<ul style="list-style-type: none"> • Pruebas con usuarios 	30/08/2021	20/09/2021
Esprint 13	<ul style="list-style-type: none"> • Mantenimiento y mejora del software 	21/09/2021	Actualidad

1.5 Estructura

Este documento está organizado en ocho capítulos, sin contar el primero ya que es introductorio, que serán brevemente explicados a continuación con el objetivo de que el lector sepa qué se va a encontrar a lo largo del mismo.

En primer lugar, encontramos el *Estado del arte* donde se explica el contexto actual de las nuevas tecnologías y cómo han afectado a la sociedad. Además, se exponen las aplicaciones que tienen relación con Searchity y las desventajas que presentan. Finalmente, se desarrollan las ventajas y diferencias que tiene Searchity respecto a las aplicaciones anteriores.

En segundo lugar, el *Análisis del problema* trata las necesidades de Searchity. Este capítulo está formado por los requisitos funcionales que tiene la aplicación que primero se han representado mediante diagramas de casos de uso y, posteriormente, se han detallado mediante las especificaciones de casos de uso. También, se muestra el diagrama de clases de la aplicación.

En tercer lugar, el *Diseño de la solución* muestra cómo se llevan a cabo los aspectos del capítulo anterior. Para ello, se expone el esquema de los componentes que forman la aplicación, el patrón de diseño que presenta la arquitectura, los prototipos de las interfaces gráficas de usuario y el diseño de la base de datos.

En cuarto lugar, se muestra el capítulo de *Tecnología utilizada*. En él, aparecen las tecnologías que se han utilizado para desarrollar la aplicación y los motivos por lo que se han empleado.

Searchcity: una aplicación móvil para la ayuda en la toma de decisión de escoger una carrera universitaria

En quinto lugar, se trata la *Implementación*. En este capítulo, se muestran las interfaces gráficas de usuario de la aplicación ya terminada.

En sexto lugar, en las *Pruebas* se detallan los testeos que se han realizado para verificar que la aplicación funciona correctamente. En concreto, las pruebas de usuario y las pruebas de calidad.

En séptimo lugar, en el capítulo de *Conclusiones* se expone las dificultades que han aparecido desarrollando la aplicación, los errores que se han cometido, los conocimientos y tecnologías que se han aprendido y la relación que tiene el documento con algunas asignaturas que se han cursado en la carrera.

En octavo lugar, se tratan los *Proyectos futuros*. Un capítulo donde se detallan las mejoras y ampliaciones que se pretenden realizar en un futuro.

Por último, en el *Anexo* se explica la relación de este documento con los objetivos de desarrollo sostenibles.

2. Estado del arte

2.1 Contexto tecnológico

Hoy en día, el progreso tecnológico ha cambiado drásticamente la forma de ver el mundo y de relacionarse con él. Ha modificado las relaciones entre las personas, la búsqueda y transmisión de la información, la comunicación, el trabajo, la educación y el ocio.

Los jóvenes han sido los que más se han visto influenciados por esta revolución tecnológica y los que mejor se han adaptado. Reciben el nombre de nativos digitales ya que han crecido y desarrollado junto a ella.

Unicef España realizó un informe que recoge más de 50.000 opiniones de jóvenes de toda España con el objetivo de conocer el impacto de los avances tecnológicos en la juventud. Se analizó la forma en la que se conectan y utilizan Internet y las redes sociales. El informe reveló que el 94,8% tiene un móvil con conexión a Internet, el 90,8% se conecta a Internet todos los días y el 49,6% se conecta más de 5 horas al día en fines de semana. (Cuesta, 2021)

Ante estos resultados, se puede afirmar que las nuevas tecnologías no solo han influenciado a los jóvenes, sino que forman parte de su estilo de vida. Para ellos, conectarse a Internet se ha convertido en la fuente de diversión y pasar el tiempo, de hacer y hablar con los amigos, de buscar información y de aprender.

2.2 Proyectos similares

En este apartado se documentará una serie de aplicaciones que buscan el mismo objetivo que Searchity, ayudar a los preuniversitarios a escoger una carrera universitaria en España, por lo que se convierten en sus principales rivales.

2.2.1 Unigow

Unigow¹ ha sido reconocida en 2021 como una de las 20 startups más esperanzadoras de Europa y cuenta con una comunidad de 13.100 usuarios. En un año y medio han ayudado a más de 8.000 estudiantes a encontrar carrera.

Para recibir orientación, los estudiantes deben registrarse en el sitio web. Cuando inician sesión, los preuniversitarios pueden seleccionar el universitario que mejor se adapte a sus necesidades y podrán ver su horario para agendar una videollamada y así resolver todas sus dudas. Para poder reservar una videollamada es necesario estar suscrito. Unigow ofrece 3 tipos de tarifa:

- Suscripción Lite por 18€ al año. Cuenta con una videollamada gratis y el resto de las videollamadas por 18€.
- Suscripción Bonus por 38€ al año. Cuenta con dos videollamadas gratis y el resto de las videollamadas por 17€.
- Suscripción Unigow por 42€ al año. Cuenta con tres videollamadas gratis y el resto de las videollamadas por 14€.

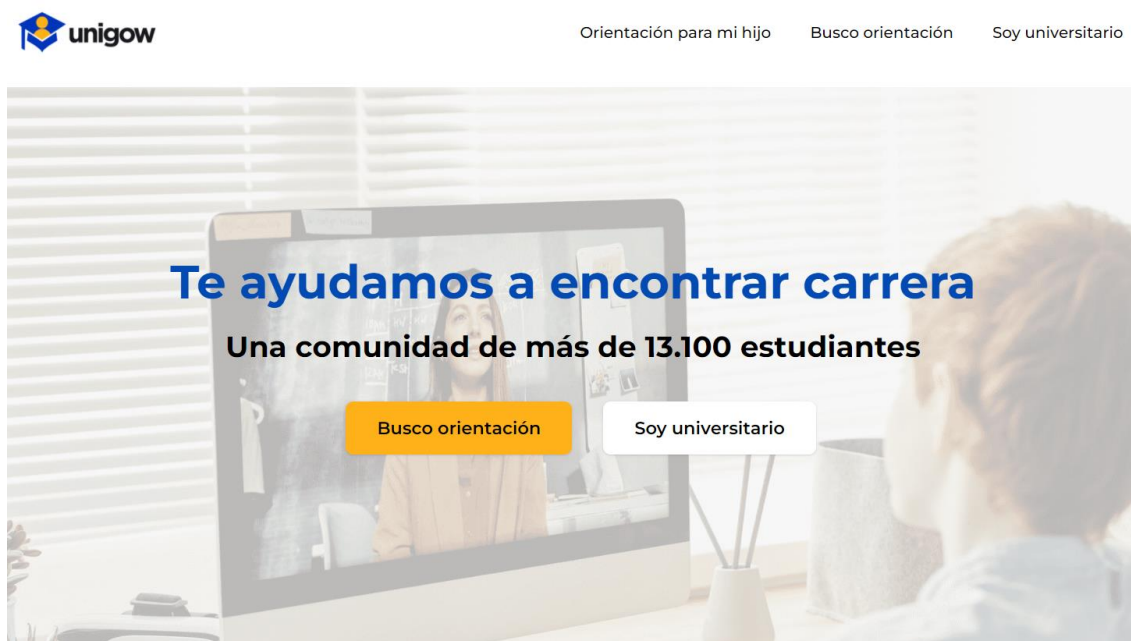


Figura 1 – Captura de pantalla de Unigow

¹ <https://unigow.com/>

2.2.2 Uniscopio

Uniscopio² es una plataforma gratuita que surgió hace aproximadamente 2 años para ofrecer a los estudiantes una ayuda en línea. Sus principales servicios son los siguientes:

- Un buscador de grados de todas las universidades de España. Cuando se selecciona un grado, aparece una breve descripción sobre los objetivos que busca conseguir, el perfil de los ingresados, la distribución de créditos de las asignaturas y las salidas profesionales. También muestra el tipo de modalidad (presencial o en línea), la duración de la carrera, los idiomas en los que puede ser impartida y el mes de inicio del curso.
- Un servicio de orientación mediante correo electrónico. Si el usuario necesita más información o tiene alguna duda puede escribir un correo que será respondido en un par de días.
- Una sección de búsqueda de alojamiento para los universitarios. Permite encontrar residencias universitarias y colegios mayores, así como elegir entre habitación individual o compartida, pensión completa o media pensión, servicios extra como limpieza o lavandería.

Grado en Ingeniería Informática – Universidad Carlos III de Madrid

The screenshot displays a contact form on the left and course details on the right. The form includes fields for 'Nombre Completo', 'Email', and 'Teléfono', a checkbox for 'Acepto recibir noticias de acuerdo al marco legal Uniscopio', and a 'Solicitar información' button. The course details section features four icons: a book for 'Presencial', a green board with the number '4' for '4 años', flags for 'Español Inglés', and a calendar for 'Septiembre'. Below these is a 'Descripción' section with text about the degree's objectives and curriculum.

Figura 2 – Captura de pantalla de Uniscopio

² <https://uniscopio.com/>

2.2.3 Yaq

Yaq³ es un sitio web que contiene los siguientes apartados para ayudar a los estudiantes a elegir una carrera:

- Un listado de los grados universitarios españoles. Cuando se selecciona un grado muestra las universidades donde se imparte esta titulación, el tipo de universidad (pública o privada), la provincia, la modalidad y la nota de corte.
- Las 10 carreras universitarias más buscadas por rama.
- Una colección de artículos como: 5 mitos de elegir una carrera universitaria, los 7 errores más frecuentes a la hora de elegir carrera, consejos para elegir una carrera universitaria, entre otros.



Figura 3 – Captura de pantalla de yaq

2.2.4 Elegircarrera

Elegircarrera⁴ es otro sitio web que tiene dos funcionalidades para guiar a los estudiantes.

La primera es un listado de consejos y puntos para tener en cuenta antes de que el estudiante elija una titulación como: realizar un listado de sus fortalezas y debilidades, hacer una lista de sus gustos y aficiones, elegir un grado con bastantes salidas laborales, no dejarse llevar por modas o tendencias, buscar la máxima información sobre la carrera, hablar con un orientador, entre otros.

La segunda es un test de orientación vocacional general que cuenta con una serie de preguntas que, una vez respondido, te indica mediante un porcentaje las áreas que mejor se ajustan al perfil como se puede ver en Figura 4. Una vez se ha realizado este primer test general se puede hacer un segundo, más específico, que indica qué tres carreras de la rama escogida se adaptan mejor al estudiante.

³ <https://yaq.es/>

⁴ <https://www.elegircarrera.net/>

Áreas destacadas

Estas son las 3 áreas donde has obtenido una mayor puntuación:

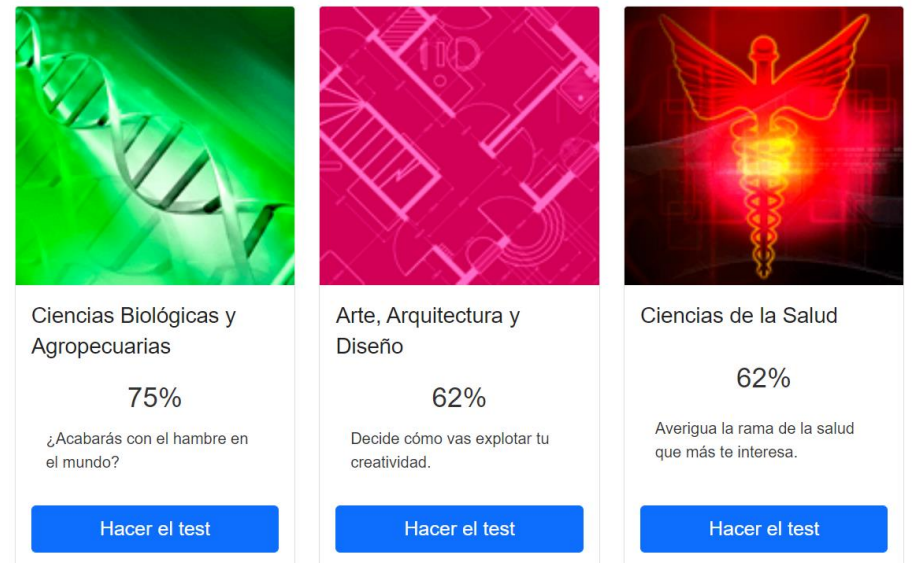


Figura 4 – Captura de pantalla del test de orientación de Elegircarrera

2.3 Crítica al estado del arte

En este apartado se analizan los proyectos presentados anteriormente para identificar las desventajas que presentan. Para ello, se compara en la Tabla 2 las características de cada uno de ellos y a continuación se explica cada punto en profundidad.

Tabla 2 – Comparación de características

Características	Unigow	Uniscopio	Yaq	Elegircarrera
Sitio web	✓	✓	✓	✓
Gratis	✗	✓	✓	✓
Información de la carrera	✗	✓	✓	✓
Contactar con universitarios	✓	✗	✗	✗
Existencia de foros	✗	✗	✗	✗
Uso diario	✗	✓	✓	✗

En primer lugar, las cuatro aplicaciones son sitios web. Para acceder a un sitio web es necesario el uso de un navegador desde un ordenador o desde un dispositivo móvil. En el caso del ordenador, su uso es un poco limitado ya que solo se puede acceder desde el lugar donde se encuentra el ordenador. Sin embargo, en el caso del dispositivo móvil, si se puede acceder al sitio web desde cualquier ubicación, pero el uso del navegador es poco cómodo ya que la pestaña del navegador ocupa parte de la pantalla dejando menos espacio a la interfaz del sitio web.

En segundo lugar, a diferencia del resto de aplicaciones, Unigow no es gratuita. La mayoría de los estudiantes, sobre todo los preuniversitarios, no trabajan por lo que no disponen de un respaldo económico. Ante este contexto, las tarifas de Unigrow son un poco elevadas. Por ejemplo, si un preuniversitario tiene dudas acerca de 3 grados debe de gastar como mínimo 42€ ya que si quisiera comparar los grados con los de otra universidad le costaría el doble.

En tercer lugar, todas las aplicaciones buscan el mismo objetivo, ayudar a los preuniversitarios a encontrar una carrera universitaria, si bien, no lo hacen de la misma manera. Por un lado, Unigow permite a los preuniversitarios contactar con los universitarios mediante videollamadas para que puedan hacer preguntas y aclarar sus dudas. Por otro lado, Uniscopio, Yaq y Elegircarrera se centran en ofrecer información que ayude a guiar a los estudiantes como artículos, tests, consejos, características de cada grado. Por tanto, no existe una aplicación donde se pueda encontrar información relacionada con la carrera y a su vez contactar con universitarios.

Por último, Unigow y Elegircarrera son sitios web que no se pueden usar a diario. En el caso de Unigow, un estudiante promedio realizará una videollamada o dos. Si se le presenta una nueva duda no pagará una nueva videollamada ya que no le compensa. En cuanto a Elegircarrera, una vez se han realizado los tests el sitio web pierde utilidad.

2.4 Propuesta

Después de conocer las funcionalidades y desventajas de algunas aplicaciones que se encuentran en el mercado, es importante explicar cuáles son las mejoras que ofrece Searchcity que hacen que la diferencie del resto, otorgándole una ventaja competitiva, y así se entienda mejor el porqué de este trabajo. Estas diferencias son las siguientes:

Aplicación móvil. Searchcity, a diferencia del resto de proyectos, es una aplicación móvil ya que busca que sea lo más cómodo posible para el usuario. No necesita un navegador para acceder, simplemente se hace clic en la aplicación lo que agiliza su uso. Además, al prescindir del navegador, la interfaz ocupa toda la pantalla del dispositivo

móvil dando como resultado que la aplicación tenga un aspecto más estético y accesible.

Precio. De las aplicaciones con las que se ha comparado Searchity, Unigow es la que más se le parece, convirtiéndose en su principal rival. Sin embargo, entre sus diferencias, la principal es el precio. Searchity es una aplicación gratuita que apuesta por ofrecer una ayuda que sea asequible para todos los estudiantes.

Funcionalidades. Searchity engloba las dos características mencionadas anteriormente: contactar con universitarios y ofrecer información de las carreras. Para conseguirlo, cuenta con 3 funcionalidades:

- Permite a los preuniversitarios crear chats con los universitarios para que puedan establecer conversaciones en tiempo real.
- Ofrece información detallada de cada carrera.
- Cuenta con un foro dividido en universidades, grados y temáticas en el que se pueden ver opiniones de otros usuarios. Esta funcionalidad no la tiene ninguna otra aplicación en el mercado por lo que se convierte en una de las fortalezas de Searchity.

Uso diario. Searchity es una red social por lo que ofrece a sus usuarios la posibilidad de usarla a diario. Los preuniversitarios si tienen alguna duda pueden resolverla mediante el chat directo o leyendo opiniones en los foros.

3. Análisis del problema

En este capítulo se procede a explicar los aspectos más significativos del análisis del problema: los requisitos funcionales, los casos de uso, la especificación de los casos de uso y el diagrama de clases de la base de datos.

3.1 Requisitos funcionales

Los requisitos funcionales son el comportamiento que debe tener una solución software y la información que debe manejar. En este caso, los requisitos se han dividido en funcionalidades de los usuarios que no se han autenticado y funcionalidades de los que sí se han autenticado.

Usuario no autenticado

- **Iniciar sesión:** el usuario se identifica y la aplicación le permite o deniega el acceso a la aplicación.
- **Registrar usuario:** el usuario puede registrarse rellorando un formulario con sus datos personales para que pueda acceder en futuras conexiones.
- **Cambiar contraseña:** permite al usuario establecer una nueva contraseña si el usuario no se acuerda de la antigua.

