



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**“YO ENTRO A LA U”: APLICACIÓN CON CONSEJOS Y MARCOS CURRICULARES PARA
AYUDAR A LAS PERSONAS DEL INTERIOR DEL PAÍS A ENTRAR A LA UNIVERSIDAD**

Miguel Angel Omar Ruano Roca

Asesorado por el Ing. Jhonatan Wilfredo Pú Morales

Guatemala, marzo de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**"YO ENTRO A LA U": APLICACIÓN CON CONSEJOS Y MARCOS CURRICULARES PARA
AYUDAR A LAS PERSONAS DEL INTERIOR DEL PAÍS A ENTRAR A LA UNIVERSIDAD**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

MIGUEL ANGEL OMAR RUANO ROCA

ASESORADO POR EL ING. JHONATAN WILFREDO PÚ MORALES

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, MARZO DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Vladimir Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Marlon Francisco Orellana López
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
EXAMINADOR	Ing. César Augusto Fernández Cáceres
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

“YO ENTRO A LA U”: APLICACIÓN CON CONSEJOS Y MARCOS CURRICULARES PARA AYUDAR A LAS PERSONAS DEL INTERIOR DEL PAÍS A ENTRAR A LA UNIVERSIDAD

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 13 de agosto de 2020.

Miguel Angel Omar Ruano Roca

Guatemala, 18 de noviembre de 2020

Ingeniero
Carlos Alfredo Azurdia
Coordinador de Privados y Trabajos de Tesis
Escuela de Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería - USAC

Respetable Ingeniero Azurdia:

Por este medio hago de su conocimiento que en mi rol de asesor del trabajo de investigación realizado por el estudiante **MIGUEL ANGEL OMAR RUANO ROCA** con carné **201503666** y CUI **2742 18798 0403** titulado **"Yo Entro A La U" Aplicación con consejos y marcos curriculares para ayudar a las personas del interior del país a entrar a la Universidad**, lo he revisado y luego de corroborar que el mismo se encuentra concluido y que cumple con los objetivos propuestos en el respectivo protocolo, procedo a la aprobación respectiva.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



Ing. Jhonatan Wilfredo Pu Morales
Colegiado No. 15628

Jhonatan Wilfredo Pu Morales
Ingeniero en Ciencias y Sistemas
Colegiado 15,628



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 30 de noviembre de 2020

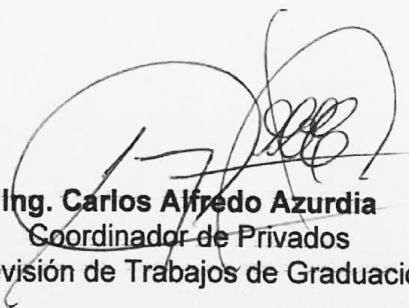
Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **MIGUEL ANGEL OMAR RUANO ROCA** con carné **201503666** y CUI **2742 18798 0403** titulado “-YO ENTRO A LA U- APLICACIÓN CON CONSEJOS Y MARCOS CURRICULARES PARA AYUDAR A LAS PERSONAS DEL INTERIOR DEL PAIS A ENTRAR A LA UNIVERSIDAD” y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“YO ENTRO A LA U”: APLICACIÓN CON CONSEJOS Y MARCOS CURRICULARES PARA AYUDAR A LAS PERSONAS DEL INTERIOR DEL PAÍS A ENTRAR A LA UNIVERSIDAD**, realizado por el estudiante, MIGUEL ANGEL OMAR RUANO ROCA aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”

Msc. Carlos Gustavo Alonzo
Director
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 22 de marzo de 2021

DTG. 124.2021.

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **“YO ENTRÓ A LA U”: APLICACIÓN CON CONSEJOS Y MARCOS CURRICULARES PARA AYUDAR A LAS PERSONAS DEL INTERIOR DEL PAÍS A ENTRAR A LA UNIVERSIDAD**, presentado por el estudiante universitario: **Miguel Angel Omar Ruano Roca**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Inga. Anabela Cordova Estrada
Decana

Guatemala, marzo de 2021.

AACE/asga

ACTO QUE DEDICO A:

Dios

Por ser mi guía en cada paso que doy diariamente y acompañarme en las buenas y en las malas.

Mis padres

Angel Augusto Ruano Quevedo y Silvia Cristina de Jesús Roca, por apoyarme desde mi niñez, por sus consejos y porque siempre creyeron en mí.

Mis hermanos

Juan Augusto, por ser siempre un modelo para seguir. Grissel Eugenia, por ser una buena hermana en todas las circunstancias. Daniela María, por estar siempre ahí para mí. Noeli Orizabal, por ser una buena amiga en cualquier momento.

Mis amigos

De San Martín Jilotepeque, Lucía Ruíz, Henry Roca, Rebeca García y Kelly Sunún, por su apoyo a través de los años.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por darme la oportunidad de desarrollarme en todos los sentidos, por permitirme ser parte de ella.
Facultad de Ingeniería	Por haberme puesto tantos retos y mostrarme de lo que soy capaz.
Familia	Por su apoyo durante el transcurso de la carrera.
Ingeniero Jonathan Pú	Por apoyarme en este trabajo de graduación y por ser tan paciente.
Mis amigos de la universidad	Por brindarme su amistad, porque siempre puedo aprender de ellos.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VII
GLOSARIO.....	IX
RESUMEN.....	XIII
OBJETIVOS	XV
INTRODUCCIÓN.....	XVII
1. MARCO TEÓRICO.....	1
1.1. El alcance de la educación superior en Guatemala	1
1.1.1. Cobertura geográfica	1
1.1.2. Carreras que se ofrecen en Guatemala	2
1.1.3. Matrícula estudiantil	2
1.2. Tecnologías para el desarrollo de aplicaciones	3
1.2.1. Plataforma de despliegue Heroku	3
1.2.2. Bases de datos NoSQL.....	4
1.2.3. Plataforma MongoDB Cloud.....	4
1.2.4. Amazon Web Services	5
1.2.5. Firebase Cloud Storage	5
1.2.6. Android Studio	5
1.2.7. Ionic Framework	6
2. IDENTIFICACIÓN DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN.....	7
2.1. Antecedentes.....	7
2.2. Identificación del problema	8
2.2.1. Acciones útiles al momento de elegir una carrera universitaria	9

	2.2.1.1.	Aprovechar la experiencia de otros.....	9
	2.2.1.2.	Comparar las carreras	9
	2.2.1.3.	Investigar opciones desconocidas	10
2.3.		Propuesta de solución.....	10
3.		ARQUITECTURA DE LA APLICACIÓN	13
3.1.		Arquitectura de despliegue del backend.....	13
	3.1.1.	Funcionamiento de la plataforma Heroku	14
	3.1.2.	Estructura del API de la aplicación	15
	3.1.3.	Conexión entre plataformas.....	15
3.2.		Arquitectura del frontend	17
	3.2.1.	Ionic Framework	17
3.3.		Alojamiento en la tienda de aplicaciones.....	18
4.		DESARROLLO DE LA APLICACIÓN.....	21
4.1.		Estructuración de la base de datos	21
	4.1.1.	Esquemas de la base de datos.....	21
		4.1.1.1. Esquema de usuarios	21
		4.1.1.2. Esquema de roles.....	21
		4.1.1.3. Esquema de publicaciones	22
		4.1.1.4. Esquema de reacciones de publicación.....	23
		4.1.1.5. Esquema de archivos	24
		4.1.1.6. Esquema de categorías	25
	4.1.2.	Diagrama de la base de datos	26
4.2.		Estructuración del API.....	27
	4.2.1.	Módulo selector de esquemas	28
	4.2.2.	Módulo genérico para consultas de base de datos	28

4.2.2.1.	Método para obtener un solo registro ...	29
4.2.2.2.	Método para obtener todos los registros	29
4.2.2.3.	Método para obtener todos los registros por filtro	29
4.2.2.4.	Método para insertar un nuevo registro	31
4.2.2.5.	Método para actualizar un registro	31
4.2.2.6.	Método para borrar un registro	31
4.3.	Estructuración de la aplicación de Ionic con Angular 10	32
4.3.1.	Reciclaje de la vista de categorías	33
5.	PROTOTIPO DE LA APLICACIÓN	37
5.1.	Página principal	37
5.2.	Pestañas inferiores	38
5.3.	Página de categoría de carreras	39
5.4.	Página del panel de carrera	40
5.5.	Página de publicaciones	42
5.6.	Módulo de reacciones	44
5.7.	Página de perfil	44
5.8.	Página para nuevas publicaciones	46
5.9.	Página para editar la publicación	47
5.10.	Página de contactos	48
5.11.	Panel de administración	49
5.12.	Página del listado de categorías	50
5.13.	Página de edición de categorías	51
5.14.	Página de recuperación de la contraseña	52

6.	ACCIONES DE USUARIO	55
6.1.	Punto de vista del usuario final de la aplicación	55
6.2.	Punto de vista del usuario administrador de la aplicación	61
7.	DOCUMENTACIÓN	65
7.1.	Requisitos	65
7.1.1.	Instalación de la aplicación desarrollada	65
7.2.	Herramientas para el desarrollo de la aplicación	65
7.2.1.	Visual Studio Code	65
7.2.2.	Android SDK.....	66
7.2.3.	Google USB Driver	66
7.2.4.	MongoDB Atlas	66
7.2.5.	Capacitor	66
7.3.	Hardware	67
7.4.	Software.....	68
7.5.	Tutorial de instalación	68
7.5.1.	Instalación en Windows	68
7.5.1.1.	Descargar e instalar Android Studio.....	68
7.5.1.2.	Descargar e instalar Visual Studio Code.....	69
7.5.1.3.	Descargar e instalar NodeJS	70
7.5.2.	Configurar dispositivo Android	70
7.6.	Acceso al repositorio y ejecución del código	71
7.6.1.	Código del backend.....	71
7.6.2.	Código del frontend	72
7.7.	Publicación en la tienda Google Play	72
	CONCLUSIONES	73
	RECOMENDACIONES	75

BIBLIOGRAFÍA.....	77
APÉNDICES.....	79

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Diagrama de funcionamiento de la plataforma Heroku	14
2.	Diagrama de funcionamiento de la plataforma	16
3.	Diagrama de arquitectura de desarrollo híbrida.....	18
4.	Panel de administración de Google Play Console	19
5.	Diagrama de la base de datos	27
6.	Diagrama del API respecto al módulo genérico de consultas a base de datos	32
7.	Secuencia de categorías.....	33
8.	Filtro de categorías a mostrar	34
9.	Flujo de la vista HTML de las categorías.....	34
10.	Pantalla principal de la aplicación	38
11.	Vista de las pestañas inferiores.....	38
12.	Vista de categorías de carreras.....	39
13.	Vista del panel de carrera	41
14.	Vista de la página de publicaciones	43
15.	Vista del módulo de reacciones.....	44
16.	Vista de la página de perfil	45
17.	Vista del módulo de nueva publicación	46
18.	Vista del segmento de archivos subidos	47
19.	Vista de la página de contactos.....	48
20.	Vista de la página de contactos.....	48
21.	Vista inicial del panel de administración	49
22.	Vista de los listados de categorías	50

23.	Vista de la edición de categorías	51
24.	Vista de la página de restablecimiento de contraseña	52
25.	Vista de correo de restablecimiento de contraseña	52
26.	Vista de la página de restablecimiento de contraseña	53
27.	Casos de uso del usuario final	56
28.	Casos de uso del usuario administrador	61
29.	Instalación de Android Studio	69
30.	Vista de la aplicación en Google Play	72

TABLAS

I.	Inscritos y graduados por década en la USAC	3
II.	CDU Registrar usuario	56
III.	CDU Iniciar sesión.....	57
IV.	CDU Cambiar datos personales	58
V.	CDU Realizar publicación.....	58
VI.	CDU Comentar publicación	59
VII.	CDU Reaccionar a publicación	60
VIII.	CDU Agregar categoría	61
IX.	CDU Editar categoría	62
X.	CDU Borrar categoría.....	63
XI.	CDU Listar categoría.....	64

GLOSARIO

<i>Hosting</i>	Servicio en línea que permite publicar un sitio o aplicación web en Internet.
No-SQL	Son sistemas de gestión de bases de datos que difieren del modelo clásico de sistemas de gestión de bases de datos en varios aspectos. El más importante es que no usan SQL como lenguaje principal de consultas.
Frontend	Es la parte de una aplicación que interactúa con los usuarios, conocida como el lado del cliente.
Backend	Es el interior de las aplicaciones, es decir, todo lo que vive en el servidor de datos.
NodeJS	Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript.
Angular 2	Es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página.

Base de datos	Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
Base de datos no relacional	Son sistemas de gestión de bases de datos que difieren del modelo clásico de sistemas de gestión de bases de datos en varios aspectos. El más importante es que no usan SQL como lenguaje principal de consultas.
MongoDB	Es un sistema de base de datos multiplataforma orientado a documentos, de esquema libre.
MongoDB Atlas	Es la base de datos como servicio que permite implementar, utilizar y escalar una base de datos de MongoDB con tan solo unos pocos clics.
Heroku	Es uno de los PaaS más utilizados en la actualidad en entornos empresariales, por su fuerte enfoque en resolver el despliegue de una aplicación.
PaaS	Proviene del nombre inglés <i>Plataform As A Service</i> . Es la encapsulación de una abstracción de un ambiente de desarrollo y el empaquetamiento de una serie de módulos.
Dominio	El propósito principal de los nombres de dominio en Internet y del sistema de nombres de dominio (DNS),

es traducir las direcciones IP de cada nodo activo en la red, a términos memorizables y fáciles de encontrar.

Enrutamiento

También llamado ruteo, es la función de buscar un camino entre todos los posibles en una red de paquetes cuyas topologías poseen una gran conectividad.

HTTP

De sus siglas en inglés, *Hypertext Transfer Protocol*, es el nombre de un protocolo que permite realizar una petición de datos y recursos, como documentos HTML.

Contenedor

Los contenedores ofrecen un mecanismo de empaquetado lógico en el que las aplicaciones se pueden abstraer del entorno en el que se ejecutan.

Git

Es un software de control de versiones diseñado por Linus Torvalds.

Dyno

Son los componentes básicos que impulsan cualquier aplicación de Heroku, desde lo simple hasta lo sofisticado.

Desarrollador

El desarrollador de aplicaciones móviles es un especialista encargado de diseñar la interfaz de usuario de cada aplicación que se le presente.

Computación en la nube

Es un paradigma que permite ofrecer servicios de computación a través de una red, que usualmente es internet.

Amazon EC2

Es una parte central de la plataforma de cómputo en la nube de la empresa Amazon.com denominada Amazon Web Services (AWS).

Amazon, Inc.

Es una compañía estadounidense de comercio electrónico y servicios de computación en la nube a todos los niveles, con sede en la ciudad estadounidense de Seattle.

RESUMEN

La aplicación “Yo entro a la U” servirá como un foro de ayuda a aspirantes que deseen ingresar a la universidad. Contará con una lista de carreras desde donde los usuarios podrán visualizar contenido perteneciente a cada carrera.

Cada carrera tendrá sus propias publicaciones, que a su vez estarán clasificadas de acuerdo con ciertas categorías para facilitar la búsqueda. Se creará la categoría material de estudio, que se usará para que los usuarios suban material perteneciente a la carrera especificada. También habrá la categoría experiencias de usuarios, que permitirá a usuarios que hayan cursado esa determinada carrera, compartir sus experiencias como estudiantes.

Habrará categorías de páginas oficiales y pensum que únicamente contarán con un hipervínculo, el cual servirá para redireccionar al usuario a los recursos especificados. Además, cada carrera contará con una sección de áreas de trabajo después de la graduación, por lo que los usuarios de la aplicación podrán ayudar a otros estudiantes a guiarse en la difícil decisión de escoger la carrera por estudiar.

Los usuarios de la aplicación también podrán compartir su información pública de contacto para que las personas interesadas en sus servicios las puedan contactar.

OBJETIVOS

General

Proveer una aplicación web para guiar a los nuevos estudiantes a conocer las experiencias de personas al entrar a la universidad y su situación, además de material necesario para estudiar e información curricular.

Específicos

1. Ofrecer el material curricular necesario para aplicar a los exámenes básicos y específicos por carrera: enlaces, documentos.
2. Mostrar información de contacto de personas que prestan servicios educativos.
3. Promover el intercambio de material educativo.
4. Publicar y compartir experiencias de las personas en la universidad.
5. Ayudar a las personas a tener un mejor conocimiento de la situación universitaria y motivarlas para que no desistan.
6. Publicar información sobre trabajos en los que se encuentran los estudiantes de las carreras.
7. Describir las carreras desde el punto de vista estudiantil.

INTRODUCCIÓN

La aplicación estará desplegada en distintas plataformas, para el API, que a su vez estará conectado a la base de datos en la plataforma Heroku. Para la base de datos se usará la plataforma MongoDB Atlas, que proporciona servidores de bases de datos gratuitos hasta cierto límite. La aplicación móvil estará escrita en Angular 10, se usará el *framework* de desarrollo denominado *Ionic*.

La aplicación contará con cinco pestañas inferiores que corresponde a las áreas más importantes de la aplicación. La primera, la pantalla principal, será un compilado de la información más relevante que tenga la aplicación hasta ese momento. La segunda pestaña contendrá una clasificación de carreras que ayudará a que la información se presente de una manera más ligera.

La tercera pestaña será un apartado de universidades donde se podrá explorar las oportunidades de carrera por universidad y así tener un mejor panorama de la oferta académica. La cuarta pestaña corresponderá al perfil de usuario, que contendrá información como el nombre de usuario y su correo electrónico. Una vez iniciada una sesión, la información del usuario se usará en toda la aplicación. La quinta y última pestaña será una lista de contactos que las personas podrán consultar si necesitan ayuda para su ingreso a la universidad.

1. MARCO TEÓRICO

1.1. El alcance de la educación superior en Guatemala

En Guatemala funcionan 13 universidades, una estatal y doce privadas. La oferta académica es de 1 086 carreras. La cantidad de estudiantes universitarios que existen por cada 100 000 habitantes del país es de 2 433,4, con una cobertura del 7,05 % para la población entre el rango de 15 a 34 años, o un 10,39 % para la población en el rango de 20 a 34 años (51,45 % mujeres y 48,55 % hombres). Para el año 2010 se matricularon 349 477 ciudadanos, 11,74 % en ciencias de la salud, 16,47 % en ciencias y tecnologías y 71,78 % en ciencias sociales.¹

En el año 2008 se graduaron 9 748 guatemaltecos, 67,38 % desde la universidad pública. El sistema universitario del país contribuye con 69 centros de investigación para beneficio de la población.

1.1.1. Cobertura geográfica

El sistema de universidades del país tiene presencia en todos los departamentos. La Universidad de San Carlos es la que más sedes tiene. Asimismo, cuenta con dos campus en ciudad de Guatemala, 18 centros en los departamentos y secciones departamentales de varias facultades.²

¹ TOBAR, Luis. *La educación superior en Guatemala en la primera década del siglo XXI*. <https://www.redalyc.org/pdf/1794/179422350009.pdf>.

² *Ibíd.*

La Universidad Galileo provee educación abierta bajo la modalidad a distancia, lo que le permite llegar a municipios muy alejados de los cascos urbanos.

También tienen una buena distribución la Universidad Rafael Landívar, Mariano Gálvez, Rural, Panamericana, que cuentan con sedes en el interior del país. El departamento en el que más universidades funcionan es el departamento de Guatemala, seguido del departamento de Quetzaltenango. Por su parte, la Universidad del Istmo y San Pablo solo tienen sedes dentro de la ciudad capital.

1.1.2. Carreras que se ofrecen en Guatemala

Cada una de las universidades autorizadas ofrece programas académicos que suman 320 carreras de pregrado, 409 de grado, 301 programas de maestrías, 31 especialidades y 25 doctorados, para un total de 1 086 programas formadores.

1.1.3. Matrícula estudiantil

Las inscripciones de estudiantes en educación superior en Guatemala han tenido un crecimiento de 210 % entre 1994 y 2010. Tanto el sector privado como el público han tenido tasas de crecimiento importante, aunque es el sector privado el que ha mostrado un aumento más significativo. La USAC sigue siendo la que más estudiantes posee, con un 42 % del total de personas matriculadas.

Según Alarcón y Luna, en 1999 en el sistema universitario nacional había inscritos 152 798 estudiantes. Sin embargo, el comportamiento de las personas graduadas es muy diferente al de las personas que ingresan a la universidad. El comportamiento histórico de los inscritos, graduados y tasas de crecimiento en

la Universidad de San Carlos se muestran en el cuadro No. 1, en donde resalta que el porcentaje de graduados sobre inscritos muestra una tendencia a crecer, como una evidencia de incremento en la eficiencia del sistema.

Tabla I. **Inscritos y graduados por década en la USAC**

Año	Inscritos	Graduados	Grad/ Inscrito %	% Crecimiento	
				Inscritos	Graduados
1970	11,722	466	3,98		
1980	35,421	1,694	4,78	202,18	263,52
1990	64,435	1,785	2,77	81,91	5,37
2000	104,186	2,781	2,67	61,69	55,80
2009	134,196	6,804	5,07	28,80	144,66

Fuente: TOBAR, Luis. *La educación superior en Guatemala en la primera década del siglo XXI*.
<https://www.redalyc.org/pdf/1794/179422350009.pdf>. Consulta: septiembre de 2020.