Usuario autenticado

- **Cerrar sesión:** permite salir de la cuenta del usuario.
- **Borrar cuenta:** elimina la cuenta y los datos del usuario.
- **Ver perfil:** permite ver el perfil del usuario.
- **Cambiar foto del perfil:** permite cambiar la foto de perfil del usuario.
- **Mostrar chats:** muestra el listado de chats del usuario.
- **Buscar universitario:** un preuniversitario selecciona el grado de la universidad que le interesa y Searchity le crea un chat con el universitario que más se adapta a sus necesidades.
- **Enviar mensaje:** permite enviar mensajes a un usuario.

- **Bloquear usuario:** Un usuario puede bloquear a otro. Se elimina el usuario bloqueado de su listado de chats y no podrá establecer de nuevo un chat con él.
- **Puntuar ayuda:** Una vez un universitario ha terminado de resolver las dudas, el preuniversitario valorará su ayuda con una calificación del 1 al 10 y una breve descripción.
- **Mostrar listado de universidades:** permite visualizar las universidades de la Comunidad Valenciana.
- **Mostrar listado de grados:** permite visualizar los grados de una universidad.
- **Mostrar opiniones:** permite visualizar las opiniones de un grado.
- **Mostrar preguntas frecuentes:** permite visualizar las preguntas más realizadas por los usuarios de un grado.
- **Escribir opinión:** un usuario puede dar su opinión sobre algún aspecto de la universidad.

3.2 Diagramas de los casos de uso

Una vez se han definido los requisitos funcionales, podemos representarlos en un diagrama de casos de uso, una forma de representar el conjunto de funcionalidades que tiene la aplicación así como sus relaciones entre sí y con el exterior. Los diagramas de caso de uso están formados principalmente por 4 objetos:

- **Actor.** Es aquella entidad que desempeña un papel en la aplicación. Puede ser tanto una persona como una organización o un sistema externo. Se representa con la silueta de una persona.
- **Caso de uso.** Son las funcionalidades o acciones de la aplicación. Se representa con un óvalo con el nombre de la acción en su interior.
- **Relaciones.** Indican las asociaciones de los casos de uso entre ellos y con los actores. Un tipo de relación particular son las de inclusión que indican que un caso de uso está incluido en otro.
- **Sistema.** Define el alcance de los casos de uso. Se representa con un rectángulo.

Estos diagramas tienen mucha importancia ya que permiten identificar las funciones y la forma en la que los actores interactúan con ellas, tener una visión de alto nivel del sistema e identificar los factores internos y externos.

En la Figura 5 y en la Figura 6 podemos observar el diagrama de los casos de uso de los usuarios sin autenticar y autenticados.

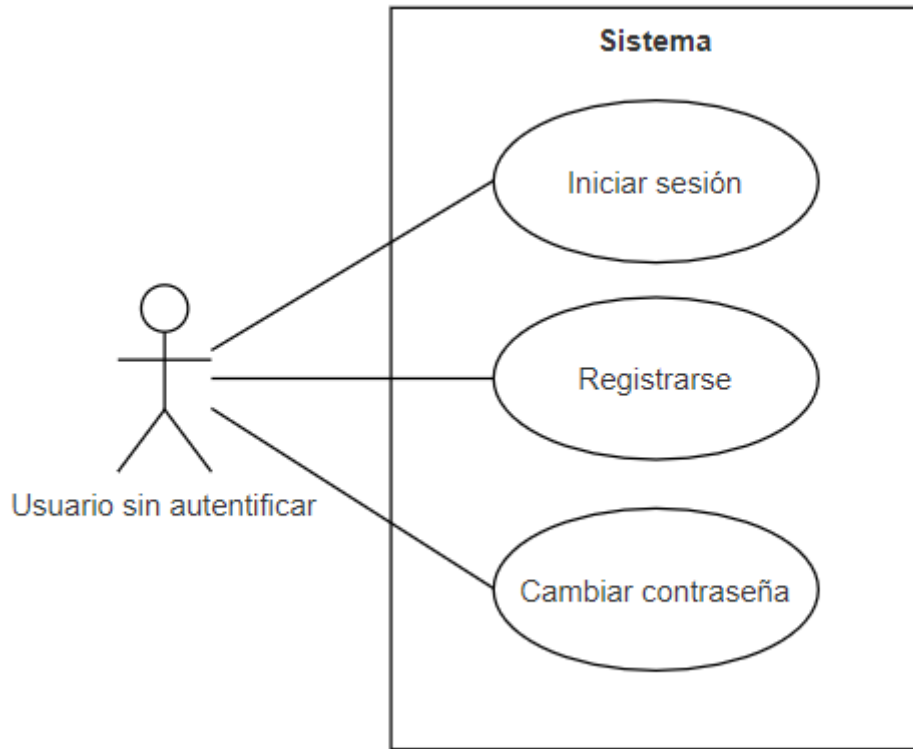


Figura 5 – Diagrama de caso de uso de un usuario sin autenticar

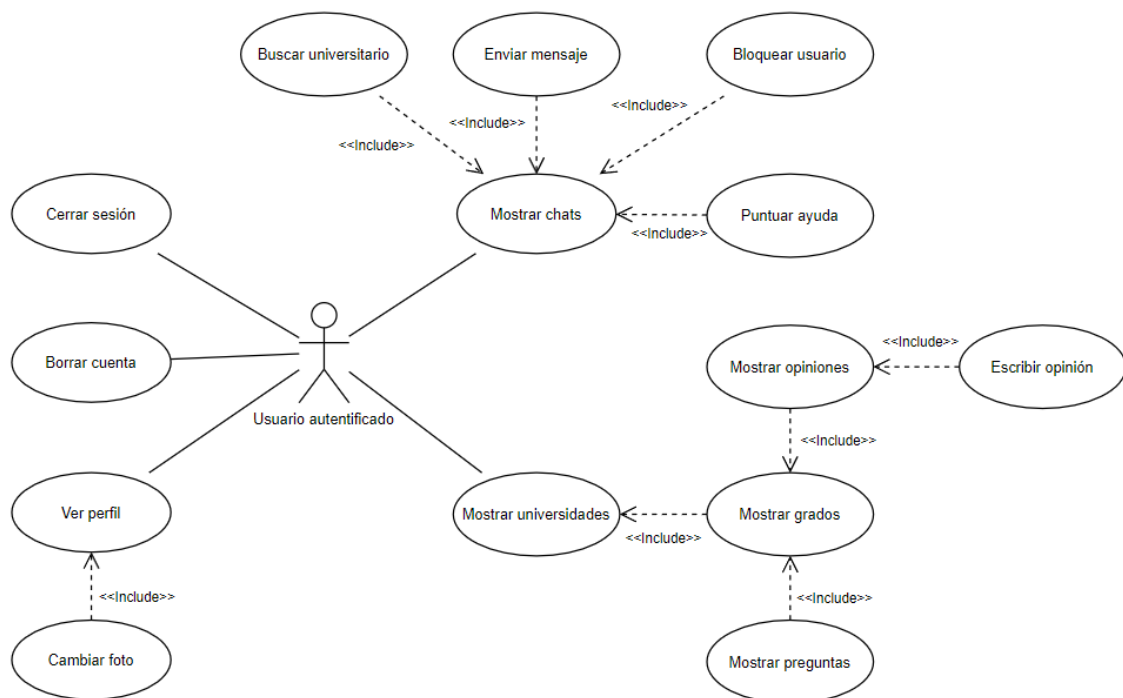


Figura 6 –Diagrama de caso de uso de un usuario autenticado

3.3 Especificación de los casos de uso

Este apartado explicará en detalle los casos de uso de la aplicación mediante tablas que indican una breve descripción del caso, el actor que la realiza, la precondición que se debe cumplir, la secuencia de acciones que ocurren y, por último, la postcondición que es el estado que se encuentra la aplicación después del caso de uso.

Tabla 3 – Caso de uso iniciar sesión

Iniciar Sesión	
Descripción	El usuario se identifica y la aplicación le permite o deniega el acceso a la aplicación.
Actor	Usuario no autenticado.
Precondición	El usuario no tiene una sesión abierta.
Secuencia	<ol style="list-style-type: none">1. El usuario introduce su correo electrónico y contraseña en los campos.2. El usuario pulsa el botón de iniciar sesión.3. El sistema comprueba si el correo electrónico y la contraseña pertenecen a un usuario. Si existe el usuario le deja acceder a la aplicación y, en caso contrario, le deniega el acceso y le muestra un mensaje de error.
Postcondición	El usuario está autenticado.

Tabla 4 – Caso de uso registrarse

Registrarse	
Descripción	El usuario puede registrarse rellenando un formulario con sus datos personales para que pueda acceder en futuras conexiones.
Actor	Usuario no autenticado.
Precondición	El usuario no tiene una sesión abierta.

Secuencia	<ol style="list-style-type: none"> 1. El usuario desplaza la pantalla a la izquierda para acceder a la ventana de registrarse. 2. El usuario selecciona el tipo de usuario que es: preuniversitario, universitario o graduado. 3. El sistema dependiendo del tipo de usuario seleccionado ofrece un tipo de formulario con un listado de campos que el usuario debe rellenar. 4. El usuario introduce sus datos. 5. El sistema comprueba el formato de los campos. Si son válidos los guarda en Firebase y, en caso contrario, muestra un mensaje de error.
Postcondición	El usuario está autenticado.

Tabla 5 – Caso de uso cambiar contraseña

Cambiar contraseña	
Descripción	Permite al usuario establecer una nueva contraseña si el usuario no se acuerda de la antigua.
Actor	Usuario no autenticado.
Precondición	El usuario no tiene una sesión abierta.
Secuencia	<ol style="list-style-type: none"> 1. El usuario selecciona “¿Has olvidado la contraseña?”. 2. El sistema le envía un correo electrónico. 3. El usuario introduce una nueva contraseña.
Postcondición	El sistema guarda la nueva contraseña.

Tabla 6 – Caso de uso cerrar sesión

Cerrar sesión	
Descripción	Permite salir de la cuenta del usuario
Actor	Usuario autenticado.
Precondición	El usuario tiene una sesión abierta.
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa en el botón de cerrar sesión 2. El sistema cierra la sesión del usuario. 3. El sistema muestra la interfaz de iniciar sesión

Postcondición	El usuario ya no tiene una sesión abierta.
----------------------	--

Tabla 7 – Caso de uso borrar cuenta

Borrar cuenta	
Descripción	Elimina la cuenta y los datos del usuario
Actor	Usuario autenticado.
Precondición	El usuario ha accedido a la interfaz de ajustes del usuario.
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa en “Ajustes”. 2. El usuario pulsa en el botón de borrar cuenta 3. El sistema borra los datos del usuario. 4. El sistema muestra la interfaz de iniciar sesión
Postcondición	El usuario ya no tiene una cuenta.

Tabla 8 – Caso de uso ver perfil

Ver perfil	
Descripción	Permite ver el perfil del usuario.
Actor	Usuario autenticado.
Precondición	El usuario tiene una sesión abierta.
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa en “Perfil”. 2. El sistema muestra la interfaz de perfil del usuario
Postcondición	El sistema muestra la información personal del usuario y su foto de perfil.

Tabla 9 – Caso de uso cambiar foto de perfil

Cambiar foto de perfil	
Descripción	Permite cambiar la foto de perfil del usuario
Actor	Usuario autenticado.
Precondición	El usuario ha accedido a la interfaz de perfil del usuario.

Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa en “Cambiar foto” o en su foto de perfil. 2. El sistema accede a la galería del dispositivo del usuario. 3. El usuario selecciona la imagen nueva que quiere para su perfil.
Postcondición	La foto queda guardada en el sistema.

Tabla 10 – Caso de uso mostrar chats

Mostrar chats	
Descripción	Muestra el listado de chats del usuario
Actor	Usuario autenticado.
Precondición	El usuario tiene una sesión abierta.
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa en “Chats”. 2. El sistema muestra la interfaz de chats.
Postcondición	El sistema muestra los chats que tiene el usuario.

Tabla 11 – Caso de uso buscar universitario

Buscar universitario	
Descripción	Un preuniversitario selecciona el grado de la universidad que le interesa y Searchity le crea un chat con el universitario que más se adapta a sus necesidades.
Actor	Usuario autenticado.
Precondición	El usuario se encuentra en la interfaz de chats.
Secuencia	<ol style="list-style-type: none"> 1. El preuniversitario pulsa en el botón con el signo de más que se encuentra abajo a la derecha de la interfaz. 2. El sistema muestra la interfaz de buscar universitario. 3. El preuniversitario indica la provincia, la universidad y el grado y pulsa el botón de aceptar. 4. El sistema crea el chat si encuentra un universitario con esas características disponible. En caso contrario, devuelve un mensaje diciendo que no hay un universitario disponible.
Postcondición	El sistema vuelve a la interfaz de chats.

Tabla 12 – Caso de uso enviar mensaje

Enviar mensaje	
Descripción	Permite enviar mensajes a un usuario.
Actor	Usuario autenticado.
Precondición	El usuario ha abierto un chat.
Secuencia	<ol style="list-style-type: none"> 1. El usuario escribe un nuevo mensaje. 2. El usuario pulsa en el botón de enviar.
Postcondición	El sistema guarda y muestra el nuevo mensaje.

Tabla 13 – Caso de uso bloquear usuario

Bloquear usuario	
Descripción	Un usuario puede bloquear a otro. Se elimina el usuario bloqueado de su listado de chats y no podrá establecer de nuevo un chat con él.
Actor	Usuario autenticado.
Precondición	El usuario se encuentra en la interfaz de chats.
Secuencia	<ol style="list-style-type: none"> 1. El usuario mantiene pulsado en un chat. 2. El sistema muestra una ventana de opciones. 3. El usuario pulsa en “Bloquear usuario”. 4. El sistema muestra una ventana de confirmación. 5. El usuario pulsa en “Aceptar”.
Postcondición	El sistema bloquea al usuario.

Tabla 14 – Caso de uso puntuar ayuda

Puntuar ayuda	
Descripción	Una vez un universitario ha terminado de resolver las dudas, el preuniversitario valorará su ayuda con una calificación del 1 al 10 y una breve descripción.
Actor	Usuario autenticado.

Precondición	El usuario se encuentra en la interfaz de chats.
Secuencia	<ol style="list-style-type: none"> 1. El usuario mantiene pulsado en un chat. 2. El sistema muestra una ventana de opciones. 3. El usuario pulsa en “Puntuar ayuda”. 4. El sistema muestra la interfaz de puntuar ayuda. 5. El usuario califica la ayuda y escribe una breve descripción. 6. El usuario pulsa en “Aceptar”.
Postcondición	El sistema guarda la puntuación.

Tabla 15 – Caso de uso mostrar universidades

Mostrar universidades	
Descripción	Permite visualizar las universidades.
Actor	Usuario autenticado.
Precondición	El usuario tiene una sesión abierta.
Secuencia	No hay secuencia de acciones.
Postcondición	El sistema muestra las universidades disponibles en la aplicación.

Tabla 16 – Caso de uso mostrar grados

Mostrar grados	
Descripción	Permite visualizar los grados de una universidad.
Actor	Usuario autenticado.
Precondición	El usuario ha seleccionado una universidad.
Secuencia	No hay secuencia de acciones.
Postcondición	El sistema muestra los grados disponibles de la universidad seleccionada.

Tabla 17 – Caso de uso mostrar preguntas frecuentes

Mostrar preguntas frecuentes	
Descripción	Permite visualizar las preguntas más realizadas por los usuarios de un grado.
Actor	Usuario autenticado.
Precondición	El usuario ha seleccionado un grado.
Secuencia	El usuario pulsa en “Preguntas Frecuentes”.
Postcondición	El sistema muestra las preguntas frecuentes que se han realizado sobre el grado seleccionado.

Tabla 18 – Caso de uso mostrar opiniones

Mostrar opiniones	
Descripción	Permite visualizar las opiniones de un grado.
Actor	Usuario autenticado.
Precondición	El usuario ha seleccionado un grado.
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa en “Foros”. 2. El usuario selecciona un foro de los 4 disponibles. Estos foros son de asignaturas, profesores, Erasmus y prácticas.
Postcondición	El sistema muestra las opiniones sobre el foro seleccionado.

Tabla 19 – Caso de uso escribir opinión

Escribir opinión	
Descripción	Un usuario puede dar su opinión sobre algún aspecto de la universidad.
Actor	Usuario autenticado.
Precondición	El usuario ha seleccionado un foro.
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa en el botón con un signo de más. 2. El sistema muestra la interfaz de escribir opinión. 3. El usuario escribe una nueva opinión y pulsa en “Aceptar”.
Postcondición	El sistema guarda la nueva opinión.

3.4 Diagrama de clases

El diagrama de clases es un tipo de diagrama que representa las clases de la aplicación (objetos de los que se pretende guardar información), sus relaciones y sus atributos.

En la Figura 7 se puede ver el diagrama de clases de Searchcity el cual tiene 7 clases (usuario, chat, mensaje, universidad, grado, opinión y pregunta) y 7 asociaciones (bloquea, da me gusta, establece, contiene, ofrece, formado por y tiene).

El diagrama se ha realizado en el sitio web Lucidchart, una aplicación de creación de diagramas UML que es gratuita y online.