1.2. **Tecnologías para el desarrollo de aplicaciones**

Existen diversas tecnologías que se pueden utilizar para construir aplicaciones. La elección de estas se hace tomando en cuenta las funcionalidades que se requieren para el sistema. La aplicación “Yo entro a la U” empleará las tecnologías descritas a continuación.

1.2.1. **Plataforma de despliegue Heroku**

Heroku es un servicio de plataforma web como servicio (PaaS) que soporta varios lenguajes de programación. Permite desplegar y correr aplicaciones en diversas plataformas, entre ellas NodeJS, sobre la cual se ejecutará la aplicación.

1.2.2. Bases de datos NoSQL

El modelo de bases de datos no relacional es una clase de sistemas de gestión de base de datos cuya característica más destacada es que no usan solamente SQL como lenguaje principal de consultas. NoSQL es un derivado de “no solo SQL”. Estas bases de datos están construidas para proporcionar y soportar grandes cargas de lectura y escritura, por lo que pueden escalar muy fácilmente. La base de datos MongoDB pertenece a este modelo.

Las bases de datos construidas en MongoDB proporcionan un modelo basado en documentos que permite tener una gran flexibilidad para guardar los datos y ordenarlos de manera eficiente. Estas estructuras se guardan en una estructura de datos llamada BSON, que es muy similar al formato de texto.

1.2.3. Plataforma MongoDB Cloud

MongoDB Cloud es una plataforma de base de datos como servicio *Database-as-a-service (DBaaS)* que proporciona una forma accesible para trabajar con la información de aplicaciones de todo tipo.

Una base de datos como servicio permite a los desarrolladores de aplicaciones librarse de la responsabilidad de instalar y mantener las bases de datos ellos mismos. El proveedor es el encargado de mantener e instalar esta base y cobra cierto porcentaje de dinero por el uso que se le da a sus servicios. La base de datos como servicio también es una forma de variación del software como servicio (SaaS). El núcleo de MongoDB Cloud es MongoDB Atlas, una base de datos en la nube totalmente funcional usada para aplicaciones modernas. Las aplicaciones se pueden alojar en Amazon Web Services, Google Cloud y Azure.

1.2.4. Amazon Web Services

Es una compañía subsidiaria de Amazon, que provee computación en la nube, así como APIs a individuos, compañías y gobiernos, en una medida de pago por uso. MongoDB Atlas se alojará en esta plataforma para más facilidad y escalabilidad. MongoDB Atlas empleará el servicio Amazon Elastic Compute o EC2, que es una herramienta de computación en la nube que proporciona servidores de aplicación que son seguros y escalables.

1.2.5. Firebase Cloud Storage

Es un servicio proporcionado por *Firebase* de almacenamiento de objetos simple, potente y rentable que se ejecuta sobre los servicios de Google. Los servicios de Cloud Storage agregan la seguridad de Google a las operaciones de carga y descarga de los archivos de la aplicación basada en *Firebase*.

Los servidores de Google también proveen acceso a estos mismos archivos. Cloud Storage es escalable automáticamente, por lo que no será necesario migrar a otro proveedor de servicios si la aplicación crece demasiado. El servicio también permite filtrado de imágenes y archivos, así como reanudar una operación de subida de archivos si la conexión de red es deficiente.

1.2.6. Android Studio

Es un entorno integrado de desarrollo para la plataforma Android. Se utilizará para compilar la aplicación y hacerla correr en un dispositivo de pruebas. También se empleará para generar los archivos compilados de la aplicación, el APK y los certificados necesarios para su publicación en una tienda de aplicaciones.

1.2.7. Ionic Framework

Es una plataforma de desarrollo de software o *SDK* completa y de código abierto para el desarrollo de aplicaciones móviles híbridas. La versión original fue lanzada en 2013 y se construyó sobre Apache Cordova y AngularJS. El principal enfoque de este *framework* es ayudar a crear aplicaciones híbridas, es decir, que sean multiplataforma. También puede ser visualizado como una aplicación móvil que es en realidad una aplicación web corriendo en un navegador móvil, con acceso a la capa de aplicación nativa del dispositivo. Cuando se accede a la capa nativa, se hace posible utilizar todos los beneficios de una aplicación nativa.

2. IDENTIFICACIÓN DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

2.1. Antecedentes

Según algunos estudios, uno de cada tres estudiantes no termina la carrera que comienza y es peor aún en carreras que demandan más horas por día, como las ingenierías y las ciencias médicas. Los estudiantes, al encontrarse con una realidad que no conocen previamente, se decepcionan, pues sus expectativas no se ajustan a lo que realmente ofrecen las carreras.³

La educación superior es muy importante para el país, pues genera capacidades, eleva los ingresos monetarios de las personas, ayuda a mejorar los indicadores sociales, promueve la inclusión social, reduce la pobreza en general, garantiza un futuro próspero y contribuye a fortalecer la democracia.⁴

En Guatemala, según los datos mostrados en la tabla I (los cuales fueron obtenidos del Instituto Nacional de Estadística de Guatemala) para la década de los años 2000, el punto más alto de estudiantes graduados fue de aproximadamente solo un 5 % del total de matriculados. Si obviamos el aspecto económico, se puede entrever que muchas de estas personas desisten de sus carreras porque no supieron a qué se enfrentaban desde el principio y no pudieron asumir los retos de manera exitosa.

³ YAQ, Paula. *Los 7 errores más frecuentes a la hora de elegir carrera.* <https://yaq.es/reportajes/elegir-carrera-nos-equivocamos>.

⁴ TOBAR, Luis. *La educación superior en Guatemala en la primera década del siglo XXI.* <https://www.redalyc.org/pdf/1794/179422350009.pdf>.

2.2. Identificación del problema

Uno de los errores más comunes que enfrentan los estudiantes universitarios es que eligen la carrera sin tener suficiente información sobre el p nsum de estudio, los trabajos a los que se puede dedicar al momento de graduarse y el nivel de dificultad que se requiere para completarla.

Otro de los inconvenientes al momento de elegir la carrera adecuada es que no hay un panorama general de todas las carreras a las que se puede aplicar. Los aspirantes eligen las carreras que est n de moda o las que prefieren sus compa eros. Esto ocasiona que los estudiantes se den cuenta de que no les gusta la carrera hasta que ya se encuentran inscritos y con unos meses de la carrera ya cursados.⁵

Se puede enlistar algunos de los errores m s frecuentes a la hora de elegir una carrera universitaria.

- Elegir sin suficiente informaci n
- No valorar todas las alternativas posibles
- Estudiar por estudiar
- Decidir por lo que hacen o escogen los dem s
- Elegir lo f cil
- Guiarse s lo por el sueldo
- Hacer la decisi n con prisas, sin dedicarle el trabajo necesario.

⁵ YAQ, Paula. *Los 7 errores m s frecuentes a la hora de elegir carrera.* <https://yaq.es/reportajes/elegir-carrera-nos-equivocamos>.

2.2.1. Acciones útiles al momento de elegir una carrera universitaria

Existe una gran cantidad de actividades que se pueden realizar para escoger una carrera que se adapte a las necesidades del estudiante y que sea una decisión que persista en el tiempo. Algunas de ellas se describen a continuación.

2.2.1.1. Aprovechar la experiencia de otros

Conocer los aspectos más importantes desde el punto de vista de los profesionales graduados y estudiantes de los últimos años de una carrera puede ser una herramienta muy útil al momento de decidir la carrera a estudiar. Estas personas saben a lo que se enfrentaron y tienen muchas experiencias que pueden ser de utilidad a nuevos estudiantes. Hablar con personas experimentadas no solo puede ayudar a tener una visión más global de la carrera sino también a evitar los errores que estas personas tuvieron que experimentar en su propio proceso.

2.2.1.2. Comparar las carreras

Comparar carreras con información que se encuentre en un solo lugar puede ser un ejercicio muy útil para ver las ventajas y desventajas de cada carrera. Con esto se puede elegir la que mejor se adapte a las necesidades, habilidades y preferencias de cada individuo.

2.2.1.3. Investigar opciones desconocidas

Buscar carreras de las que no se sabe mucho puede ser otro ejercicio muy útil para definir lo que se desea estudiar. Informarse más a fondo de la gran cantidad de opciones ayuda a que se ejerza una decisión más acertada.

2.3. Propuesta de solución

La problemática expuesta con anterioridad crea la necesidad de proporcionar a los futuros estudiantes material con el que puedan informarse previamente sobre los aspectos más importantes de las carreras que tienen a su disposición, los retos y obstáculos con los que se deberán enfrentar.

Se propone crear una aplicación móvil que cuente con secciones de contenido que expongan los aspectos más importantes de las carreras universitarias que se ofrecen en el país. Será construida en la plataforma Ionic por la facilidad de uso de su *framework* de desarrollo Angular2.

Habrà una sección de material de estudio por carrera para que los aspirantes se puedan informar acerca de lo que deberían estudiar para entrar a la carrera elegida. También una sección con recursos útiles varios para informarse mejor sobre la situación de la carrera. La aplicación tendrá una sección de enlaces a páginas oficiales y a páginas de pensum. Se publicará únicamente los enlaces con el objetivo de mantener la información actualizada y estar al tanto de cambios en dichas páginas.

Por último, la aplicación tendrá las áreas de experiencias de egresados, consejos sobre la carrera y áreas de trabajo después de la graduación, para que

los aspirantes tengan una mejor vista de la situación a futuro y así tomar mejores decisiones sobre la carrera a elegir.

Para llegar al público en general, la aplicación se publicará en la tienda de aplicaciones de Google, a través de la cual cualquier persona con un celular que funcione con Android igual o mayor a Android 5.0 *Lollipop* podrá acceder y usar.

3. ARQUITECTURA DE LA APLICACIÓN

El proyecto se desplegará al público en herramientas de hosting gratuitas. La plataforma manejará comentarios y publicaciones, por lo que se utilizará una herramienta de base de datos No-SQL. La plataforma tendrá una dupla frontend-backend por lo que se podrán usar herramientas de última generación como NodeJS y Angular2.

3.1. Arquitectura de despliegue del backend

El backend de la aplicación estará alojado en la herramienta gratuita Heroku, la cual proporciona un despliegue basado en un sistema de versiones estables; se puede usar de forma simple y es gratuita siempre y cuando no se sobrepasen los límites mensuales de la plataforma gratuita. Sobre esta plataforma estará la aplicación basada en NodeJS que contendrá la conexión con la base de datos del sistema.

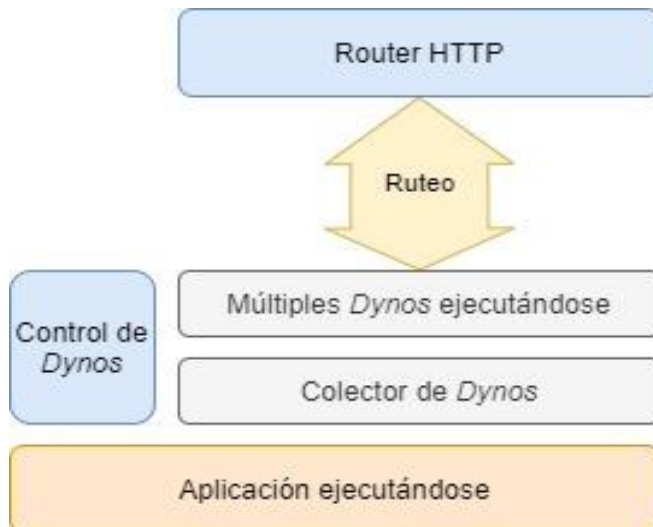
La aplicación deberá manejar una gran cantidad de información, por lo que se usará una base de datos no relacional, o No-SQL. En este caso se escogerá MongoDB por la versatilidad que proporciona y su facilidad de uso y despliegue. Esta base de datos estará desplegada en un servidor dedicado que será proporcionado por MongoDB Atlas, que a su vez correrá sobre la capa gratuita de Amazon Web Service, con su servicio Amazon EC2.

3.1.1. Funcionamiento de la plataforma Heroku

Las aplicaciones que se ejecutan en Heroku por lo general tienen un dominio único que se utiliza para enrutar las solicitudes HTTP al contenedor de aplicaciones correcto, también llamado *dyno*. En este caso, el dominio único para la aplicación será `https://yo-entro-a-la-u-backend.herokuapp.com/`. Cada uno de los *dynos* se distribuye en un *dyno grid* que consta de varios servidores.

El servidor git de Heroku maneja los cambios al repositorio de aplicaciones de los usuarios permitidos. Esto quiere decir que maneja las versiones de la aplicación que el desarrollador sube a la plataforma. Todos los servicios de Heroku están alojados en la plataforma de computación en la nube EC2 de Amazon.

Figura 1. Diagrama de funcionamiento de la plataforma Heroku



Fuente: elaboración propia.

El hacer uso de la arquitectura de Heroku evita inconvenientes de despliegue como el dominio HTTP y el mantenimiento de la aplicación de NodeJS dentro de un servidor dedicado.

3.1.2. Estructura del API de la aplicación

El API de la aplicación estará corriendo sobre el entorno de desarrollo NodeJS usando la librería Express JS. Esta proporciona las herramientas necesarias para el desarrollo como manejo de peticiones HTTP o HTTPS, enrutamiento, manejo de archivos, conexiones y seguridad de la aplicación.

3.1.3. Conexión entre plataformas

Para la conexión del servidor de MongoDB Atlas con los servidores de Heroku se utilizan variables de entorno que tienen a cargo permitir o bloquear la entrada de peticiones a ambos servidores. En el caso de MongoDB Atlas se emplea un usuario y contraseña que permiten tener seguridad y restringir el acceso no permitido. También se desbloqueó el *firewall* para que los diferentes usuarios de la aplicación pueden acceder a la aplicación desde cualquier dirección IP. Con esta política se logra que cualquier usuario con un dispositivo con acceso a internet pueda acceder a la información desde cualquier parte del mundo.

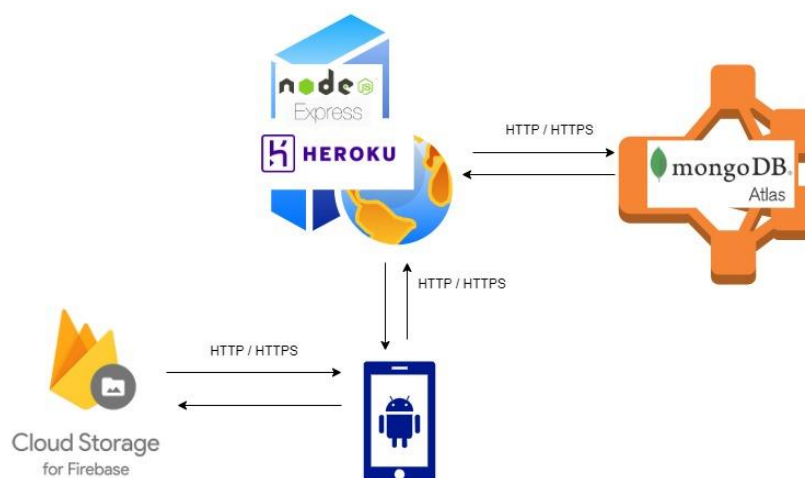
Para la plataforma Heroku hay variables de sesión que permiten el acceso a la consola de Heroku y variables de entorno que permiten definir hacia donde se guardará la información. En cuanto a la conexión con la base de datos, solamente es necesario definir la dirección IP del servidor donde se aloja MongoDB Atlas para que se pueda compartir información.

La comunicación entre las plataformas se realiza a través de peticiones HTTP que viajan de extremo a extremo a través de internet, conectando el dispositivo del usuario final con la aplicación de Heroku, la base de datos en MongoDB Atlas y el servicio de Google Cloud Storage.

La plataforma de Google Cloud Storage se accederá a través de las credenciales que proporciona Google. Este servicio se conectará tanto al servicio de backend proporcionado por Heroku como al servicio de frontend proporcionado por la aplicación móvil, el cual se definirá más adelante.

La aplicación móvil enviará una secuencia de archivo a través del protocolo HTTP hacia los servidores de Google Cloud Storage, que a su vez enviarán la metadata que contendrá la información para la descarga y manejo del archivo. Una vez recopilada la información del archivo esta se enviará en un formato específico hacia la base de datos proporcionada por MongoDB Atlas.

Figura 2. Diagrama de funcionamiento de la plataforma



Fuente: elaboración propia, empleando diagrams.net.

3.2. Arquitectura del frontend

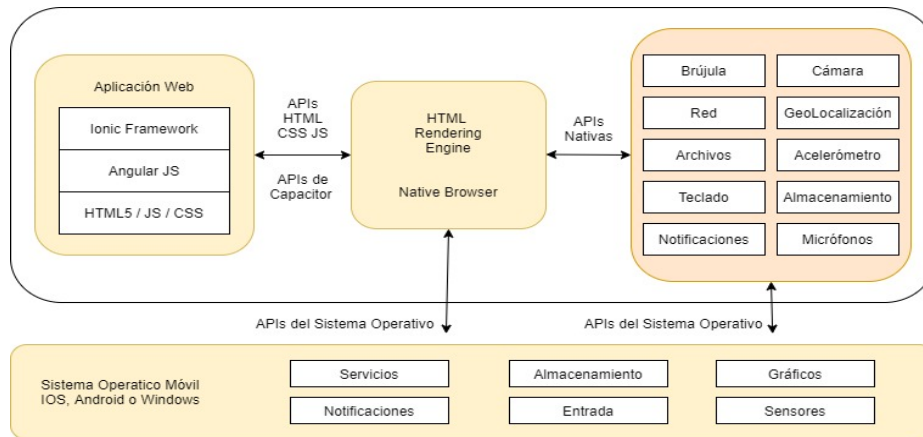
El frontend de la aplicación estará construido sobre la plataforma Ionic Framework usando Angular 9, otro framework basado en Javascript. Una vez compilado el código fuente, se generarán los archivos de Java necesarios para que la aplicación se pueda abrir y editar directamente con Android Studio, lo que permite el desarrollo de funciones nativas de la aplicación.

3.2.1. Ionic *Framework*

Las aplicaciones de Ionic se construyen encima de Capacitor, que se encarga de manejar los paquetes de la aplicación como si fuera una aplicación nativa que pueda correr en Android, IOS y la plataforma de Windows. Ionic ofrece librerías de HTML5, CSS y JS, componentes y herramientas para construir aplicaciones progresivas y de alto rendimiento. Las aplicaciones web convencionales presentan generalmente una experiencia de usuario lenta y si se trata de construir directamente como una aplicación híbrida, este problema todavía persiste. Ionic *Framework* provee una mejor experiencia de usuario al convertir las aplicaciones a código nativo.

Ionic está construido con SASS, que corresponde a la estructura de los estilos y proveerá al desarrollador el poder de configurar su aplicación usando los componentes nativos de Ionic.

Figura 3. Diagrama de arquitectura de desarrollo híbrida



Fuente: elaboración propia, basada en: <https://techsnapshot.files.wordpress.com/2016/08/wepapparchitecture.png?w=1200>. Consulta: octubre de 2020.

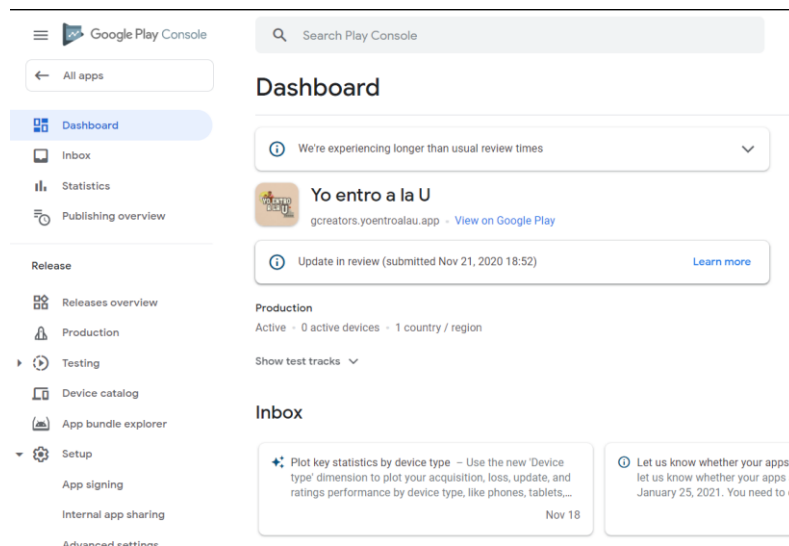
3.3. Alojamiento en la tienda de aplicaciones

La aplicación se publicará en Google Play Store, la cual es una tienda de aplicaciones para Android. El proceso de publicación consiste en una serie de pasos donde se especifica el tipo de aplicación que se está ingresando, la edad de los usuarios a los que va dirigida, el tipo de contenido, el país, entre otras cosas. La aplicación publicada contará con un logo, el cual se puede apreciar en el apéndice 1. Este logo aparecerá en el teléfono móvil como ícono de la aplicación.