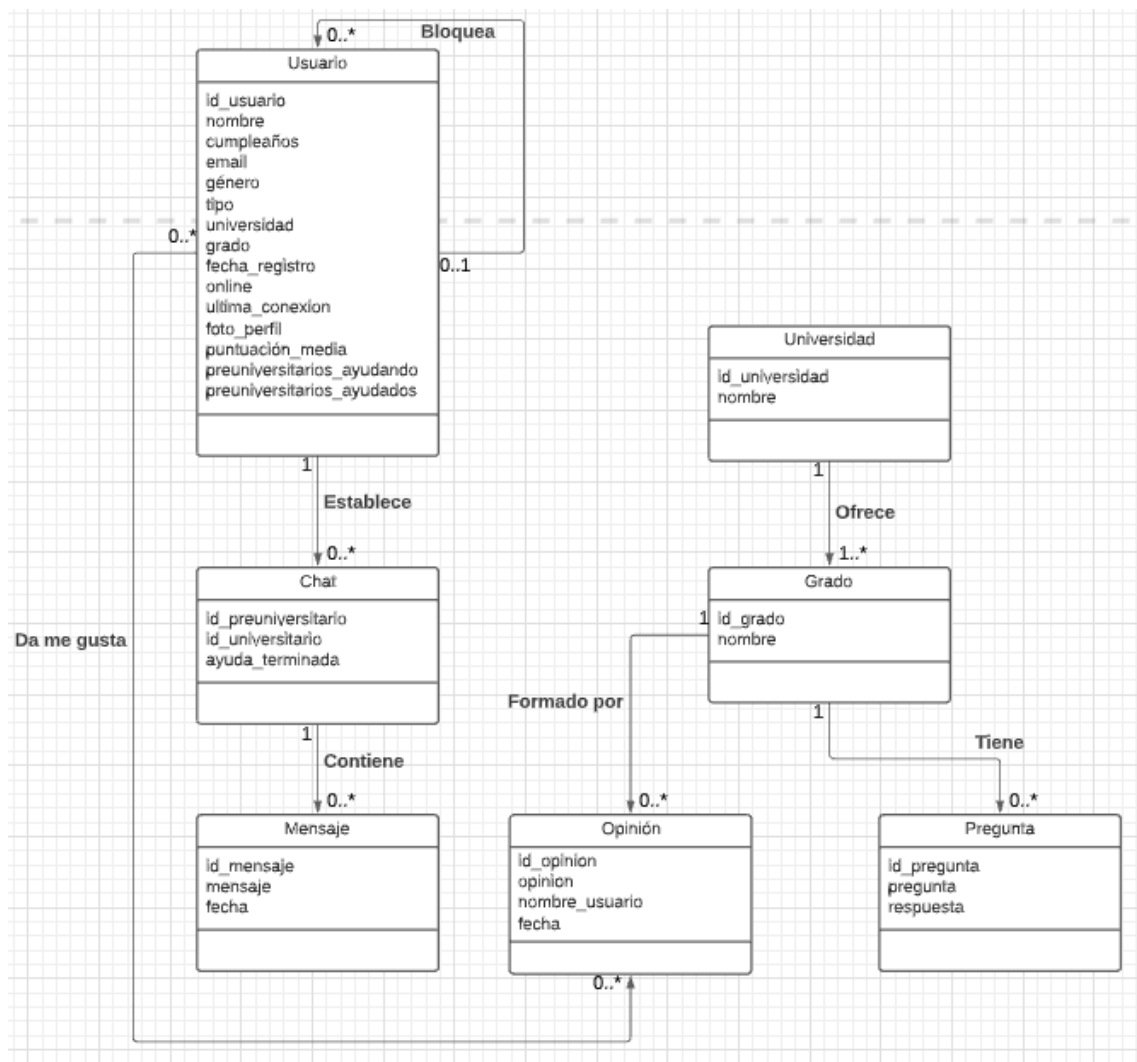


Figura 7 – Diagrama de clases de Searchcity

4. Diseño de la solución

Una vez se han explicado los requisitos de la aplicación, sus casos de usos y el modelo de datos que se va a utilizar (aspectos que indican qué necesita la aplicación), hay que decidir su diseño (aspectos de cómo se llevará a cabo la aplicación). Por ello, en este capítulo se tratará el esquema de los componentes que forman la aplicación, la arquitectura de software que presenta, el diseño de las interfaces gráficas de usuario y el diseño de la base de datos.

4.1 Esquema de componentes

En la Figura 8 se muestran los principales componentes que presenta Searchity:

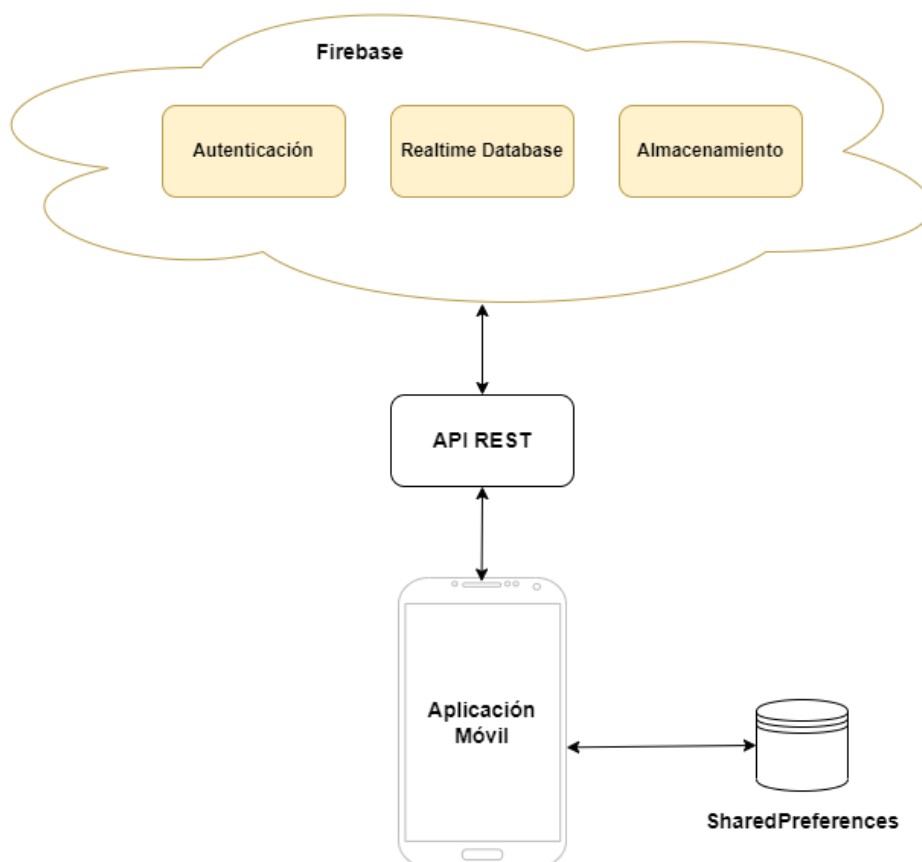


Figura 8 – Esquema de los componentes de Searchity

- **Firestore.** Es una plataforma diseñada para el desarrollo de aplicaciones web y aplicaciones móviles. Proporciona varios servicios, pero los que se usarán son los siguientes:
 - Autenticación: se encarga de gestionar la identificación de los usuarios.
 - Realtime Database: permite la obtención de datos en tiempo real, por lo que las consultas que se estén visualizando en la aplicación se refrescarán automáticamente si cambian los datos.
 - Almacenamiento: permite guardar documentos, imágenes y vídeos. En nuestro caso, se utilizará para almacenar las imágenes de las universidades y las fotos de perfil de los usuarios.
- **API REST.** Se encarga de la comunicación entre Firestore y la aplicación móvil para que todos los servicios funcionen correctamente.
- **Aplicación Móvil.** En este caso, se trata de Searchcity. Contiene el software que permite realizar los casos de uso, mencionados en el capítulo 3.
- **SharedPreferences.** Es un objeto que proporciona Android para guardar una pequeña colección de datos mediante el formato clave-valor. El objeto se almacena en el dispositivo móvil en un fichero XML. En Searchcity se utiliza *SharedPreferences* para guardar datos que se utilizan con mucha frecuencia como el nombre del usuario.

4.2 Arquitectura de software

La arquitectura de software es cómo se organiza un sistema. En esta organización se incluyen todos los componentes que forman parte, cómo interactúan entre sí, el entorno en el que operan y los principios que siguen para el diseño del software.

4.2.1 Importancia de la arquitectura de software

Según Robert C. Martin, conocido como *Uncle Bob*, determina los cinco principios de diseño de software cuyas iniciales forman la sigla SOLID. Estos principios son los siguientes:

- **S - Single-Responsibility Principle.** El principio de responsabilidad única determina que una clase debe tener solo una razón para cambiar, es decir, una clase debe tener solo un trabajo. Las clases no deben de tener varias responsabilidades lógicas a la vez.

- **O - *Open-Closed Principle***. El principio abierto-cerrado establece que los objetos deben estar abiertos para la extensión, pero cerrados para la modificación. Se debe poder extender el comportamiento de las clases sin cambiar su código.
- **L - *Liskov Substitution Principle***. El principio de sustitución de Liskov afirma que cada subclase o clase derivada debe poder sustituir a su clase padre. Si en una parte del código se usa una clase, y posteriormente esta se extiende, se debe poder utilizar la clase hija y que el programa siga funcionando correctamente.
- **I - *Interface Segregation Principle***. El principio de segregación de interfaces asegura que a un cliente no se le debe obligar a usar una interfaz o método que no usa. Por tanto, ninguna clase debe depender de métodos que no utiliza. Esto evita las “fat interfaces” (interfaces con muchos métodos de los cuáles solo se necesitan unos pocos).
- **D - *Dependency Inversion Principle***. El principio de inversión de dependencia define que las entidades deben depender de abstracciones, no de concreciones. Este principio determina que el código no puede depender de los detalles de la implementación como *frameworks*, bases de datos, entre otros.

Aunque aplicar estos principios puede parecer abrumador y costoso, cuando se incorporan en el diseño del software nos ayuda a tener una mayor accesibilidad, facilidad de refactorización, depuración, extensibilidad y legibilidad en el código. Por tanto, se escribe un código más limpio, de mayor calidad y más fácil de leer, ampliar, testear y mantener. (Ridwan, 2021)

4.2.2 Patrón MVVM

Un patrón de diseño es una solución general y reutilizable para problemas comunes que ocurren en el desarrollo de software. Los patrones no son una solución final que se transforma en código, sino que son una guía o plantilla que puedes utilizar en tu proyecto y que permiten la formalización de buenas prácticas, como los principios SOLID.

En el desarrollo de aplicaciones móviles existen una gran variedad de patrones arquitectónicos. Aunque estas arquitecturas varían en unos pocos detalles, es necesario conocerlas para saber cuál es la que mejor se adapta a los requerimientos



del proyecto. Hoy en día, los patrones arquitectónicos en Android más importantes y utilizados son:

- *Model View Controller* (MVC)
- *Model View Presenter* (MVP)
- *Model View View Model* (MVVM)

En Searchcity se escogió el patrón *Model View View Model* ya que es el más popular desde que Google lo convirtió en su arquitectura oficial. Como se puede ver en la Figura 9, este patrón organiza las aplicaciones en tres módulos o partes distintas:

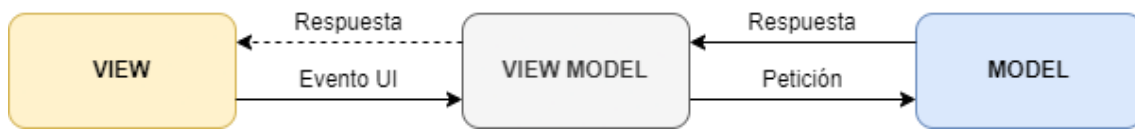


Figura 9 – Esquema del patrón MVVM

- **View.** La vista está formada por las interfaces de usuario (en XML), las actividades y los fragmentos. La vista se caracteriza por no realizar ninguna acción. Las interfaces gráficas de usuario son la apariencia de la aplicación y las actividades y los fragmentos son clases que corresponden con una interfaz de usuario y se encargan de recoger los eventos de éstas para mandárselos al *ViewModel*. Por tanto, en la vista no se encuentra la lógica de negocio de la aplicación.
- **ViewModel.** El *ViewModel* tiene dos funciones: establecer el estado de la vista y ser la conexión entre el modelo y la vista. Las vistas se suscriben a sus respectivos *ViewModels* y cuando detectan un evento se lo comunican. Los *ViewModels* gestionarán el evento o, si es necesario la base de datos, se comunicarán con el modelo. De esta forma, la información siempre fluye en una única dirección: la vista captura el evento, el *ViewModel* reacciona al evento y actualiza su estado y finalmente la vista muestra el nuevo estado al cliente.
- **Model.** El modelo representa la parte de datos. Está formado por las fuentes de datos y los repositorios. Por un lado, las fuentes de datos son clases que se encargan de conectar con la base de datos o con un servicio web y realizar las consultas. Por otro lado, los repositorios son abstracciones que permiten acceder a las fuentes de datos de forma transparente, sin necesidad de conocer los detalles de implementación. Son las clases que comunican los *ViewModels* con las fuentes de datos. (Rodríguez, 2020)

4.3 Diseño de las interfaces

El diseño de la interfaz gráfica de usuario constituye una parte importante en el desarrollo de una aplicación móvil ya que es el medio con el que los usuarios interactúan con ella. Por ello, para tener un buen diseño, se realizó antes de programar las interfaces, un diagrama de flujo para establecer cómo se relacionarán las interfaces entre ellas y se crearon sus respectivos prototipos para tener una idea completa de cómo se verá el producto final.

4.3.1 Diagrama de flujo

En la Figura 10 se puede observar el diagrama de flujo de las interfaces de usuario que seguirá Searchcity. Está compuesto por 14 interfaces distintas y se puede navegar por ellas en el orden que indican las flechas.

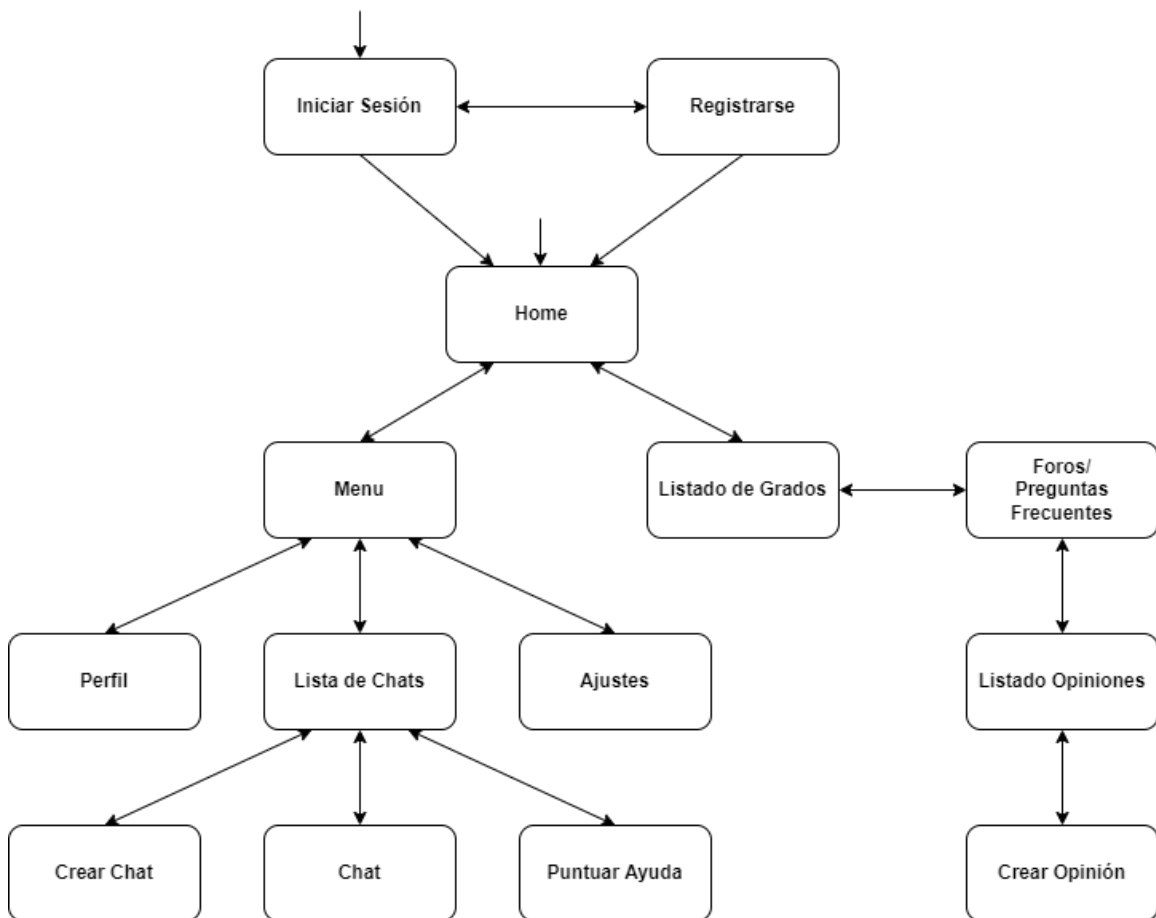


Figura 10 – Diagrama de flujo de las interfaces de usuario

La interfaz de iniciar sesión y de home tienen una flecha que no conecta con ninguna otra interfaz ya que son interfaces a las que se puede acceder directamente cuando se abre la aplicación. El acceso a una u a otra depende de si el usuario tiene una sesión activa o no. Si no tiene una sesión activa, la aplicación le mostrará la interfaz de iniciar sesión, y si ya la tiene, le mostrará la interfaz de home. De esta manera, el usuario no tiene que iniciar sesión cada vez que entre en la aplicación.

4.3.2 Prototipos de las interfaces

En la Figura 12 se puede observar el caso de uso de iniciar sesión y el de cambiar contraseña y en la Figura 11, la primera parte del caso de uso de registrarse. Para acceder a ella, el usuario desplaza la pantalla con el dedo hacia la izquierda o pulsa en la pestaña de registrarse.