Todo el proceso de publicación de la aplicación se realiza con el panel denominado *Google Play Console*, el cual es un módulo donde se puede administrar la aplicación publicada a través de una cuenta de desarrollador de Google. Una cuenta de desarrollador de Google permite publicar cualquier tipo de aplicaciones, ya sea gratuitas o de pago, con un pago inicial único de

25 dólares estadounidenses. El logotipo de la cuenta de desarrollador se puede apreciar en el apéndice 2.

Figura 4. **Panel de administración de Google Play Console**



Fuente: elaboración propia.

4. DESARROLLO DE LA APLICACIÓN

4.1. Estructuración de la base de datos

La base de datos estará implementada en MongoDB, por lo que en lugar de estructuras convencionales de SQL se guardarán estructuras de datos BSON (que son similares a una estructura JSON). Para la construcción de la base de datos se tendrá un enfoque simplista en el que se comparten propiedades entre tablas, de manera que haya la menor cantidad posible de las mismas.

4.1.1. Esquemas de la base de datos

Un esquema determina la disposición de los elementos de una base de datos y sus relaciones. Este esquema se describe a continuación.

4.1.1.1. Esquema de usuarios

Para empezar, se tendrá un esquema de usuarios, el cual contendrá información del usuario que está utilizando el sistema. Los campos serán los siguientes: nombre de usuario, email, contraseña, nombre completo, profesión, dirección, universidad, imagen. También se tendrá la propiedad rol, la cual conectará con el esquema de roles.

4.1.1.2. Esquema de roles

Habrán un esquema de roles que definirán y limitarán las acciones que puede realizar un usuario del sistema. Los campos de este esquema son: título,

descripción, tipo del rol. Este último es un número único que tendrá cada uno de los roles. Los números de rol serán los siguientes:

- Número 1: administrador del sistema
- Número 2: usuario final del sistema

4.1.1.3. Esquema de publicaciones

El esquema de publicaciones servirá para mostrar y guardar todas las publicaciones que hagan tanto los usuarios finales como los administradores de la aplicación. Las publicaciones contarán con los siguientes campos:

- Usuario: funcionará como referencia al usuario que hizo la publicación.
- Categoría de la publicación: hace referencia al esquema “categorías” y servirán para dividir las publicaciones en ciertas áreas como la carrera a la que pertenecen, la universidad, entre otras.
- Clasificación de publicaciones: servirá para clasificar las publicaciones de acuerdo con su tipo. Serán las siguientes:
 - Material de estudio
 - Páginas oficiales
 - Pénsum de estudios
 - Áreas de trabajo de graduandos
 - Experiencias de estudiantes y graduandos
 - Recursos útiles
 - Consejos sobre la carrera

- Fecha de creación
- Título
- Contenido de la publicación: se guardará en formato HTML para que se puedan visualizar estilos como tamaño de letra y fuente.
- Archivos: esta propiedad será un arreglo del esquema “archivos” que servirá para guardar la información de los archivos subidos al sistema. Cada publicación puede tener varios.
- Reacciones: este campo será una referencia al esquema “reacciones” para poder contabilizar las reacciones que haya.
- Comentario hijo: este campo será una referencia cíclica al mismo esquema “publicaciones”. Se hace de esta manera para poder copiar las mismas propiedades al comentario y simplificar el código fuente de la aplicación.
- Número de anclaje: este campo será de tipo numérico y servirá para anclar comentarios en lo más alto de una lista de publicaciones. Servirá para resaltar algunas publicaciones y que siempre aparezcan al principio. Un número más alto se traduce en una mayor precedencia, mientras que un número menor hará que las publicaciones aparezcan más abajo.

4.1.1.4. Esquema de reacciones de publicación

El esquema de reacciones de publicación servirá para agregar un símbolo de reacción a cada una de las publicaciones de la plataforma. La reacción estará asociada a una publicación y a un usuario. Se tendrá como tercer campo el tipo

de reacción, que corresponderá a un número del 1 al 6. Cada tipo de reacción tendrá su número definido.

4.1.1.5. Esquema de archivos

Este se usará para guardar la información de los archivos subidos a la plataforma de Google Cloud Storage Service. Contará con los siguientes campos:

- **Publicación asociada:** funcionará como referencia a la publicación a la que pertenece el archivo. Se puede decir que es la publicación padre, al estar un nivel arriba de este esquema.
- **Categoría:** esta servirá como referencia para hacer un barrido de información que recopile los archivos en grupos y los muestre de determinadas maneras.
- **Nombre del archivo:** es el nombre que tendrá el archivo al momento de guardarse en la base de datos.
- **Tipo de archivo:** es la extensión del archivo subido. Algunos ejemplos de los archivos que pueden subirse a la plataforma son: JPG, PDF, PNG.
- **Identificador del archivo de *firebase*:** es una referencia a la ruta del archivo que se subirá, descargará o borrará. Por ejemplo, “imagen.png”.
- **Folder del archivo de *firebase*:** este campo indicará la carpeta en la que estará guardado el archivo en el servidor de *firebase*.

- Enlace de descarga del archivo: este campo guardará una dirección URL que cuenta con un token de acceso al archivo, que permitirá descargarlo directamente desde cualquier dispositivo conectado a internet.

4.1.1.6. Esquema de categorías

El esquema de categorías se empleará para clasificar y categorizar cada una de las publicaciones que se hagan dentro de la plataforma. Las categorías contarán con una referencia cíclica que funcionará como padre de las mismas. Esto permitirá una navegación más ordenada dentro de las distintas categorías con las que cuenta el sistema. Aparte de las categorías padres, en este esquema se guardará un arreglo con todas las categorías de tipo hijo que tenga el esquema. El esquema de categorías contará con los siguientes campos:

- Categoría padre: este campo servirá para tener niveles dentro de la aplicación y que se pueda navegar dentro de ellos.
- Categorías hijas: este campo será un arreglo que hará referencia a todas las categorías hijas que tenga una categoría específica. Se forman tomando todas las categorías que tengan padres comunes.
- Palabra clave de categoría: este campo servirá para clasificar las categorías para que se presenten de diferente manera. Por ejemplo, cuando se seleccione una categoría cuya palabra clave sea “carrera” se presentará una imagen superior que corresponda a una carrera universitaria, en otras situaciones esto no será así.
- Imagen del avatar: en algunas categorías se tendrá una imagen de avatar que será muy similar a una imagen de perfil de usuario.

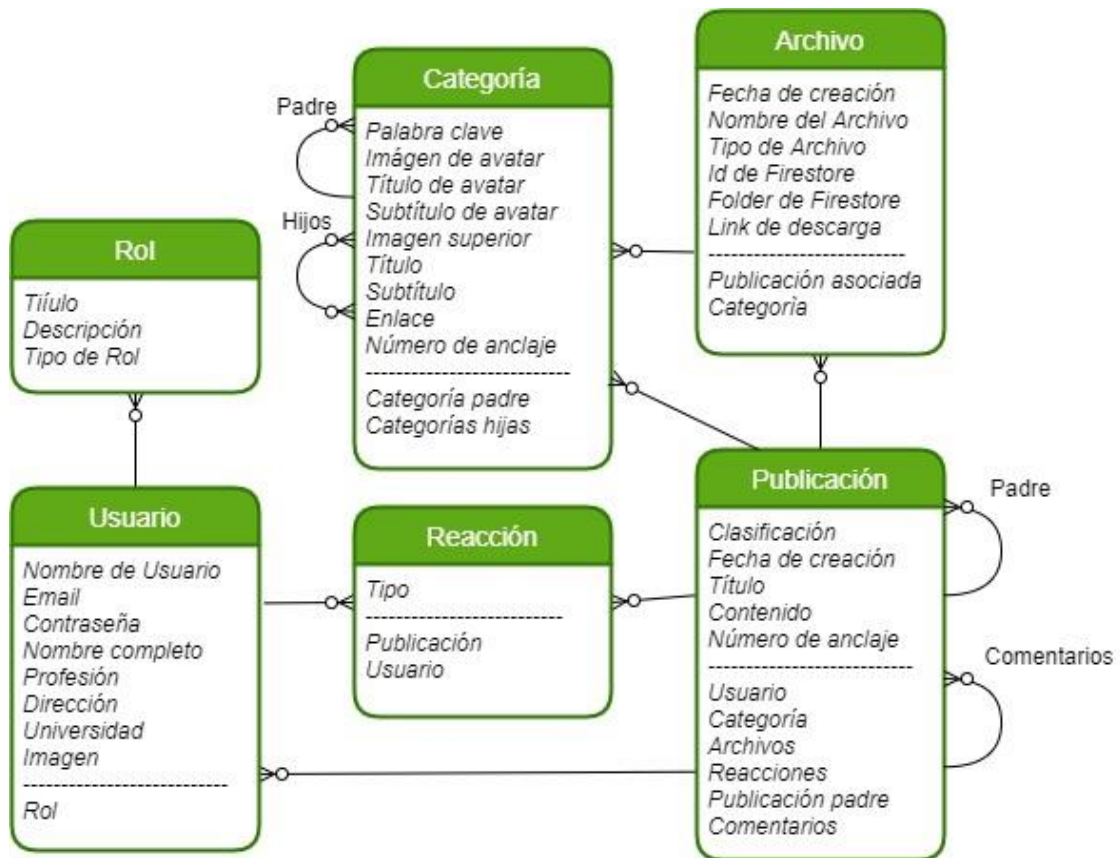
- Título del avatar: en determinadas situaciones se mostrará junto a la imagen del avatar.
- Subtítulo del avatar: al igual que el título del avatar, este texto se mostrará junto a la imagen del avatar.
- Imagen superior: en algunas categorías funcionará como la carta de presentación de la categoría.
- Título de la categoría: servirá para indicar el título de las categorías.
- Subtítulo de la categoría: servirá para indicar alguna información secundaria de las categorías.
- Enlace: en algunas categorías será un hipervínculo a otros recursos.
- Número de anclaje: servirá para priorizar algunas categorías sobre otras.

4.1.2. Diagrama de la base de datos

Las relaciones entre los esquemas de la base de datos se pueden describir en un diagrama. Tomando en cuenta que se trata de una base de datos no relacional, o NoSQL, muchas de estas relaciones y campos no estarán del todo presentes, ya que hablamos de un sistema y estructura muy versátiles. En el diagrama, cada cuadro redondeado representa un esquema; en la parte superior se encuentra el título y en la inferior el contenido. Dentro del contenido, arriba de la línea punteada se indican las propiedades individuales (las que solo existen dentro de su propio esquema) y debajo de la línea punteada, las propiedades compartidas que se usarán para conectar los diferentes esquemas. Por ejemplo,

la propiedad *rol* del usuario, que será la encargada de clasificar al usuario en un rol específico para que pueda realizar sus acciones asignadas.