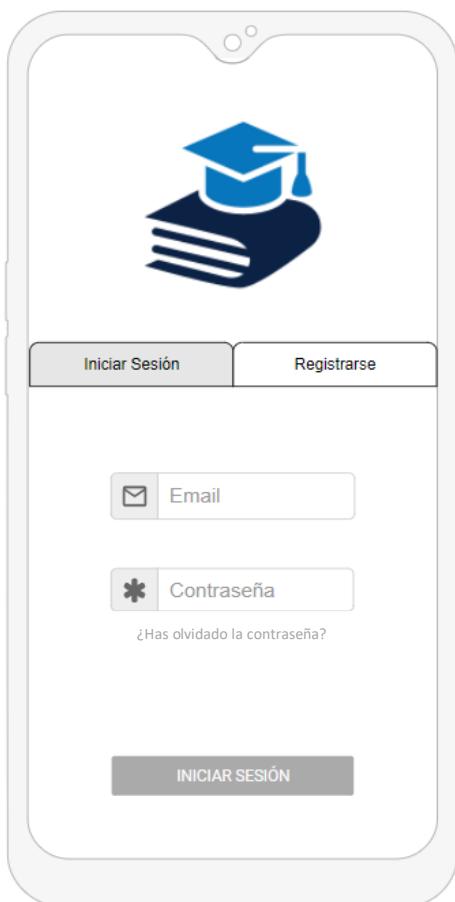
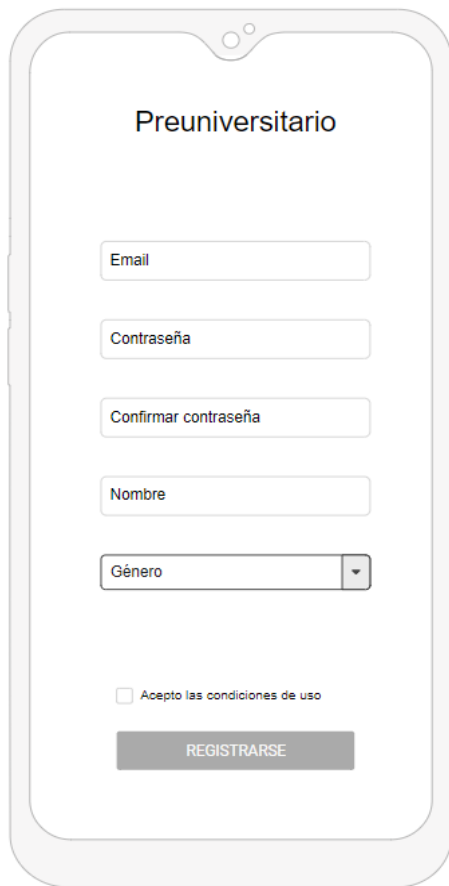


Figura 12 – Prototipo de iniciar sesión



Figura 11 – Prototipo de registrarse 1

En la Figura 13 se observa la segunda parte del caso de uso de registrarse, en ella el usuario rellena el formulario con sus datos para crear un nuevo perfil. En Figura 14 se muestra el prototipo del menú que permite navegar al usuario entre el home, los chats, el perfil y los ajustes. Para acceder al menú, el usuario desplaza la pantalla con el dedo hacia la derecha. En el menú se encuentra el caso de uso de cerrar sesión.



Preuniversitario

Email

Contraseña

Confirmar contraseña

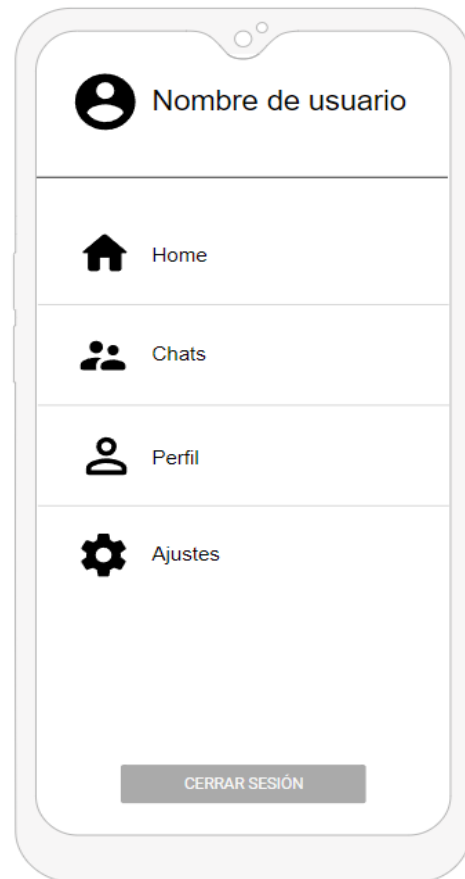
Nombre

Género

Acepto las condiciones de uso

REGISTRARSE

Figura 13 – Prototipo de registrarse 2



Nombre de usuario

Home

Chats

Perfil

Ajustes

CERRAR SESIÓN

Figura 14 – Prototipo de menú

En la Figura 15 se puede ver el caso de uso de mostrar universidades y en la Figura 16 el caso de uso de mostrar grados. Un usuario pulsa sobre la universidad que tiene interés y a continuación se le muestran los grados de la universidad seleccionada.

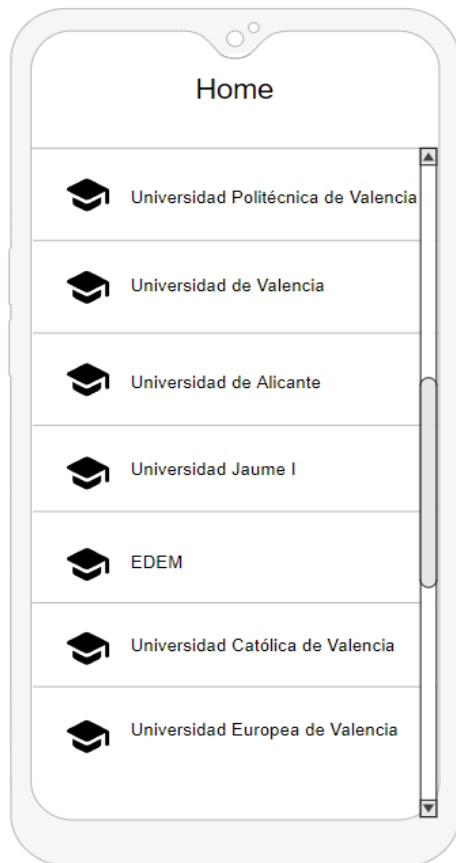


Figura 15 – Prototipo de home



Figura 16 – Prototipo de lista de grados

Cuando el usuario pulsa un grado, la aplicación le muestra los 4 tipos de foros (asignaturas, profesores, prácticas y Erasmus). Desplazando la pantalla a la izquierda aparecen las preguntas más frecuentes que tienen los usuarios sobre el grado seleccionado.

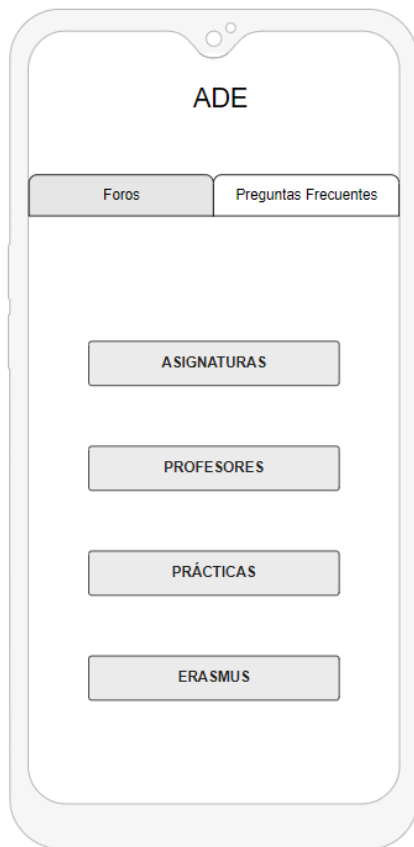


Figura 18 – Prototipo de foros

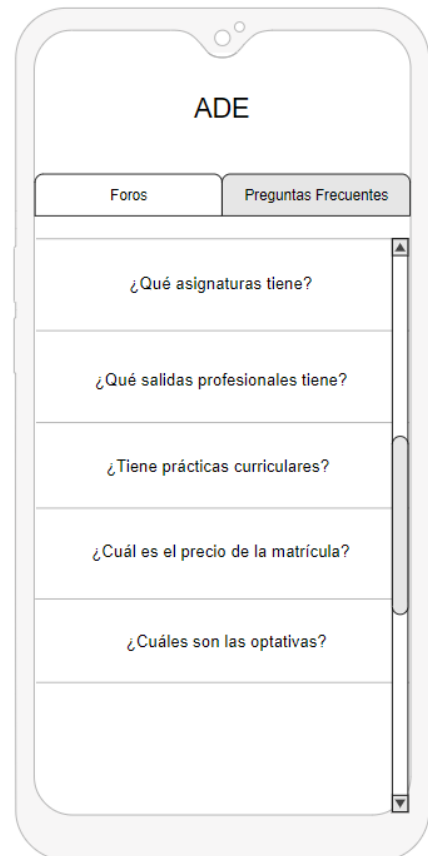


Figura 17 – Prototipo de preguntas frecuentes

Una vez el usuario ha seleccionado un tipo de foro, en este caso el de asignaturas, la aplicación muestra el caso de uso de mostrar opiniones como se puede ver en la Figura 20. El orden en el que aparecen las opiniones puede variar, si nos encontramos en la pestaña de destacados aparecerán las opiniones ordenadas por me gustas y en la pestaña de recientes aparecerán las opiniones ordenadas por la fecha de publicación. Pulsando en el botón flotante de abajo a la derecha se accede al prototipo de la Figura 19, donde el usuario puede crear opiniones.

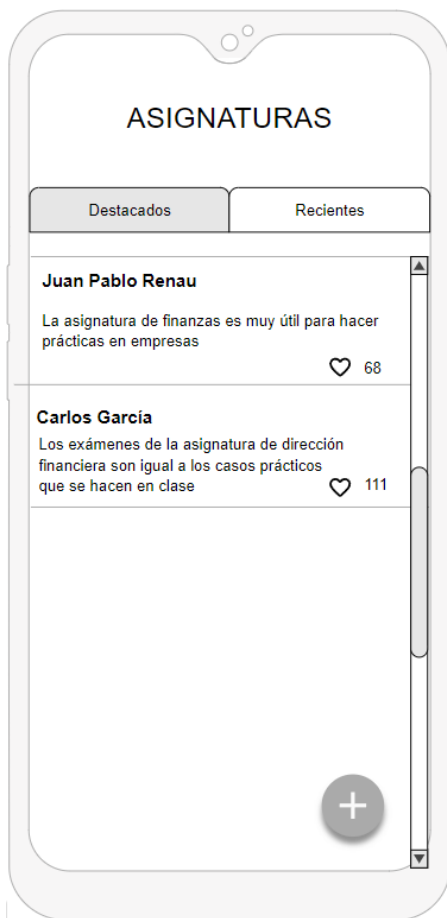


Figura 20 – Prototipo de mostrar opiniones



Figura 19 – Prototipo de crear opinión

En la Figura 21 se puede observar el prototipo de perfil en la que se encuentra el caso de uso de cambiar la foto de perfil y en la Figura 22 se muestra el prototipo de ajustes en la que, de momento, solo hay el caso de uso de borrar la cuenta del usuario.



Figura 21 – Prototipo de perfil

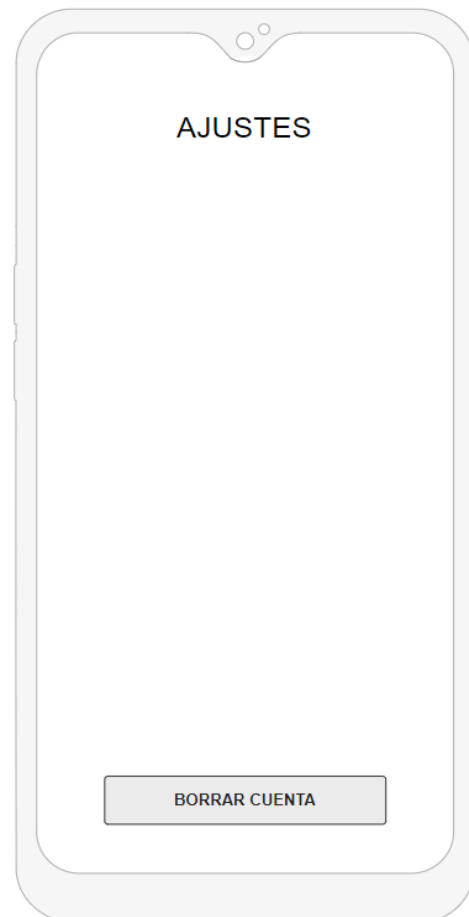


Figura 22 – Prototipo de ajustes

Por un lado, en la Figura 24 se puede observar el prototipo de mostrar chats de un preuniversitario (ya que la interfaz de los universitarios no tiene el botón que redirige al usuario a la interfaz de buscar universitario). Manteniendo pulsado sobre un chat aparece un menú flotante, como se puede ver en la Figura 25, con la opción de puntuar la ayuda del universitario o bloquear usuario (los universitarios solo tienen la opción de bloquear usuario). Por otro lado, seleccionando sobre un chat se accede al prototipo de la Figura 23 donde se encuentra el caso de uso de enviar mensaje.



Figura 24 – Prototipo de mostrar chats

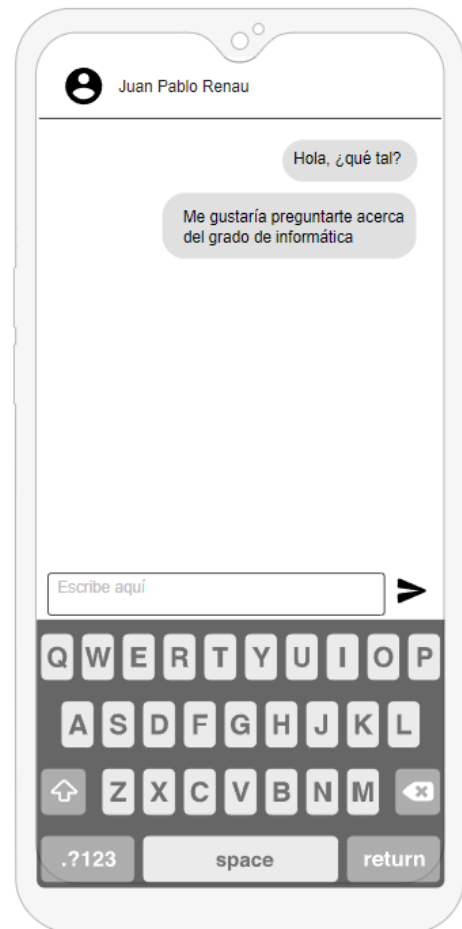


Figura 23 – Prototipo de chat

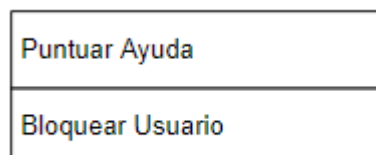


Figura 25 – Menú flotante de mostrar chats

En la Figura 27 se puede ver el caso de uso de puntuar ayuda y en la Figura 26 el caso de uso de buscar universitario.

Prototipo de pantalla 'Puntuar Ayuda'. El título es 'Puntuar Ayuda'. Hay un campo de entrada con el texto 'Escribe aquí un número del 1 al 10'. Debajo hay un campo de entrada con el texto 'Escribe aquí cómo ha sido la ayuda'. En la parte inferior hay un botón 'GUARDAR'.

Figura 27 – Prototipo de puntuar ayuda

Prototipo de pantalla 'BUSCAR UNIVERSITARIO'. El título es 'BUSCAR UNIVERSITARIO'. Hay tres campos de entrada con botones de flecha hacia abajo: 'Provincia', 'Universidad' y 'Grado'. En la parte inferior hay un botón 'BUSCAR'.

Figura 26 – Prototipo de buscar universitario

4.4 Diseño de la base de datos

El almacenamiento de datos se realizará mediante una de las bases de datos que ofrece Firebase llamada Realtime Database. Esta base de datos se caracteriza por guardar los datos en la nube, sincronizar los datos entre los usuarios en tiempo real y mantener los datos disponibles cuando la aplicación no tiene acceso a Internet.

Realtime Database se caracteriza por almacenar los datos utilizando un modelo NoSQL, es decir, guarda los datos sin utilizar tablas ni asociaciones entre ellas. Almacena los datos en colecciones de datos en formato JSON, cuyo nombre corresponde a las siglas JavaScript *Object Notation* o Notación de Objetos de JavaScript. Es un formato que es sencillo de leer y escribir para los programadores y fácil de interpretar y crear para las máquinas.

En Searchcity se utilizan 8 colecciones como se puede ver en la Figura 28:

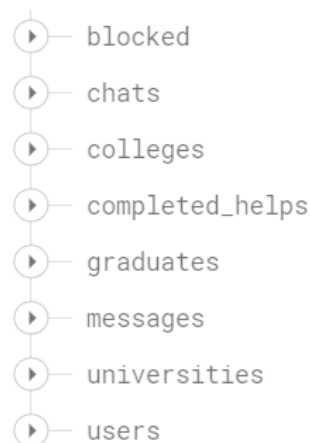


Figura 28 – Colecciones de datos que utiliza Searchcity

- **Blocked.** Almacena los usuarios bloqueados. Permite saber qué usuario bloqueó a qué usuario y cuándo lo hizo. La colección *blocked* se puede ver en la Figura 29.

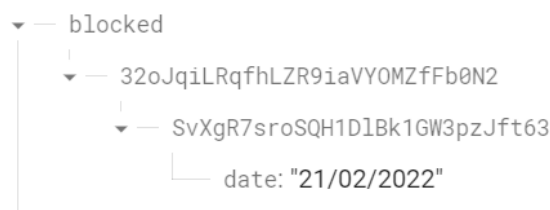


Figura 29 – Colección de blocked

- **Chats.** Almacena los chats de los usuarios e información relacionada de los chats. Permite saber los usuarios con los que está hablando un usuario. La colección *chats* se puede ver en la Figura 30.



Figura 30 – Colección de chats

- **Completed_helps.** Guarda las puntuaciones que han recibido los universitarios. Permite saber qué preuniversitario calificó a qué universitario y qué puntuación le dio. La colección *completed_helps* se puede ver en la Figura 31.



Figura 31 – Colección de completed_helps

- **Messages.** Almacena los mensajes entre usuarios e información de éstos. La colección *messages* se puede ver en la Figura 32.

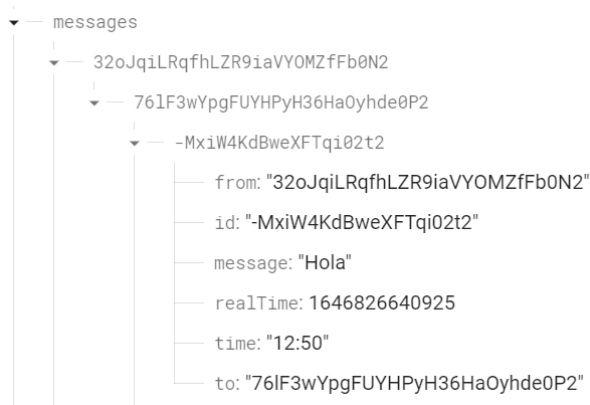


Figura 32 – Colección de messages

- **Users.** Guarda información de los preuniversitarios, de los universitarios y de los graduados. En la Figura 33 no aparecen todos los campos del usuario.

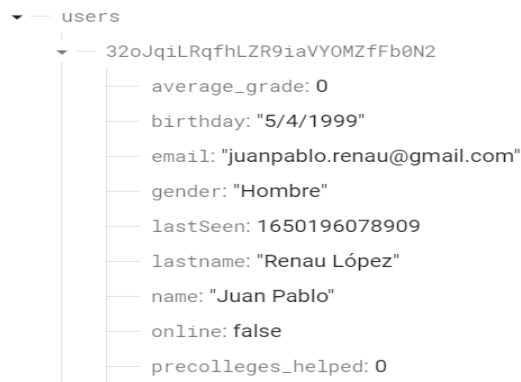


Figura 33 – Colección de users

- **Universities.** Guarda las universidades y sus grados. Además, almacena los foros y las opiniones de cada grado. La colección *universities* se puede ver en la Figura 34.



Figura 34 – Colección de universities

- **Colleges y Graduates.** Almacena simplemente el id de los universitarios y los graduados para que la búsqueda de éstos sea inmediata usando solo el id y no tener que realizar una búsqueda por la colección de *Users*. La colección *colleges* se puede ver en la Figura 35.

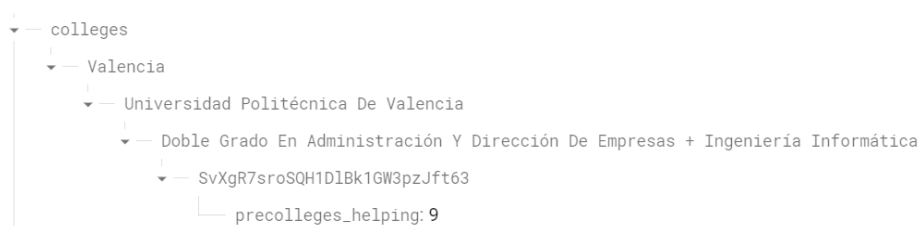


Figura 35 – Colección de colleges

5. Tecnología utilizada

En este capítulo de la memoria se explicarán las tecnologías utilizadas para el desarrollo de la aplicación, así como los motivos por los que se escogieron.

5.1 Android

android 

Figura 36 – Logo de Android

La primera decisión a la hora de desarrollar una aplicación móvil es elegir el tipo de sistema operativo donde se ejecutará. Un sistema operativo es el software que se encarga de iniciar el dispositivo, gestionar los recursos hardware (almacenamiento, batería, botones, pantalla...) y software del sistema con el objetivo de ejecutar aplicaciones y otros programas. Existen distintos tipos de sistema operativo en el mercado entre los que destacan: Android⁵, iOS, Symbian, Windows Phone y BlackBerry OS. Según estudios realizados por Statista en España, país donde se lanzará la aplicación, en enero de 2022 Android representaba el 79,68% de la cuota de mercado de dispositivos móviles, iOS el 19,8% y el resto de sistemas solo el 0,52%. Dado que se pretende llegar al máximo número de usuarios se eligió Android como el sistema operativo de Searchity. (Mobile Operating System Market Share Spain, 2022)

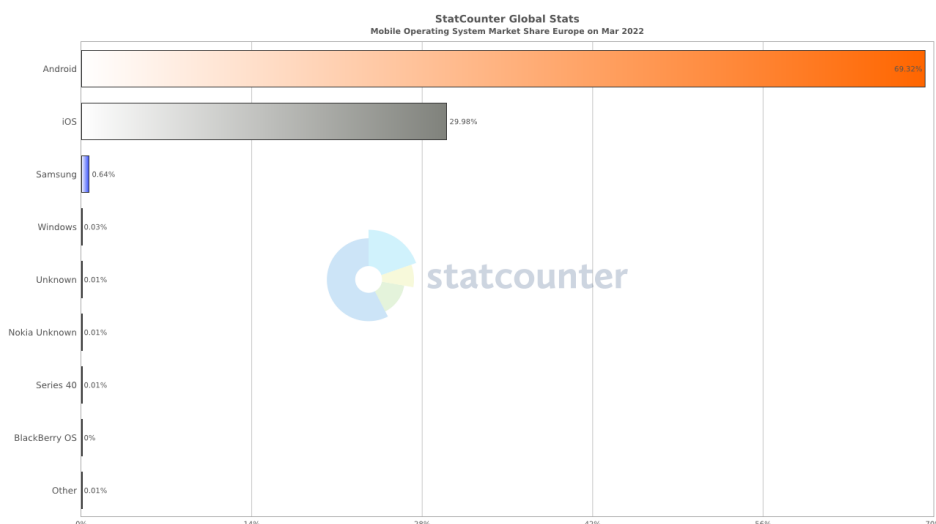


Figura 37 – Cuota de mercado de los sistemas operativos en España en marzo de 2022

⁵ <https://www.android.com/>

5.2 Android Studio



Figura 38 – Logo de Android Studio

Una vez elegido el sistema operativo, hay que escoger el entorno de desarrollo. Un entorno de desarrollo es un conjunto de procedimientos y herramientas que facilitan el desarrollo de un programa ya que permite escribir, probar y depurar código y visualizar la interfaz gráfica de usuario. En el caso de desarrollar una aplicación Android, existen varios entornos pero los principales son Android Studio⁶, Xamarin y Eclipse. Por un lado, Android Studio es el IDE por excelencia ya que es el entorno más completo que se puede encontrar. Cuenta con la documentación oficial y las librerías de Google y con un emulador que permite montar una máquina virtual donde probar y depurar las apps. Por otro lado, Xamarin es un entorno creado por Microsoft y está orientado para crear apps multiplataforma, es decir, para que la app pueda usarse en varios sistemas operativos. Por último, Eclipse era el entorno más usado hasta que llegó Android Studio. De hecho, tienen muchas similitudes. Sin embargo, al no estar respaldado por Google, el desarrollo de aplicaciones es más lento y dejar pasar errores es más fácil. Comparando los 3 entornos se eligió Android Studio por dos motivos. El primero porque, como se ha mencionado anteriormente, es el IDE más completo y es el que recomienda Google, y el segundo porque como Searchcity va a estar orientado para Android no hace falta programar en un entorno multiplataforma. (Sims, 2019)

5.3 Java



Figura 39 – Logo de Java

En cuanto al lenguaje de programación, destacan Java⁷, C# y Kotlin. Por un lado, Java es el lenguaje oficial por excelencia para el desarrollo de aplicaciones Android y es soportado por Android Studio. Por otro lado, C# es un lenguaje un poco más sencillo que Java aunque son bastante similares. Es soportado por Xamarin y su uso se centra en el desarrollo de videojuegos. Por último, Kotlin es un nuevo lenguaje oficial para programación Android y fue desarrollado por Google. De los 3 lenguajes es el más simple y limpio ya que tiene menos formalidades y reglas por lo que cada vez son más las empresas que usan Kotlin para desarrollar aplicaciones. Aunque la opción más válida a priori es Kotlin, se escogió programar Searchcity con Java porque era el lenguaje que más dominaba ya que lo había aprendido en la universidad y consideraba que programar una red social desde cero ya suponía un gran reto como para programarla con un lenguaje que desconocía. Además, aunque el uso de Kotlin está en auge, a día de hoy Java sigue siendo el lenguaje más usado para programar aplicaciones Android. (Gawin, 2021)

⁶ <https://developer.android.com/>

⁷ <https://www.java.com/es/>

5.4 Firebase



Figura 40 – Logo de Firebase

A la hora de elegir la base de datos, donde se almacenará la información de Searchity de manera estructurada, se plantearon varios escenarios. Al principio, mi idea era usar el ordenador portátil como servidor. Sin embargo, suponía que la aplicación solo estaría disponible cuando el portátil estuviera encendido por lo que no era una práctica viable. Luego, pensé en alquilar un servidor en la nube como aws de Amazon pero los precios me parecían un poco elevados sobre todo para una aplicación que no tiene una fuente de ingresos. Por ello, mi objetivo era encontrar un servidor en la nube que fuese gratuito. La búsqueda terminó cuando descubrí Firebase⁸, una plataforma gratuita desarrollada por Google que ofrece portentosos servicios para desarrollar, gestionar y mejorar aplicaciones móviles y sitios web. Entre los servicios que Firebase ofrece destacan:

- **Autenticación de usuarios.** Este servicio se encarga de gestionar el iniciar sesión, el registrarse y el borrar cuenta de los usuarios para que no puedan acceder a la aplicación usuarios no autenticados.
- **Base de datos.** Firebase proporciona dos soluciones de bases de datos NoSQL basadas en la nube que te permiten almacenar datos:
 - **Cloud Firestore.** Es la base de datos más reciente de Firebase. Cuenta con un nuevo modelo de datos más sencillo e intuitivo. Además, con Cloud Firestore se pueden realizar consultas más ricas y rápidas que Realtime Database.
 - **Realtime Databe.** Es la base de datos original de Firebase, destinada para aplicaciones que necesitan sincronizar los datos entre los usuarios en tiempo real.

Tabla 20 – Comparación de las características de Cloud Firestore y Realtime Database

Características	Cloud Firestore	Realtime Database
Función principal	Realizar búsquedas, transacciones y ordenamientos avanzados	Sincronizar datos con consultas básicas principalmente.
Tamaño	Cientos de GB o TB de datos que se leen con mucha más frecuencia de la que cambian.	Algunos GB que cambian con frecuencia.
Disponibilidad	Garantía de un 99.95%	Garantía de un 99.999%

⁸ <https://firebase.google.com/>



En la Tabla 20 se puede apreciar como Realtime Database se adapta mejor a Searchity ya que al ser una red social se necesita una sincronización constante de datos en tiempo real y con una disponibilidad elevada por lo que fue la base de datos elegida.

- **Storage.** Un servicio donde se pueden almacenar distintos archivos como imágenes o vídeos.
- **Crashlytics.** Muestra estadísticas de errores y a qué usuario le ha ocurrido el error, incluso indica la línea en la que se ha producido facilitando considerablemente la depuración del código.
- **Test Lab.** Usa dispositivos reales en un centro de datos de Google para probar tu app. Permite tener una idea más clara de cómo se comportará la aplicación en dispositivos reales.
- **Extensiones.** Firebase cuenta con servicios adicionales que se pueden agregar como procesos que controlan los pagos de la aplicación, traductores, entre otros.

Por tanto, Firebase no se limita a ser solo un servicio de base de datos sino que ofrece un conjunto de servicios que facilitan y mejoran el desarrollo de las aplicaciones por lo que me decanté por Firebase para que diera soporte a Searchity.

5.5 GitHub



Figura 41 – Logo de GitHub

GitHub⁹ es un repositorio online y gratuito que permite subir y gestionar el código de los proyectos de los desarrolladores. Utiliza Git, como sistema de control, para poder ver las diferencias entre versiones, restaurar versiones antiguas, trabajar con distintas ramas del proyecto, añadir modificaciones al código o depurar los errores. Además, se puede integrar a Android Studio.

5.6 Lucidchart



Figura 42 – Logo de Lucidchart

Lucidchart¹⁰ es una herramienta online que permite a los usuarios crear distintos tipos de diagramas (como diagramas de flujo, diseños UML, mapas mentales, organigramas, esquemas y muchos otros tipos) y almacenarlos en la nube. Para realizar los diagramas cuenta con una gran variedad de figuras y formas clasificadas por temática. Además, permite trabajar en un mismo documento a varios usuarios en tiempo real.

⁹ <https://github.com/>

¹⁰ <https://www.lucidchart.com>

5.7 Moqups



Figura 43 – Logo de Moqups

Moqups¹¹ es un sitio web que ofrece la posibilidad de crear diagramas, esquemas, tablas y gráficos, pero su principal herramienta es la creación de prototipos (uso por el que se ha utilizado Moqups). Para el diseño de los prototipos, Moqups ofrece modelos, plantillas, una gran cantidad de imágenes e iconos que ayudan a validar el diseño de las interfaces de usuario de sitios web y aplicaciones móviles.

¹¹ <https://app.moqups.com/>



6. Implementación

En este apartado se expondrá el aspecto final de Searchcity. Para ello, se mostrarán las interfaces de usuario finales y se explicará para cada una de ellas su composición y su funcionamiento.

6.1 Presentación

En la Figura 44 se puede observar la interfaz de inicio que ven los usuarios siempre que abren la aplicación. Tiene una duración de 1,5 segundos y se utiliza para presentar la aplicación con el objetivo de que los usuarios interioricen el logo y el nombre.



Figura 44 – Interfaz de presentación

6.2 Inicio de sesión

La interfaz de inicio de sesión como se puede ver en la Figura 45 está formada por 2 pestañas que se accede pulsando sobre ellas o desplazando la pantalla. Para conseguirlo, se ha creado:

- Un **activity**. Es la interfaz principal que muestra la aplicación.
- Dos **fragments**. Un *fragment* es una parte de la UI que es reutilizable y que necesita de un *activity* para poder existir. En este caso, se utiliza uno para la pestaña de iniciar sesión y otro para la pestaña de registrarse.
- Una **LoginAdapter**. Es una clase que extiende de *FragmentStateAdapter* que se encarga de gestionar las pestañas

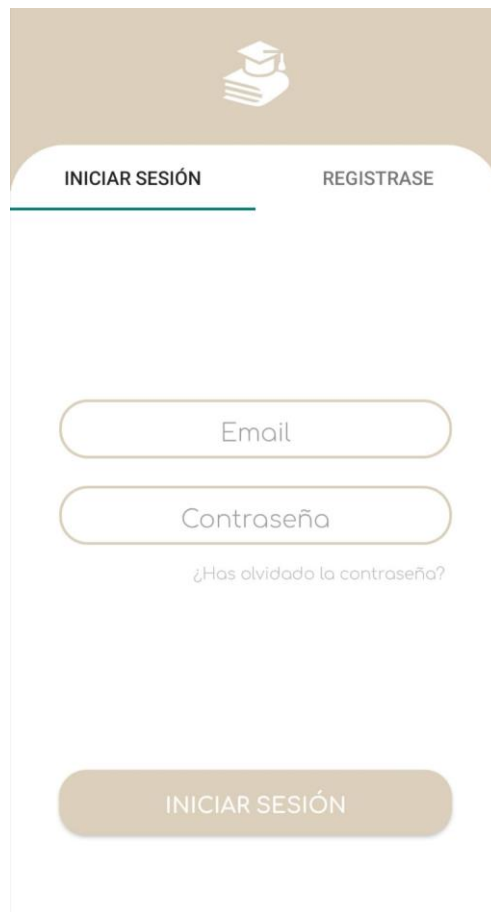


Figura 45 – Interfaz de inicio de sesión

En cuanto al funcionamiento, el usuario introduce su correo y su contraseña y pulsa en el botón de iniciar sesión. En ese momento, la aplicación comprueba el formato y la longitud de los campos y, si son válidos, el servicio de autenticación de Firebase comprueba si los datos pertenecen a un usuario existente. En este caso, la aplicación le

muestra la interfaz de home. En caso contrario, le muestra un mensaje de error. Además, el servicio de autenticación también permite restablecer la contraseña del usuario. Para ello, el usuario introduce su email y pulsa en el texto de “¿Has olvidado la contraseña?”. En unos pocos segundos, Firebase le enviará un correo con un enlace para poder cambiar la contraseña.

6.3 Registro

En la Figura 47 se puede observar la interfaz de la pestaña de registrarse. En ella, se le pregunta qué tipo de usuario es para redirigirlo a la interfaz de registro correspondiente. Cada tipo de usuario (preuniversitario, universitario y graduado) tiene una interfaz distinta ya que algunos atributos son únicos. Por ejemplo, el universitario tiene que rellenar la universidad donde estudia, el grado que realiza y el curso en el que se encuentra.