Figura 5. Diagrama de la base de datos



Fuente: elaboración propia, empleando diagrams.net.

4.2. Estructuración del API

Un API es un conjunto de definiciones y protocolos para construir e integrar aplicaciones de software. Las siglas API vienen del inglés *Application Programming Interface*. El API deja comunicar a los servicios sin tener que preocuparse por cómo están contruidos cada uno de ellos.

El código del API estará escrito en Javascript usando el entorno NodeJS, el cual permitirá emplear un tipo de programación multiparadigma que tendrá una gran versatilidad para trabajar la aplicación. El proyecto contará con diversas librerías que agregarán funcionalidad al código sin necesidad de volverlo a programar. Para el manejo de la base de datos, se usará una librería denominada *mongoose*, la cual se encargará de mantener viva la conexión al servidor de base de datos, mapear los esquemas a *Javascript*, entre otros.

El API de la aplicación “Yo entro a la U” tiene un enfoque minimalista que procura que el software tenga la menor cantidad posible de clases y que se reutilice el código escrito tanto como sea posible. Tendrá una estructura de módulos, los más importantes se describen a continuación.

4.2.1. Módulo selector de esquemas

Es el encargado de conectar cada uno de los esquemas de la base de datos con los controladores de datos, de manera que se pueda crear, editar, enlistar y borrar cada uno de los esquemas. La petición HTTP hacia cualquier acción de un esquema pasará primero por este módulo selector de esquema, por lo que será un transporte intermedio entre el consumidor y el controlador de la base de datos. La petición HTTP deberá tener en su ruta, una referencia al nombre del esquema que se quiere utilizar.

4.2.2. Módulo genérico para consultas de base de datos

Este módulo contará con los métodos necesarios para crear, eliminar, editar y enlistar la información. Al momento de ejecutar el módulo, este ya sabrá el esquema de base de datos al que tendrá que apuntar para extraer la información,

pues esta tarea ya fue realizada por el “Módulo selector de esquemas”. Cada uno de los métodos de este módulo se describen a continuación.

4.2.2.1. Método para obtener un solo registro

Este método se encargará de retornar un registro de base de datos haciendo uso del campo `_id` del esquema de MongoDB que fue proporcionado en el cuerpo de la petición HTTP. Retornará todos los campos existentes en la tabla para este único registro.

4.2.2.2. Método para obtener todos los registros

Este método se encargará de retornar todos los registros de la base de datos pertenecientes a un esquema específico. Retornará todos los campos existentes para el esquema seleccionado, sin filtrarlos. El cuerpo de la petición HTTP solo contará con el nombre del esquema a consultar.

4.2.2.3. Método para obtener todos los registros por filtro

Este método se encargará de retornar todos los registros de la base de datos pertenecientes a un esquema específico, que cumplan con el filtro especificado. Este retorno de datos se denominará conjunto de registros. El cuerpo de la petición HTTP contará con ciertos parámetros que se describen a continuación.

- Parámetro *find*: este parámetro utilizará dos parámetros internos para retornar la información de cierta manera, denominados query y projection. El parámetro query contará con una lista de pares campo-valor; campo es

el nombre de un campo del esquema de base de datos especificado y valor, el valor del campo que coincidirá con un conjunto de registros. Si el valor no coincide con ningún registro, se retornará un conjunto vacío. El parámetro *projection* contará también con una lista de pares campo-valor, donde campo es el nombre de un campo del esquema de base de datos especificado y valor, un valor que fluctuará entre dos valores específicos, 1 o 0, o en su forma booleana, verdadero o falso. El valor 1 o verdadero servirá para indicar que el conjunto de registros a retornar debe contar con el campo especificado y el 0 o falso indicará que en el conjunto registros no se debe incluir ese campo del esquema.

- Parámetro *sort*: este se utilizará para ordenar los registros obtenidos de acuerdo con los campos especificados. El parámetro contará con una lista de pares campo-valor, siendo campo el nombre de un campo del esquema de base de datos especificado y valor un número que puede ser -1 o 1, que sirve para indicar un orden descendente o ascendente, respectivamente, para cada campo del conjunto de registros retornado.
- Parámetro *populate*: el método *populate* reemplaza automáticamente los campos que estén ligados a otro esquema con el contenido de este mismo esquema de manera automática; el parámetro *populate* se utilizará para esta misma tarea. El parámetro contará con una lista de campos, donde cada campo el nombre de un campo del esquema de base de datos especificado que permitirá que el conjunto de registros retornado posea también la información de este otro esquema.
- Parámetro *limit*: será de tipo numérico e indicará el número de registros que se retornaran para la petición solicitada.

- Parámetro *skip*: será de tipo numérico e indicará el número de registros que se tienen que saltar antes de empezar a obtener los registros de la base de datos. De esta manera, el servidor solo retornará los registros a partir del número especificado, haciendo que se pueda delimitar aún más el conjunto de registros retornado.

4.2.2.4. Método para insertar un nuevo registro

Este método se encargará de insertar un registro de la base de datos. Los campos que no se especifiquen no se guardarán en la base de datos. Algunos campos de los esquemas son requeridos, por lo que este método deberá respetar esas reglas para que se puedan insertar dichos registros.

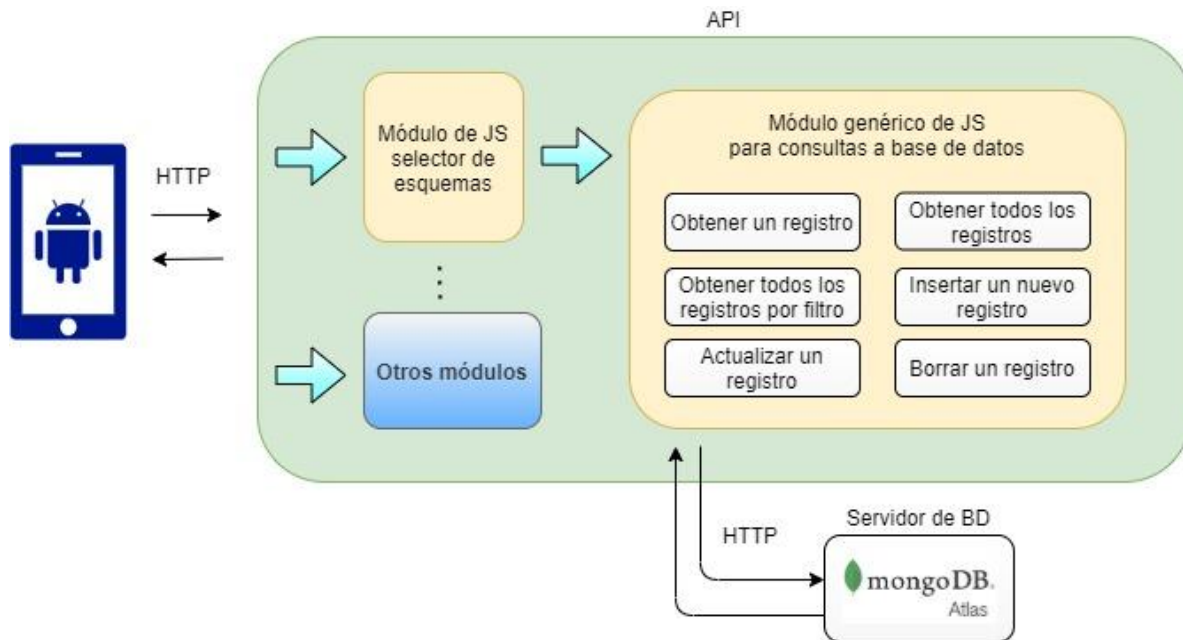
4.2.2.5. Método para actualizar un registro

Este método se encargará de actualizar los campos especificados de un registro de la base de datos. La petición HTTP hacia este método contará con dos propiedades. La primera será *query* o filtro que ayudará a encontrar el registro que se quiere actualizar. Por ejemplo, en este *query* se puede incluir el campo *_id* del registro. La segunda será la propiedad *data*, la cual contendrá la información que se actualizará en el registro.

4.2.2.6. Método para borrar un registro

Este método se encargará de eliminar un registro de base de datos haciendo uso del campo *_id* del esquema de MongoDB que fue proporcionado en el cuerpo de la petición HTTP. El sistema buscará el elemento a borrar usando este identificador único; de esta manera se elimina solo un registro específico.

Figura 6. **Diagrama del API respecto al módulo genérico de consultas a base de datos**



Fuente: elaboración propia, empleando diagrams.net.

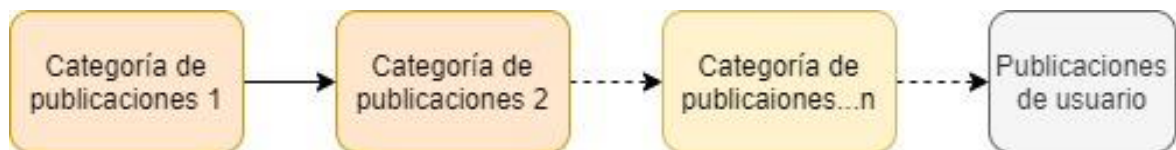
4.3. Estructuración de la aplicación de Ionic con Angular 10

La aplicación será escrita en Angular 10 y correrá sobre la plataforma Ionic Framework, por lo que la aplicación podrá utilizar todas las características que ofrece esta herramienta, como componentes y servicios. Esta parte de la aplicación también tomará un enfoque minimalista, por lo que se procurará reusar los componentes de la aplicación que compartan características y así minimizar la cantidad de código que se escribe.

4.3.1. Reciclaje de la vista de categorías

La vista de categorías es una característica de la aplicación que funciona de manera que un conjunto inicial de categorías redirige o apunta a otro conjunto diferente de categorías, lo que hace que se forme una especie de secuencia de pasos hasta llegar a una categoría que finalmente redirige a un conjunto específico de publicaciones. Esta secuencia se describe de mejor manera en la figura número 6.

Figura 7. **Secuencia de categorías**



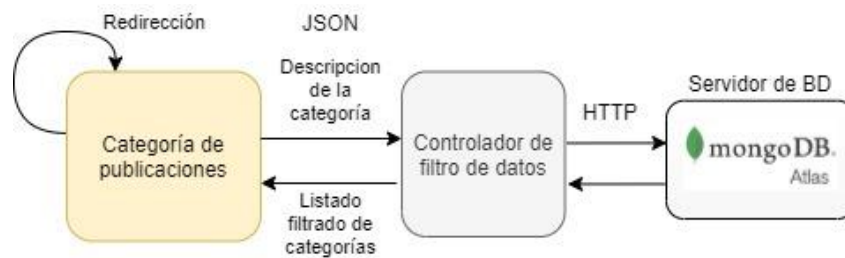
Fuente: elaboración propia, empleando diagrams.net.