Figura 47 – Interfaz de registrarse 1

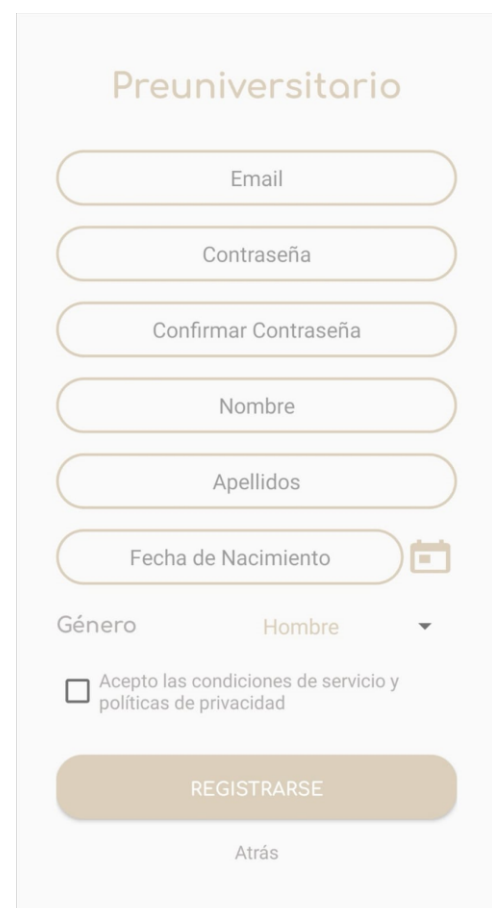


Figura 46 – Interfaz de registrarse 2

En la Figura 46 se muestra la interfaz de registrarse 2 de un preuniversitario. El funcionamiento es igual que al iniciar sesión. Una vez el usuario ha rellenado los campos, el sistema comprueba si el formato es válido y, si es así, crea un nuevo usuario y le muestra la interfaz de home. En caso contrario, aparece un mensaje de error explicando el problema.

6.4 Menú

En La Figura 48 se puede apreciar la interfaz de menú que se ha desarrollado mediante un *NavigationView*, un elemento de interfaz que permite la navegación en las aplicaciones móviles. Éste está formado por 3 elementos:

- Un *menu_header* en la parte de arriba que contiene la foto de perfil y el nombre del usuario.
- Un *drawer_menu* formado por cuatro ítems (Home, Chats, Perfil y Ajustes) que pulsando sobre ellos se puede acceder a su *activity* correspondiente.
- Un botón que permite cerrar la sesión del usuario. El servicio de autenticación de Firesebase se encarga de esta tarea.

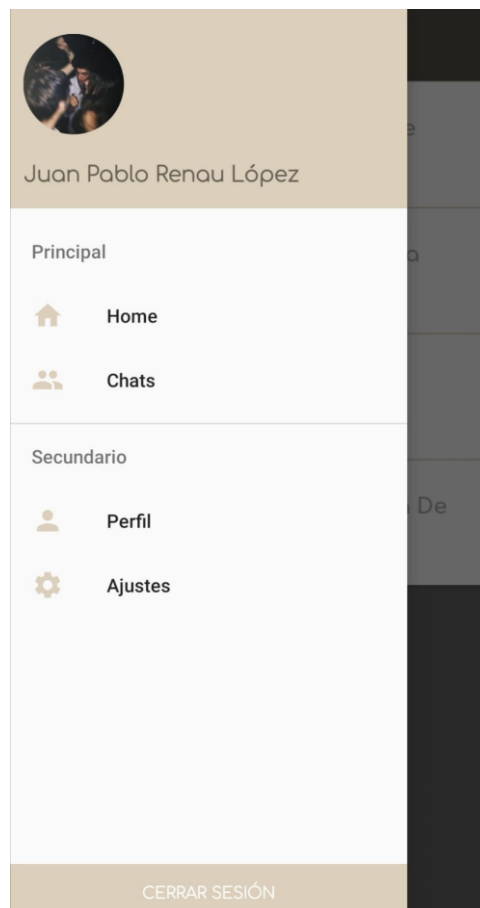


Figura 48 – Interfaz de menú

6.4.1 Home

La interfaz de *home* es la pantalla principal de la aplicación y muestra un listado de universidades de la Comunidad Valenciana. En *home* el usuario puede seleccionar la universidad sobre la que tenga alguna duda o simplemente busque información y la aplicación le mostrará el listado de grados que tiene dicha universidad (Searchcity aún no contiene todas las universidades ni todos sus grados).

Como se puede apreciar en la Figura 49 y en la Figura 50, ambas interfaces son un listado de elementos que necesitan de los siguientes elementos:

- Un *RecyclerView* es un componente que se agrega a la vista y se encarga de gestionar los contenedores que forman la lista.
- Una clase que extienda de *RecyclerView.ViewHolder* que se encarga de definir los contenedores y atribuirle los datos. En el caso de la interfaz de *home*, le añade la imagen y el nombre al contenedor.
- Una clase que extienda de *RecyclerView.Adapter* que tiene el objetivo de vincular los contenedores (los *ViewHolders*) a la vista (el *RecyclerView*).

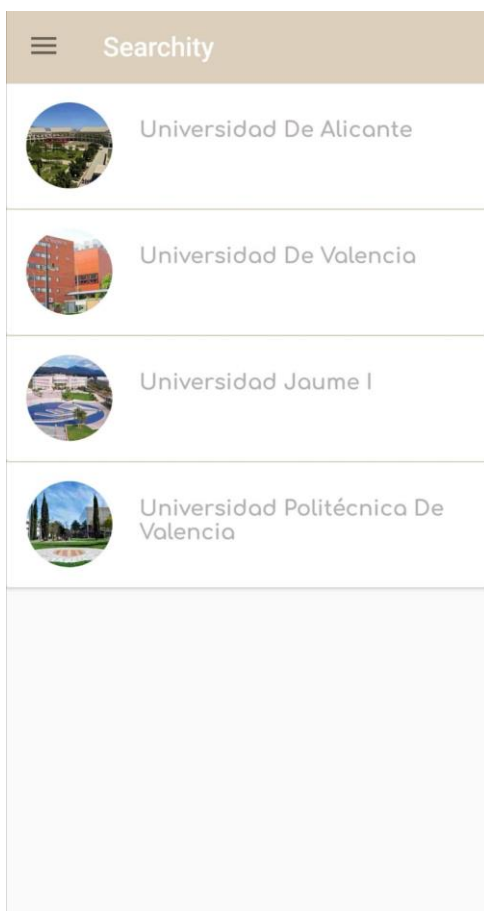


Figura 50 – Interfaz de home

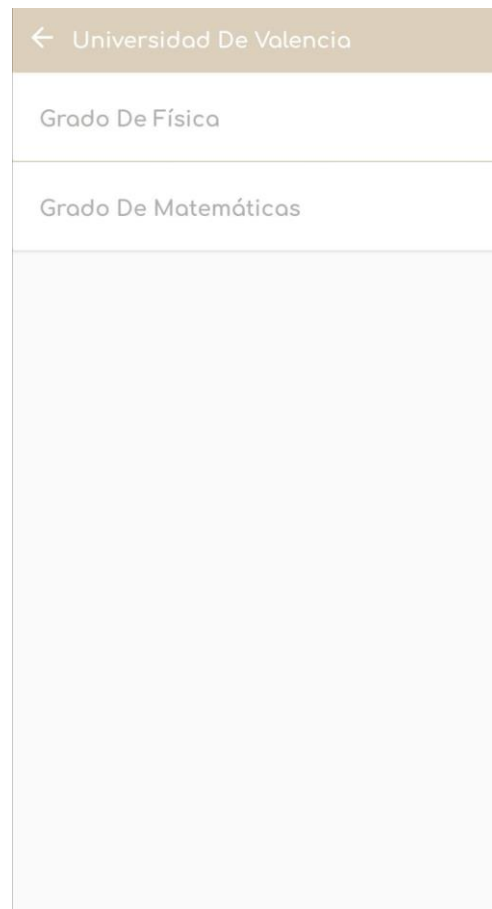


Figura 49 – Interfaz de mostrar grados

Una vez el usuario ha seleccionado un grado, aparece la interfaz de la Figura 52. Está formada por un *activity* y dos *fragments*, uno para la pestaña de foros y otro para la pestaña de preguntas frecuentes. El *fragment* de foros contiene cuatro botones, cada uno para un foro distinto: un foro de asignaturas, un foro de profesores, un foro de prácticas y un foro de Erasmus.

Desplazando la pantalla hacia la izquierda se accede a la interfaz de la Figura 51 donde aparecen un listado de preguntas frecuentes. De momento, no hay preguntas frecuentes por lo que la aplicación muestra un mensaje por pantalla. Este *fragment* contiene un *RecyclerView* para gestionar de manera dinámica las preguntas.

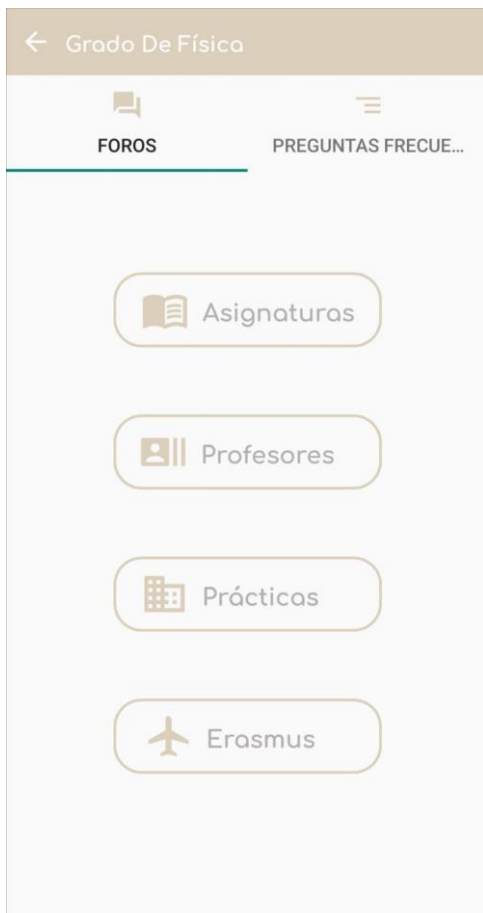


Figura 52 – Interfaz de foros

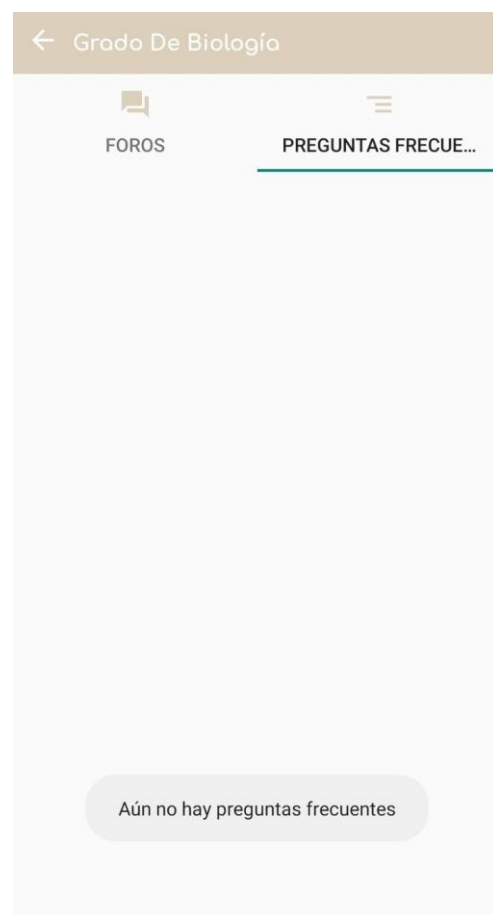


Figura 51 – Interfaz de preguntas frecuentes

Cuando el usuario selecciona un foro, el sistema le muestra la interfaz de la Figura 54. En ella, se pueden ver las opiniones de dicho foro ordenados por el número de me gustas y, si se desplaza la pantalla hacia la izquierda, están ordenadas por la fecha de publicación. El listado de opiniones de destacados y de recientes se ha desarrollado siguiendo la misma estructura que el listado de preguntas frecuentes, con un *RecyclerView*. Sin embargo, en cada listado se realiza una llamada distinta a la base de datos de Firebase como se puede apreciar en las líneas de código siguientes:

```
Query opinionsQuery = opinionReference.orderByChild("likes")
```

```
Query opinionsQuery = opinionReference.orderByChild("realtime")
```

Pulsando sobre el botón flotante con el signo de más, el sistema muestra la interfaz de la Figura 53 que permite al usuario escribir una opinión y publicarla. Cuando el usuario pulsa en el botón de aceptar, la aplicación válida el texto y, si es correcto, Firebase guarda la nueva publicación en la base de datos que será detectada por el *RecyclerView* y añadida al resto de opiniones. En caso contrario, aparecerá un mensaje de error.



Figura 54 – Interfaz de mostrar opiniones

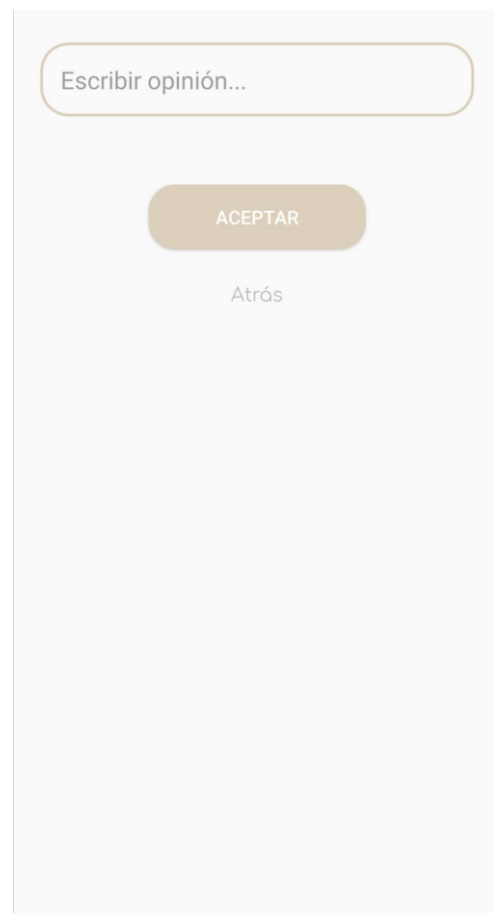


Figura 53 – Interfaz de crear opinión

6.4.2 Chats

Los chats son la segunda funcionalidad más importante de Searchity. Tienen por objetivo conectar a preuniversitarios con universitarios y establecer conversaciones en tiempo real. En la Figura 56 se puede observar un listado de los chats que tiene un usuario y un botón flotante que permite crear un nuevo chat con un universitario. Este botón solo es visible para los preuniversitarios ya que son los que inician la interacción. Una vez el preuniversitario ha creado un nuevo chat, se le aparecerá directamente al universitario. Cuando un usuario selecciona un chat, la aplicación le dirige a la interfaz de la Figura 55. Esta interfaz contiene los elementos clásicos de un chat:

- En la parte superior se encuentra la información acerca del usuario. Está formada por su nombre, su última conexión y su foto de perfil.
- En el centro aparecen los mensajes ordenados por fecha de publicación gracias a un *RecyclerView*.
- En la parte inferior hay un campo de texto y un botón de enviar que comprueba el campo y guarda el nuevo mensaje en la colección de *messages* de Firebase.

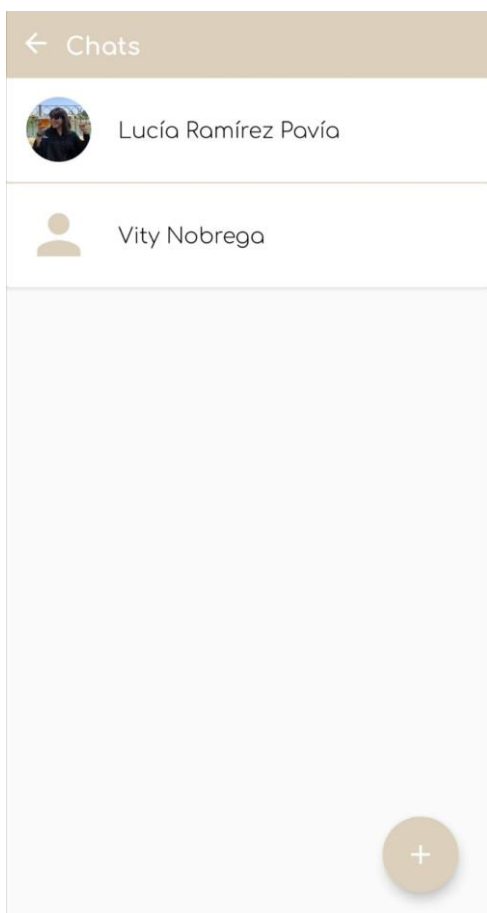


Figura 56 – Interfaz de chats

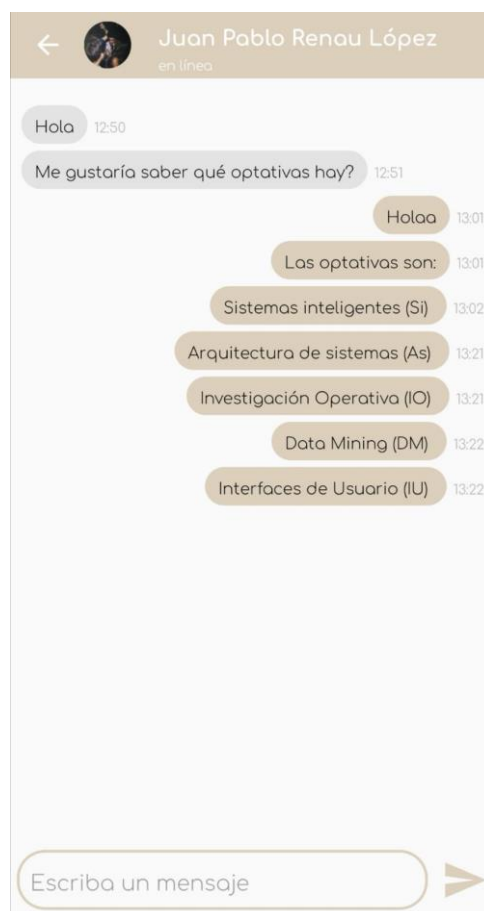


Figura 55 – Interfaz de chat

Manteniendo pulsado sobre un chat, se muestra el menú flotante de la Figura 58. El primer texto dirige al usuario a la interfaz de la Figura 57 y el segundo bloquea al usuario seleccionado. Para conseguir la función de mantener pulsado, hace falta lo siguiente:

- En la clase *RecyclerView.Adapter* se crea una interfaz con un método sin implementar, sin código, que tendrá como argumento el índice de la posición del chat en el *RecyclerView*.
- En la clase *ViewHolder* se implementa el método *View.OnLongClickListener* y se sobrescribe el método utilizando el método de la interfaz del *RecyclerView.Adapter* y pasándole como argumento la posición del *ViewHolder*.
- En la clase del *RecyclerView* se implementa el método del *RecyclerView.Adapter* y se sobrescribe. Aquí es donde se añade la lógica una vez se ha mantenido pulsado el chat.

La interfaz de la Figura 58 permite realizar una valoración de la ayuda del universitario. Para ello, el sistema comprueba el formato y si es correcto la guarda en Firebase.

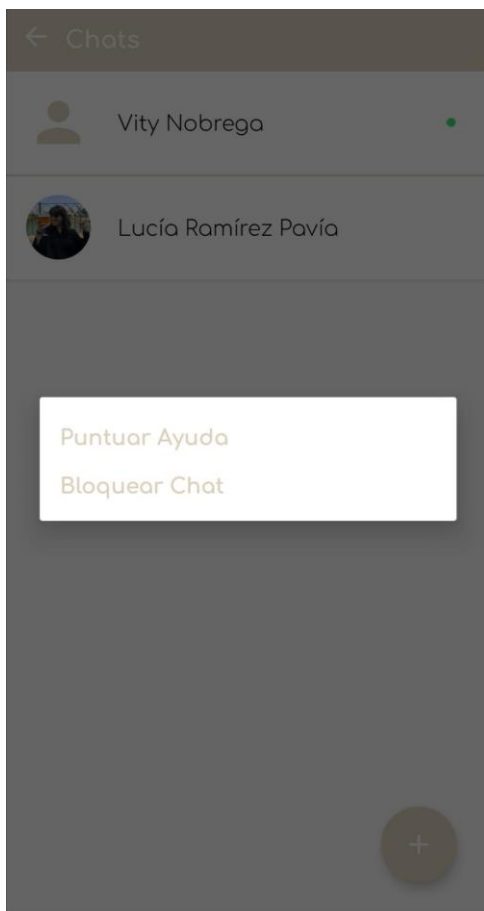


Figura 58 – Interfaz de menú flotante



Figura 57 – Interfaz de puntuar ayuda

Cuando un preuniversitario pulsa el botón flotante para crear un chat aparece la interfaz de la Figura 59. En ella, el preuniversitario puede seleccionar las características del usuario con el que quiere establecer una conversación:

- Tipo de usuario: universitario o graduado.
- Provincia: Castellón, Valencia o Alicante.
- Universidad: un listado de las universidades de la provincia elegida.
- Grado: un listado de los grados de la universidad seleccionada.

Una vez el usuario ha escogido todos los campos y pulsado en el botón de aceptar, la aplicación busca un usuario con esas características que se encuentre disponible.



The image shows a mobile application interface titled "Crear Chat". It features four dropdown menus for selection:

- Tipo de Usuario:** Set to "Universitario".
- Provincia:** Set to "Valencia".
- Universidad:** Set to "Universidad Politécnica De Valencia".
- Grado:** Set to "Doble Grado En Administración Y Dirección De Empresas + Ingeniería Informática".

At the bottom, there is a large "ACEPTAR" button and a smaller "Atrás" button.

Figura 59 – Interfaz de crear chat

6.4.3 Perfil

En la Figura 60 se puede observar el perfil de un universitario que contiene los siguientes elementos:

- El grado que cursa el universitario
- La universidad donde estudia.
- El número de preuniversitarios que ha ayudado.
- La calificación media de las ayudas que ha realizado a los preuniversitarios.

Pulsando sobre el botón, la aplicación accede a la galería del usuario y permite cambiar la foto de perfil del usuario.



Figura 60 – Interfaz de perfil

6.4.4 Ajustes

En la Figura 61 se puede observar la interfaz de ajustes. Contiene el botón de borrar cuenta que elimina todos los datos del usuario que se encuentran en la base de datos. En un futuro habrá más opciones de ajustes como elegir el idioma o cambiar los colores de la aplicación.

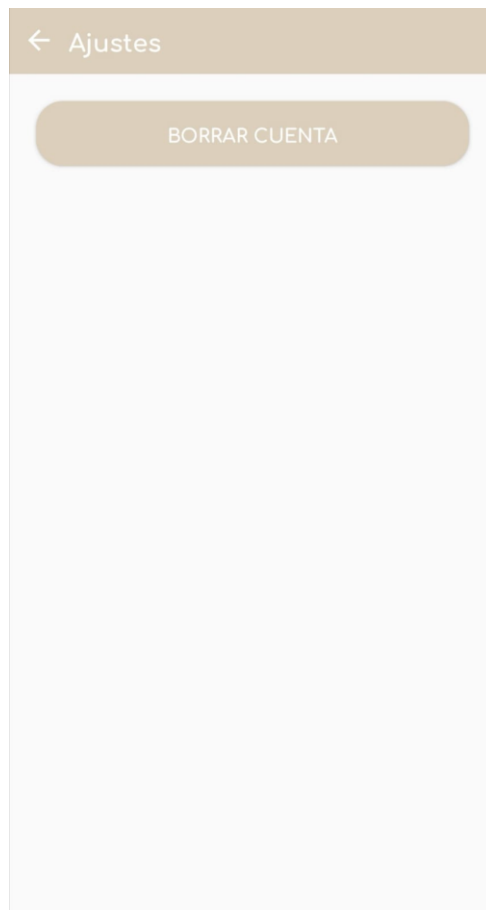


Figura 61 – Interfaz de ajustes

7. Pruebas

La realización de pruebas en las aplicaciones móviles permite la detección temprana de errores por lo que es un aspecto clave para asegurar la calidad del software, garantizar la satisfacción del cliente y ahorrar costes futuros. En Searchcity se han realizado dos tipos de pruebas distintas que se explicarán a continuación: pruebas de usuario y pruebas de calidad.

7.1 Pruebas de usuario

Para llevar a cabo las pruebas de usuario, se ha realizado un cuestionario¹² para determinar y calificar la experiencia que han tenido los usuarios con la aplicación. El cuestionario está formado por seis preguntas y las respuestas están basadas en la escala Likert, un método de medición que se utiliza para conocer el grado de conformidad o disconformidad de los encuestados. La muestra del cuestionario está formada por 25 personas: 10 preuniversitarios y 15 universitarios de distintas universidades de la Comunidad Valenciana. Los encuestados descargaron la aplicación mediante un archivo APK. (Muguira, 2022)

En la Figura 62, se puede ver los resultados de la primera pregunta del cuestionario en la que el 100% de los encuestados afirman que la aplicación se carga rápidamente. Esta pregunta es muy importante ya que, a los usuarios, sobre todo a la gente joven, les molesta cuando las aplicaciones van lentas.

Los datos de la aplicación se cargan rápidamente

25 respuestas

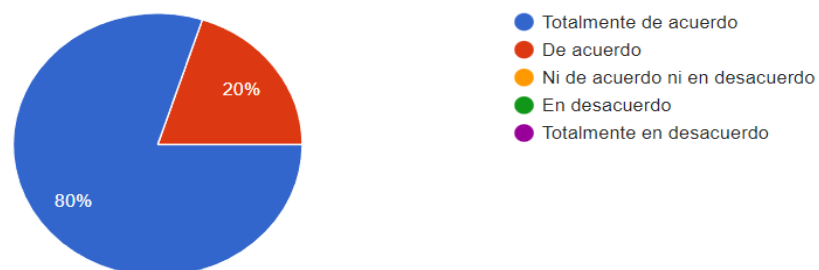


Figura 62 – Primera pregunta de las pruebas de usuario

¹² <https://forms.gle/T64U95t5r4Dih1hE9>

Un aspecto importante es que los elementos visuales de la aplicación no se vean distorsionados ni pixelados ya que reduce la calidad de la aplicación y puede molestar a la vista a los usuarios, provocando que no se sientan cómodos al utilizarla. En la Figura 63, se observa que los elementos visuales se ven con claridad.

Los elementos visuales no aparecen distorsionados ni pixelados

25 respuestas

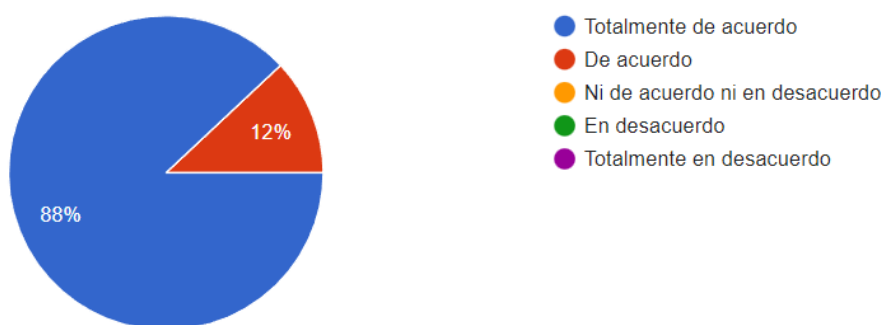


Figura 63 – Segunda pregunta de las pruebas de usuario

Como se aprecia en la Figura 64, todos los encuestados consideran que la navegación por la aplicación es buena ya que los usuarios pueden acceder a las distintas interfaces de la aplicación en unos pocos clics. De esta manera, al tener una navegación rápida y sencilla, los usuarios tardan menos tiempo en encontrar la información que desean.

La navegación por la aplicación es fácil y ágil

25 respuestas

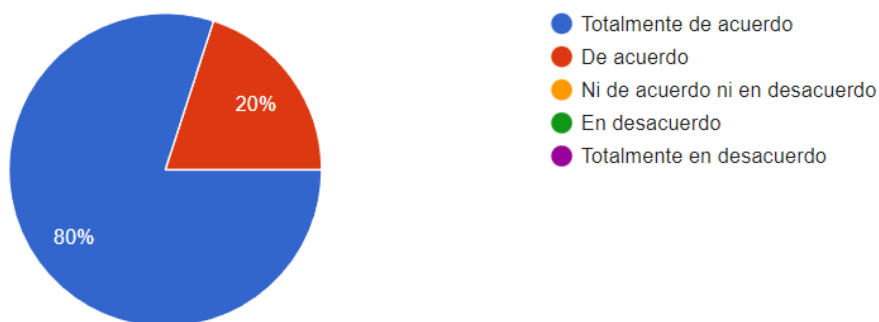


Figura 64 – Tercera pregunta de las pruebas de usuario

Uno de los objetivos propuestos era conseguir que la aplicación fuera atractiva para el usuario con el fin de aumentar el valor de la aplicación y la comodidad de los usuarios. Al sentirse los usuarios cómodos pasarán más tiempo en la aplicación. En la Figura 65 se puede apreciar que se ha conseguido dicho objetivo.

Las interfaces son atractivas

25 respuestas

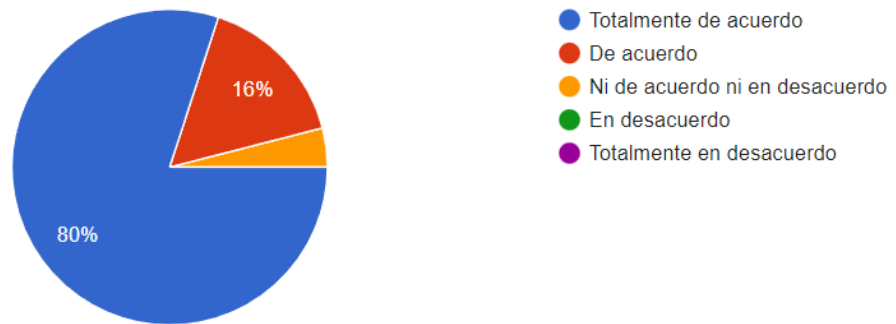


Figura 65 – Cuarta pregunta de las pruebas de usuario

Otro de los objetivos era lograr interfaces intuitivas. Si una aplicación es difícil o confusa de utilizar, los usuarios evitarán su uso. En la Figura 66, se comprueba que las interfaces de la aplicación son claras e intuitivas.

Las interfaces son intuitivas

25 respuestas

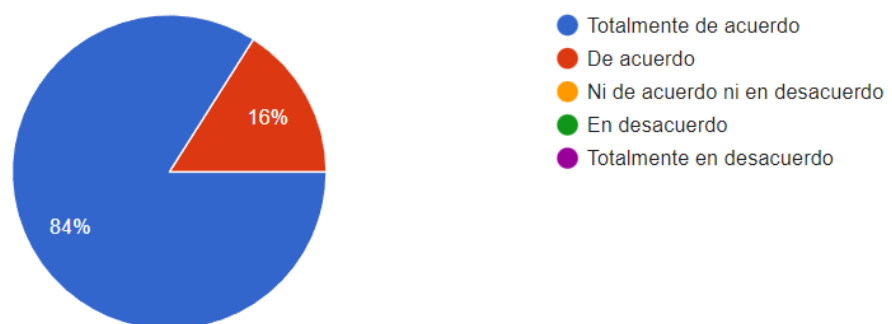


Figura 66 – Quinta pregunta de las pruebas de usuario

Por último, en la Figura 67 se muestran los resultados de la sexta pregunta de las pruebas de usuario. Esta pregunta tenía el objetivo de comprobar si los usuarios encontraban algún fallo que no se había podido identificar con anterioridad. Se puede observar que de los 25 usuarios, ninguno encontró un fallo.

Sin embargo, aunque ninguno de los usuarios haya encontrado un error no significa que la aplicación esté libre de errores.

La aplicación no falla ni se bloquea
25 respuestas

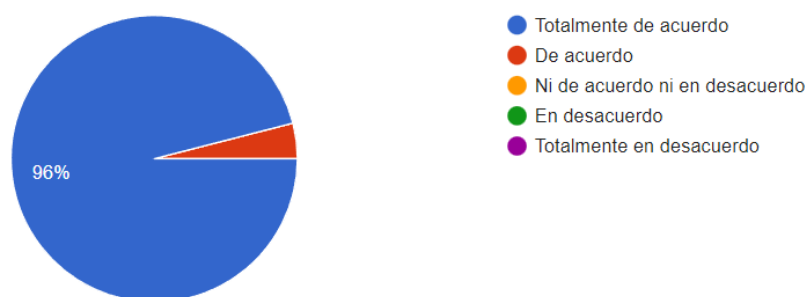


Figura 67 – Sexta pregunta de las pruebas de usuario

En conclusión, las pruebas de usuario han sido un éxito porque todas las respuestas del cuestionario han sido satisfactorias.

7.2 Pruebas de calidad

Google ofrece un listado de criterios y pruebas que ayudan a los programadores a evaluar la calidad de sus aplicaciones. Además, permiten garantizar la calidad mínima que deben tener todas las aplicaciones. Están organizados en distintas áreas: experiencia visual, funcionalidad, rendimiento y estabilidad, privacidad y seguridad y Google Play (esta área no se realizará ya que ya la aplicación aún no se ha subido al público). (Android Developers, 2021)

7.2.1 Experiencia visual

Toda aplicación debe cumplir los patrones de interacción y diseño visual estándares de Android con el objetivo de asegurar una experiencia visual del usuario intuitiva y coherente.

Tabla 21 – Pruebas de la experiencia visual de Google

Área	Pruebas	Descripción	Resultado
Navegación	CR-0	Se puede navegar por todas las interfaces de la aplicación.	Cumple.
	CR-3	La aplicación admite la navegación de retroceder mediante el botón atrás del dispositivo.	Cumple.
	CR-5	Las interfaces permiten la orientación horizontal y vertical.	Cumple.
	CR-1, CR-3 y CR-5	La aplicación mantiene correctamente el estado del usuario.	Cumple.
Notificaciones	CR-9	Las notificaciones no realizan promoción.	No se aplica. Actualmente, la aplicación no dispone de notificaciones.
		Cada notificación se envía a un canal adecuado.	
		Las notificaciones tienen prioridad.	
		Si es posible, las notificaciones se agrupan.	
		Las notificaciones son persistentes.	
IU y gráficos	CR-5	La aplicación ocupa toda la pantalla en ambas orientaciones.	Cumple.
		La aplicación renderiza bien entre las transiciones de orientación.	Cumple.
Calidad visual	CR-Todos	La aplicación muestra sus elementos visuales sin distorsión ni pixelado.	Cumple.

		La aplicación admite el tema oscuro.	No cumple. El tema oscuro aún no se ha implementado.
--	--	--------------------------------------	--

En la Tabla 21 se puede apreciar que se cumplen todas las pruebas excepto dos: las notificaciones y el tema oscuro. Estos aspectos se implementarán en un futuro próximo por lo que se concluye que la aplicación cumple el área de experiencia visual.

7.2.2 Funcionalidad

En cuanto a funcionalidad, se puede observar en la Tabla 22 que Searchity no se aplica ningún apartado ya que no reproduce audio ni videos.