Al programar cada una de las categorías por separado se incurriría en una mayor repetición de código que de otra forma se podría optimizar. Siguiendo el planteamiento de la minimización del código, el enfoque elegido para esta característica será usar una misma lógica de *Javascript* y una misma plantilla HTML para cada una de las categorías de la aplicación.

Para lograr el cometido de la reutilización de código se empleará dos estrategias: la primera consistirá en limitar la información que se consulta del API de la aplicación; esto quiere decir que por cada categoría se tendrá un conjunto de registros de base de datos diferenciado por una palabra clave y que se podrán filtrar desde la aplicación de usuario haciendo uso de una consulta o *query* de MongoDB diferente por cada categoría. Para simplificar la explicación se puede

decir que al seleccionar alguna categoría, el sistema deberá redirigir a la misma vista, pero con diferentes datos.

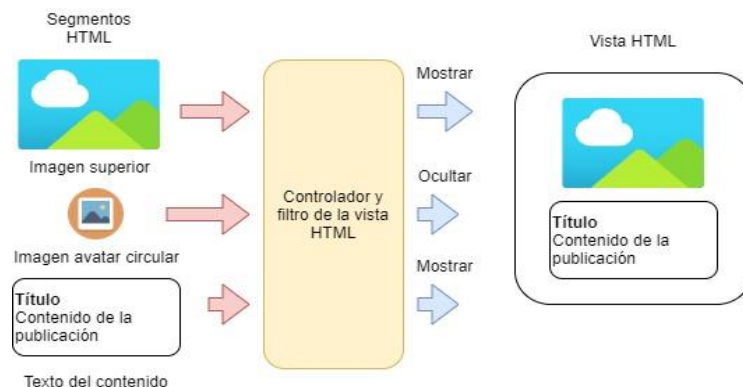
Figura 8. Filtro de categorías a mostrar



Fuente: elaboración propia, empleando diagrams.net.

La segunda estrategia consistirá en tener definido lo que se muestra y lo que no se muestra, de acuerdo con tipo de categoría que se le esté presentando al usuario. Esto quiere decir que cada segmento interno de la plantilla tendrá una condicional que hará que cada vista parezca diferente pero internamente se seguirá utilizando la misma lógica.

Figura 9. Flujo de la vista HTML de las categorías



Fuente: elaboración propia, empleando diagrams.net.

El controlador de usuarios tendrá una estructura diferente, ya que este necesita que las contraseñas se guarden de manera encriptada para proporcionar la seguridad necesaria que debería tener una contraseña. Para lograr el reto de la seguridad, a las contraseñas se les aplicará un algoritmo MD5, el cual producirá un valor de tipo *hash* de 128 bits, el cual, a su vez, se guardará de esta forma en la base de datos.

Para los controladores restantes de la aplicación se usará una estrategia parecida a la descrita anteriormente, tratando de redireccionarse a sí mismos con datos diferentes y utilizando condicionales para cada segmento de la vista HTML del usuario.

5. PROTOTIPO DE LA APLICACIÓN

La aplicación contará con una pantalla principal donde se tratará de mostrar la información más relevante que posea la aplicación. Contará con un conjunto de pestañas inferiores que proveerán la navegación del sistema. Cada publicación estará categorizada en uno o más niveles. Por cada publicación se tendrán reacciones y comentarios con los que se podrá establecer lo que piensan los usuarios finales de estas publicaciones.

5.1. Página principal

La pantalla principal contará con una imagen de fondo principal y encima de ella, una barra de búsqueda donde se podrán buscar cada una de las carreras disponibles en el sistema. Además, esta pantalla estará dividida en pequeños segmentos con forma de tiras horizontales que albergarán una serie de cartillas con un contenido de tipo específico.

El primer segmento mostrará las categorías de carreras que se encuentran disponibles en el sistema; cada una albergará un conjunto de carreras de características similares. Usualmente las categorías coincidirán con las escuelas de la Universidad de San Carlos de Guatemala.

El segundo segmento mostrará las publicaciones de tipo material de estudio más populares que se hayan realizado dentro de la plataforma. Se tomará como prioridad las publicaciones hechas por el administrador.

El tercer segmento mostrará las publicaciones de tipo experiencia de egresado más populares que se hayan realizado dentro de la plataforma.

Figura 10. **Pantalla principal de la aplicación**

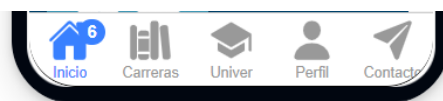


Fuente: elaboración propia.

5.2. Pestañas inferiores

Las pestañas inferiores tendrán como función principal redirigir hacia las distintas funcionalidades de la aplicación. Al momento de seleccionar alguna de ellas la pantalla que se muestra arriba cambiará su contenido.

Figura 11. **Vista de las pestañas inferiores**



Fuente: elaboración propia.

5.3. Página de categoría de carreras

Esta pantalla servirá para categorizar las carreras y que se presenten de una manera más ordenada. Se tendrá una barra de búsqueda para filtrar los resultados que es estén presentando en ese momento. Cada elemento de categoría contará con título de la categoría, descripción de categoría y una imagen descriptiva. Al seleccionar un nivel superior se redirigirá a un nivel inferior que presentará una lista diferente, hasta llegar al panel contenido de cada carrera.

Figura 12. Vista de categorías de carreras



Fuente: elaboración propia.

5.4. Página del panel de carrera

Esta pantalla servirá para mostrar toda la información publicada sobre una carrera específica. Contará con subsecciones, cada una con un conjunto filtrado de publicaciones. Las subsecciones de este panel se describen a continuación.

- **Material de estudio para ingreso:** esta sección contendrá publicaciones referentes a material de estudio para ayudar con los exámenes de ingreso a la carrera seleccionada. También se podrá subir archivos para que se puedan descargar directamente dentro de la aplicación.
- **Páginas oficiales:** tendrá publicaciones con hipervínculos hacia las páginas más relevantes de la carrera. Ya sea su página web oficial, las páginas oficiales de la escuela a la que pertenece la carrera específica o sus páginas de Facebook y Twitter.
- **Péñsum de estudios:** tendrá una publicación única que será un hipervínculo a la página oficial del péñsum de estudio de la carrera seleccionada. Con esto se mantendrá actualizada la información, aunque cambie la estructura curricular de la carrera.
- **Áreas de trabajo:** contendrá publicaciones sobre los tipos de trabajos en los que ejercen las personas egresadas de las carreras. Los usuarios podrán publicar una lista de posibles áreas de empleo de la carrera y explicar cada una de ellas. Esto para ayudar al usuario a encontrar nuevos horizontes de la carrera.
- **Experiencias de egresados:** servirá para que los usuarios que están estudiando la carrera o ya están graduados, publiquen sus experiencias

durante su estancia en la carrera. Con esto se pretende a ayudar a los estudiantes para que no se desanimen y que vean que hay muchas personas que comparten su situación.

- Recursos útiles: esta sección se utilizará para compartir recursos útiles sobre la carrera, ya se material extracurricular, reseñas desde otras páginas, documentos y en general recursos de internet relacionados a la carrera seleccionada por el usuario.
- Consejos sobre la carrera: esta área contará con consejos de usuarios que estén llevando la carrera o ya la hayan culminado.

Figura 13. Vista del panel de carrera



Fuente: elaboración propia.

5.5. Página de publicaciones

Esta página servirá para mostrar cada una de las publicaciones que los usuarios realicen en la sección. Internamente solo se consultarán 10 registros a la vez, para no sobrecargar tanto el servidor como la aplicación móvil; en otras palabras, se implementará un tipo de paginación en la vista.

En la parte superior bajo el título se tendrá un *dropdown* que tendrá dos opciones de ordenamiento de las publicaciones. La primera, denominada “Más popular”, se usará para mostrar las publicaciones con más reacciones de la sección. La segunda opción, denominada “Más reciente”, se usará para mostrar las publicaciones con fecha más reciente de manera descendente.

Cada publicación se mostrará en una cartilla separada con varias secciones diferenciadas. En la parte superior se mostrará el autor de la publicación, la foto de perfil del autor y la fecha de publicación. El siguiente segmento de la cartilla estará conformado por el título de la publicación y más abajo su contenido. El contenido internamente tendrá un formato HTML, por lo que se podrá jugar con los tamaños de la letra, se podrá poner el texto en negrilla o cursiva, establecer links, entre otros.

Se tendrá también una pequeña área de reacciones que será un compilado de las reacciones realizadas por los usuarios, de manera que la más popular aparezca más a la izquierda y la menos popular, más a la derecha. Se tendrá un botón de tipo ícono para abrir la pantalla de comentarios y otro botón con la leyenda “Ver comentarios”, que tendrá la misma función. La pantalla de comentarios contendrá todos los comentarios hechos por los usuarios para esa publicación.

En la parte inferior de la publicación se tendrá primero un botón con un ícono de “me gusta” el cual, al seleccionarlo, abrirá un pequeño módulo con más botones que se denominará “Módulo de reacciones”. Posteriormente, cuando se haga click a alguno de estos botones, la interacción se guardará en la publicación seleccionada. Cada usuario podrá realizar una sola interacción por publicación. Junto al botón “me gusta” se tendrá una cajetilla de comentarios donde se podrá escribir un comentario sobre la publicación, que luego se podrá publicar haciendo click en el botón “Enviar”.

Figura 14. Vista de la página de publicaciones

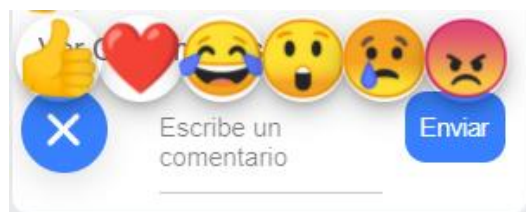


Fuente: elaboración propia.

5.6. Módulo de reacciones

Este módulo servirá para mostrar y contabilizar las interacciones que realice un usuario en una publicación. El usuario final podrá escoger una de las 6 opciones que tiene disponibles. Cada usuario solo podrá reaccionar una vez a cada publicación, por lo que cada reacción internamente tendrá guardada la información de la persona que realizó dicha interacción en la publicación.

Figura 15. Vista del módulo de reacciones



Fuente: elaboración propia.