Tabla 22 – Pruebas de la funcionalidad de Google

Prueba	ID	Descripción	Resultado
Audio	CR-1 y CR-8	La reproducción de audio se reanuda cuando la aplicación vuelve a primer plano.	No se aplica. La aplicación no reproduce audio.
	CR-1, CR-2 y CR-8	Si la función principal es el audio, la aplicación debe permitir la reproducción en segundo plano.	
Multimedia	CR-0	La reproducción de videos debe admitir la reproducción pantalla en pantalla (una pequeña pantalla que permite ver el video mientras el usuario puede navegar por otras aplicaciones)	No se aplica. La aplicación no reproduce videos

7.2.3 Rendimiento y estabilidad

Searchity presenta un buen rendimiento y estabilidad debido a que no falla ni se bloquea, carga rápidamente sus elementos visuales y los datos de la base de datos y proporciona indicadores de progreso como se muestra en la Tabla 23.

Tabla 23 – Pruebas del rendimiento y la estabilidad de Google

Prueba	ID	Descripción	Resultado
Estabilidad	CR-Todos	La aplicación no falla ni se bloquea.	Cumple.
Rendimiento	CR-Todos	La aplicación se carga rápidamente.	Cumple.
		Proporciona indicadores de progresos.	

7.2.4 Privacidad y seguridad

Por último, la aplicación cumple con todas las pruebas de privacidad y seguridad como se indica en la Tabla 24.

Tabla 24 – Pruebas de la privacidad y seguridad de Google

Prueba	ID	Descripción	Resultado
Permisos	SC-4 y CR-0	La aplicación solicita la cantidad mínima de permisos necesaria para su funcionamiento.	Cumple.
		Solicita los permisos cuando son necesarios, en lugar de hacerlo cuando se inicia la aplicación.	
Datos y archivos	SC-1	Los datos sensibles no son accesibles por otras aplicaciones.	Cumple.
Identidad	CR-0	Proporciona sugerencias para autocompletar las credenciales de la cuenta del usuario.	Cumple.
Bibliotecas	SP-2	Las bibliotecas de la aplicación están actualizadas.	Cumple.
		No se utilizan bibliotecas de depuración que ralentizan la ejecución.	

8. Conclusiones

Searchity surgió con el propósito de crear un lugar donde los preuniversitarios pudieran obtener información que les ayudara a la hora de elegir una carrera universitaria. Para ello, se establecieron una serie de objetivos que servían como guía para el desarrollo de la aplicación. De los ocho objetivos que se determinaron en el apartado 1.2 de *Objetivos* de este documento se han cumplido 7 por lo que se considera la realización de Searchity todo un éxito. El único objetivo que no se ha llevado a cabo es publicar la aplicación en la Play Store debido a que durante la realización de este documento se me ocurrieron algunas mejoras que me gustaría implementar antes de publicarla. Este objetivo no afecta a que la aplicación se encuentre completa y terminada conforme a los objetivos establecidos.

Esta red social es un proyecto que se ha desarrollado desde cero por lo que en el proceso han surgido algunas dificultades. La principal dificultad ha sido que al no tener una asignatura durante la carrera de desarrollo de aplicaciones móviles me encontraba bastante perdido al inicio debido a que había muchos conocimientos y tecnologías que desconocía. Por ejemplo, un aspecto que me llevó bastante tiempo fue implementar la base de datos ya que sabía realizar consultas a la base de datos pero no crear una desde cero. Además, tenía el problema de que aunque la creara, estaría implantada en mi ordenador y éste debería estar siempre encendido para garantizar la disponibilidad. Sin embargo, durante la búsqueda de soluciones para los problemas que se me presentaban con la base de datos encontré Firebase, una plataforma de Google que resolvía todas las dificultades que me estaban apareciendo ya que disponía de una base de datos en la nube que ya estaba desarrollada.

En el transcurso del proyecto, se han cometido algunos errores que han alargado la duración del mismo pero que me han aportado ya que he aprendido de ellos. Uno de esos errores, que considero el más importante, es empezar a realizar el proyecto sin haberlo planificado bien porque aunque la planificación es un aspecto que requiere de tiempo, a la larga facilita y agiliza muchísimo el desarrollo de la aplicación.

Por último, gracias a Searchity he aprendido muchísimos conocimientos y nuevas tecnologías que desconocía. Por un lado, he aprendido conocimientos muy importantes en el desarrollo de las aplicaciones móviles como los distintos patrones de diseño que existen así como saber implementar el patrón de MVVM. Por otro lado, he aprendido a usar las diferentes tecnologías mencionadas en el capítulo 5 de *Herramientas utilizadas*, en específico Android Studio y Firebase (dos herramientas clave para el desarrollo de aplicaciones móviles).



Searchcity: una aplicación móvil para la ayuda en la toma de decisión de escoger una carrera universitaria

En definitiva, gracias a este proyecto he aprendido los conocimientos y habilidades necesarios para la creación de aplicaciones móviles desde cero y más importante aún, a saber a lo que me quiero dedicar el día de mañana.

8.1 Relación del trabajo desarrollado con los estudios cursados

Al haber estudiado el Doble Grado de Ingeniería Informática y ADE, no he tenido la oportunidad de cursar la asignatura optativa de “Desarrollo de aplicaciones para dispositivos móviles” por lo que no he contado con una asignatura que tuviera una relación directa con este proyecto. Sin embargo, he cursado otras asignaturas que me han permitido tener una base para poder desarrollar la aplicación.

En primer lugar, asignaturas como “Introducción a la informática y la programación”, “Programación” y “Estructuras de datos y algoritmos” me enseñaron a tener un conocimiento avanzado de Java, lenguaje que he utilizado para programar Searchcity.

En segundo lugar, en la asignatura de “Ingeniería de software” aprendí a identificar los requisitos funcionales de una aplicación y a saber realizar sus diagramas de casos de uso, así como sus especificaciones.

Finalmente, la asignatura de “Gestión de proyectos” me ayudó a tener una mejor organización y planificación en el desarrollo de la aplicación.

En conclusión, el Grado de Informática me ha permitido sentar las bases del desarrollo de software que me servirán para enfrentarme a diferentes proyectos y desafíos en el futuro.

9. Trabajos futuros

A pesar de que Searchity es una aplicación completamente funcional y que ha cumplido todos los objetivos que se habían establecido en la planificación, existen algunas mejoras que podrían realizarse. Estas modificaciones se han ordenado en función de su dificultad, de menos a más, y son las siguientes:

- **Tema oscuro.** La aplicación admitirá la posibilidad de cambiar la temática de colores de claros a oscuros. Para ello, sustituirá el blanco por el negro y el beige por uno con una tonalidad más oscura. De esta manera, la aplicación cumplirá el apartado de calidad visual del área de “Experiencia Visual” que como se comentó en el capítulo de *Pruebas*, no lo hacía.
- **Sistema de obtención de puntos.** Hasta ahora el algoritmo de búsqueda de un universitario dependía de tres parámetros: las características introducidas por el preuniversitario (tipo de usuario, universidad y grado), el número de preuniversitarios que el universitario está ayudando (un máximo de 10) y si el universitario había sido bloqueado por el preuniversitario. Por tanto, para que la búsqueda fuera exitosa tendría que existir un usuario con las características definidas por el preuniversitario que estuviera ayudando a menos de diez preuniversitarios y que no hubiera sido bloqueado por el preuniversitario que realiza la búsqueda. Sin embargo, ahora se añadirá un nuevo parámetro que se tendrá en cuenta para la búsqueda y son los puntos del universitario. Cuando un universitario termina una ayuda recibirá una calificación del uno al diez que se añadirá a los puntos del universitario. Estos puntos serán acumulativos por lo que a cuantos más preuniversitarios ayude y mejores calificaciones reciba, más puntos tendrá. Además, los puntos también se podrán conseguir en función del número de me gustas de las opiniones que haya publicado el universitario. De esta manera, los universitarios con más puntos serán seleccionados antes en la creación de un chat porque son aquellos que están ayudando más y mejor a los preuniversitarios. El sistema de obtención de puntos tiene el objetivo de garantizar y mejorar la calidad de la ayuda de Searchity.

- **Comentarios.** De momento, en los foros, los usuarios solo pueden publicar opiniones y no preguntas y respuestas. Por ello, se quiere implementar la posibilidad de escribir comentarios en una opinión. De esta forma, los preuniversitarios podrán publicar una pregunta y los universitarios podrán responderla, haciendo que los foros sean más ricos y dinámicos.
- **Notificaciones.** Un sistema de implementación de notificaciones de tipo *push* mediante la plataforma de Firebase. Las notificaciones se utilizarán para los siguientes casos: cuando se dan me gustas a las opiniones, cuando se reciben nuevos mensajes y cuando un preuniversitario crea un chat con un universitario.

Una vez se hayan implementado estas mejoras, se publicará Searchity en la tienda de aplicaciones Play Store. De esta manera, se habrán cumplido todos los objetivos que se propusieron al inicio del proyecto.

10. Referencias

- Android Developers. (30 de Marzo de 2021). *Calidad básica de las apps | Desarrolladores de Android*. Recuperado el 10 de Mayo de 2022, de Android Developers: <https://developer.android.com/docs/quality-guidelines/core-app-quality?hl=es-419#CR-3>
- BBVA. (25 de abril de 2019). *Fundación BBVA*. Recuperado el 12 de Enero de 2022, de Un 33% de los alumnos no finaliza el grado que inició y un 21% abandona sin terminar estudios universitarios - FBBVA: <https://www.fbbva.es/noticias/un-33-de-los-alumnos-no-finaliza-el-grado-que-inicio-y-un-21-abandona-sin-terminar-estudios-universitarios/#>
- Cuesta, L. (30 de Noviembre de 2021). *El impacto de la tecnología en los jóvenes*. Recuperado el 16 de Enero de 2022, de <https://www.lavanguardia.com/vida/junior-report/20211130/7897657/impacto-tecnologia-jovenes.html>
- Dorsey, J. (Septiembre de 2021). *Jason Dorsey*. Recuperado el 16 de Enero de 2022, de Gen Z and tech dependency: How the youngest generation interacts differently with the digital world: <https://jasondorsey.com/blog/gen-z-and-tech-dependency-how-the-youngest-generation-interacts-differently-with-the-digital-world/>
- Gawin, M. (2021). *INVO Blog*. Recuperado el 2 de Febrero de 2022, de Best Android App Development Tools & Software: <https://invotech.co/blog/best-android-app-development-tools-software/>
- INE. (Marzo de 2021). *Indicadores de la Agenda 2030 para el Desarrollo Sostenible*. Recuperado el 27 de Mayo de 2022, de Instituto Nacional de Estadística: <https://www.ine.es/dyngs/ODS/es/index.htm>
- Mena, M. (30 de Agosto de 2021). *Statista*. Recuperado el 2 de Febrero de 2022, de Infografía: Android e iOS dominan el mercado de los smartphones: <https://es.statista.com/grafico/18920/cuota-de-mercado-mundial-de-smartphones-por-sistema-operativo/>

Searchcity: una aplicación móvil para la ayuda en la toma de decisión de escoger una carrera universitaria

Mobile Operating System Market Share Spain. (2022). Recuperado el 26 de Marzo de 2022, de Statcounter Global Stats: <https://gs.statcounter.com/os-market-share/mobile/spain>

Moran, M. (2021). *Naciones Unidas*. Recuperado el 8 de Febrero de 2022, de Objetivo 4: Garantizar una educación inclusiva, equitativa y de calidad y promover oportunidades de aprendizaje durante toda la vida para todos: <https://www.un.org/sustainabledevelopment/es/education/>

Muguirra, A. (2022). *Escala de Likert: Qué es y cómo utilizarla en tus encuestas*. Recuperado el 14 de Mayo de 2022, de QuestionPro: <https://www.questionpro.com/blog/es/que-es-la-escala-de-likert-y-como-utilizarla/>

ONU. (2019). *¿Qué es la pobreza? | ONG Manos Unidas*. Recuperado el 26 de Mayo de 2022, de Manos Unidas: <https://www.manosunidas.org/observatorio/pobreza-mundo/definicion-pobreza#:~:text=Esta%20idea%20inspir%C3%B3%20la%20creaci%C3%B3n,US%241%2C90%20al%20d%C3%ADa.>

PICK-DREAM ayuda a decidir carrera universitaria con test vocacionales. (Marzo de 2021). Recuperado el 28 de Enero de 2022, de Imagina una forma exitosa de ELEGIR CARRERA: <https://www.pick-dream.com/#/inicio>

Ridwan, A. (9 de Enero de 2021). *Introduction to the Solid Principle | Engineering Education (EngEd) Program*. Recuperado el 21 de Abril de 2022, de Section.io: <https://www.section.io/engineering-education/introduction-to-solid-principle/>

Rodriguez, E. (21 de diciembre de 2020). *MVVM – Qué es y como funciona*. Recuperado el 28 de Abril de 2022, de Inmediatum: <https://inmediatum.com/blog/ingenieria/mvvm-que-es-y-como-funciona/>

Sims, G. (10 de Agosto de 2019). *I want to develop Android apps — What languages should I learn?* Recuperado el 19 de Marzo de 2022, de Android Authority: <https://www.androidauthority.com/develop-android-apps-languages-learn-391008/>

UNISCOPIO . (2021). Recuperado el 28 de Enero de 2022, de Plataforma Digital de Orientación al Estudiante.: <https://uniscopio.com/>

UPV. (2017). *Residuos : Unidad de Medio Ambiente*. Recuperado el 27 de Mayo de 2022, de UPV: <https://www.upv.es/entidades/AMAPUOC/infoweb/ov/info/1000100normalc.html>

UPV. (7 de Marzo de 2022). *La UPV, con motivo del Día Internacional de la Mujer, se suma a la campaña HeForShe por la igualdad de género e invita a la comunidad universitaria a unirse a la campaña sumando sus retos | Universitat Politècnica de València*. Recuperado el 27 de Mayo de 2022, de UPV:
<http://www.upv.es/noticias-upv/noticia-13481--heforshe-es.html>

What Is Software Architecture - Examples, Tools, & Design | CAST. (2021). Recuperado el 29 de Abril de 2022, de CAST Software:
<https://www.castsoftware.com/glossary/what-is-software-architecture-tools-design-definition-explanation-best>

11. Anexo

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (los ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.			X	
ODS 4. Educación de calidad.		X		
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.			X	
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

Reflexión sobre la relación del TFG con los ODS más relacionados.

Miles de millones de personas intentan sobrevivir con menos de 1,9€ al día; existen colectivos discriminados por sexo, raza, religión, orientación sexual y poder adquisitivo; los derechos humanos básicos no están garantizados en la mayoría de los países del mundo; aumenta la desigualdad entre ricos y pobres; el incremento de la contaminación afecta a la biodiversidad, al medio ambiente y al clima; vivimos en un modelo socioeconómico que no es consciente de que los recursos del planeta son finitos... Por ello, en 2015, para solventar esta realidad, se aprobó en las Naciones Unidas un acuerdo internacional, la Agenda 2030. Un acuerdo cuyo propósito era ponerles fin a estos problemas, mediante la colaboración de todos, antes de 2030. Esta agenda está formada por 17 objetivos con 169 metas y 231 indicadores. (INE, 2021) (ONU, 2019)

Son muchos los países, empresas e instituciones que se han comprometido y han tomado acción para estar más cerca de estos objetivos. Por ejemplo, la Universidad Politécnica de Valencia tiene programas para combatir la desigualdad, así como medidas que buscan el objetivo de las 3 erres (reducir, reutilizar y reciclar) y que permiten convertirla en una institución más sostenible con el medio ambiente. (UPV, 2017) (UPV, 2022)

En cuanto a Searchity, también está implicada con la agenda 2030. De hecho, está relacionada con 3 objetivos: educación de calidad, salud y bienestar y trabajo decente y crecimiento económico (el objetivo de educación de calidad es el que está más relacionado con la aplicación, los otros dos son más una consecuencia de este objetivo).

En primer lugar, el propósito principal de Searchity es ayudar a los preuniversitarios, es decir, proporcionarles una herramienta más donde puedan obtener información acerca de los grados que les gustaría cursar o de los que tienen dudas. Para conseguirlo, la aplicación ofrece, como se ha comentado a lo largo de este documento, dos funcionalidades principales: un foro dividido en diferentes categorías (prácticas, Erasmus, asignaturas y profesores) donde se pueden publicar opiniones y la creación de un chat entre un preuniversitario y un universitario para que puedan conversar en tiempo real. Por tanto, Searchity no busca mejorar la calidad de la educación sino garantizar un mejor acceso a la misma ya que si un preuniversitario está más informado sobre cómo son las carreras, podrá tomar una mejor decisión y habrá menos probabilidades de que se equivoque. Por este motivo, considero que el grado de relación con el objetivo de educación de calidad es medio.

En segundo lugar, al reducir el número de universitarios que han tomado una decisión equivocada, se reducirá el número de universitarios que no estén a gusto en su carrera. Una carrera es un desafío muy duro que requiere de mucho esfuerzo, dedicación y constancia por lo que supone una carga mental para aquellos alumnos que prefieran estudiar otra carrera porque pensaban que la actual sería de otra manera. Por ello, Searchity busca indirectamente favorecer el bienestar y la salud mental de los universitarios por lo que considero que tiene una relación baja con el objetivo de salud y bienestar.

Por último, Searchity tiene relación, aunque en menor medida, con el objetivo de trabajo y crecimiento sostenible. Como aumenta el número de universitarios que están contentos en la carrera, éstos no se cambiarán de carrera por lo que la acabarán antes y, por tanto, entrarán al mercado laboral a una edad más temprana, favoreciendo que tendrán más oportunidades de conseguir un trabajo que les guste.