5.7. Página de perfil

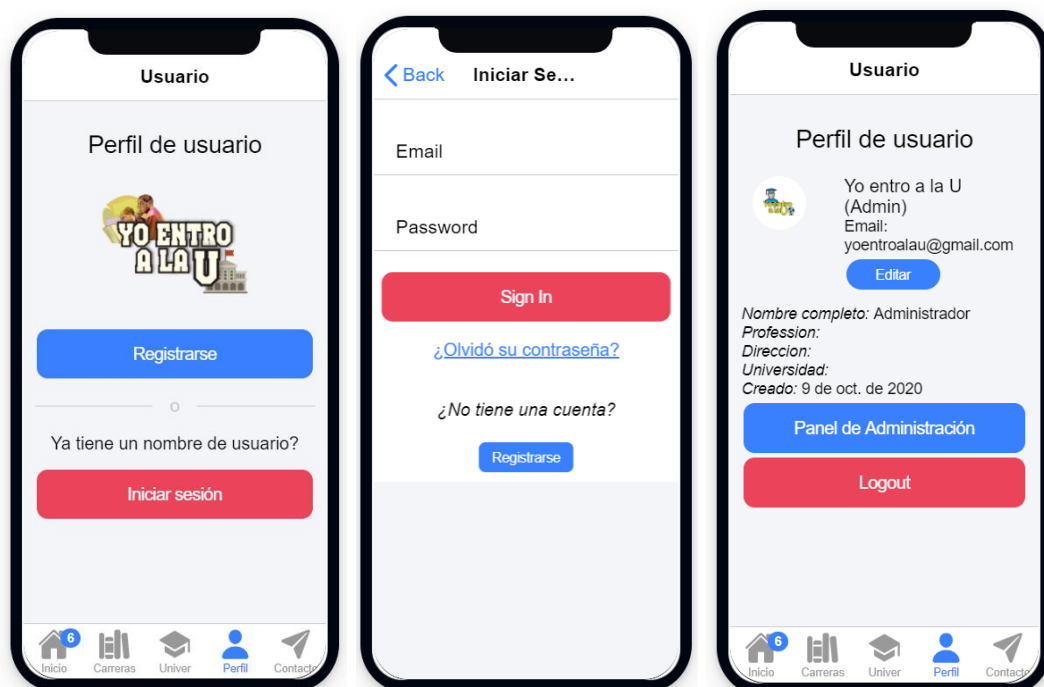
Esta pantalla mostrará la información de la persona que haya iniciado sesión en el sistema. Para iniciar sesión se deberá ingresar el usuario y contraseña utilizados en el momento del registro. Una vez iniciada la sesión, los usuarios podrán cambiar sus datos personales como el nombre de usuario y su profesión. También podrán comentar en las publicaciones y hacer publicaciones que quedarán guardadas con su nombre de usuario.

En esta pantalla también existirá una página de inicio de sesión que servirá para entrar al perfil, así como también una de registro de nuevos usuarios. En la

página de registro se podrá registrar una nueva cuenta para poder realizar las acciones explicadas con anterioridad.

Si se inicia sesión con la cuenta de administrador, en la página de perfil aparecerá un botón denominado “Panel de Administración” con el cual se podrá acceder a un panel administrativo para poder agregar carreras, categorías, entre otros.

Figura 16. Vista de la página de perfil



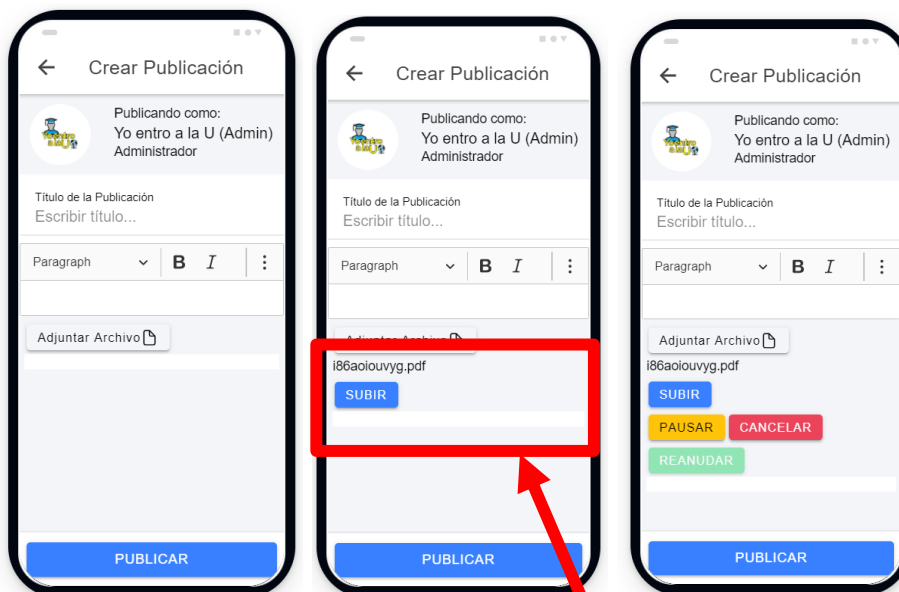
Fuente: elaboración propia.

5.8. Página para nuevas publicaciones

La pantalla para nuevas publicaciones consistirá en un primer *input* para colocar el título de la publicación y un segundo *input* para publicar el contenido. La cajilla de contenido está basada en una librería de *JavaScript* que permite darle ciertos tipos de formatos ya definidos a cada una de las palabras que se están escribiendo en esta entrada de texto.

Por debajo de estas dos entradas de texto se encontrará un botón denominado “Adjuntar Archivo”, que tendrá como finalidad guardar en una memoria de tipo temporal los archivos que se suban a la publicación. Cuando se seleccione un archivo para subirlo aparecerá un botón con la etiqueta “Subir”, el cual, al momento de pulsarlo, empezará el proceso de subida del archivo al servidor y desplegará una serie de botones que permitirán pausar, reanudar o cancelar dicho proceso.

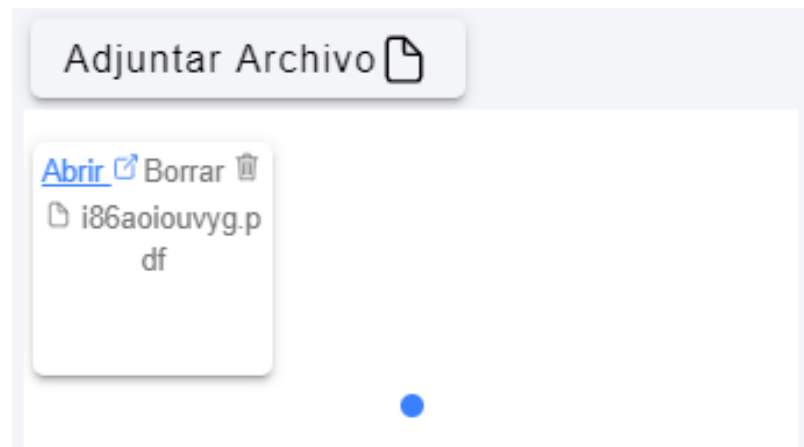
Figura 17. Vista del módulo de nueva publicación



Fuente: elaboración propia.

Los archivos subidos localmente aparecerán por debajo del botón “Adjuntar Archivo”. Estos archivos no se asociarán a ninguna publicación hasta que se presione el botón “Publicar” que se encuentra al final de la pantalla. Cada archivo tendrá un botón de borrar, el cual permite remover el archivo de la publicación que se está escribiendo actualmente.

Figura 18. **Vista del segmento de archivos subidos**

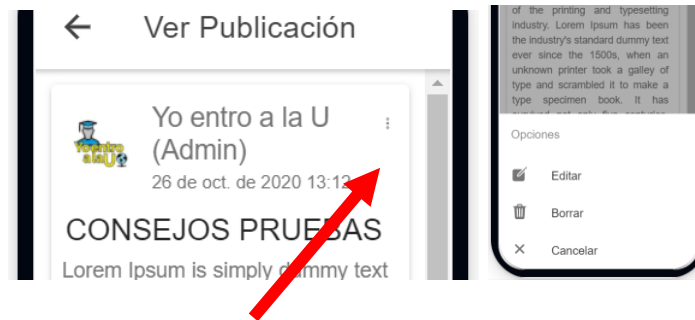


Fuente: elaboración propia.

5.9. **Página para editar la publicación**

A esta pantalla se accederá presionando el botón de tres puntos ubicado en la parte superior derecha. Esta acción abrirá un menú inferior en el cual habrá dos opciones: la primera, para editar la publicación actual; la segunda servirá para eliminar definitivamente la publicación. Para la parte de edición de la publicación se usará la misma vista de nueva publicación, pero llenando los datos con la información del post actual. El botón de eliminación abrirá una pequeña ventana para confirmar dicha acción.

Figura 19. **Vista de la página de contactos**

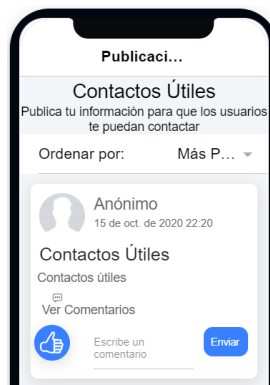


Fuente: elaboración propia.

5.10. **Página de contactos**

Esta pantalla servirá para que los usuarios que quieran ayudar a otros usuarios con su proceso de ingreso a la universidad se puedan comunicar con ellos. Se pretende que los usuarios no compartan información sensible. La aplicación no se hace responsable por cualquier inconveniente que se pueda tener con los usuarios de la aplicación.

Figura 20. **Vista de la página de contactos**



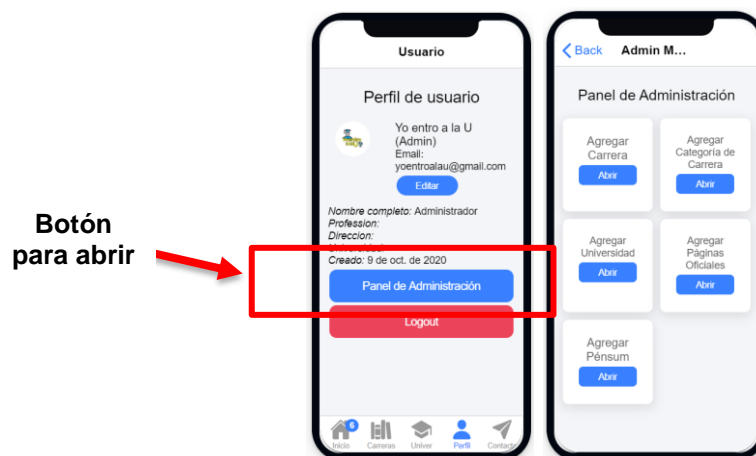
Fuente: elaboración propia.

5.11. Panel de administración

Esta pantalla servirá para acceder al contenido de cada una de las categorías que se presentan en la aplicación (carreras, universidades, entre otros). Solo tendrá acceso el usuario administrador de la aplicación. Habrá varios botones que corresponderán a las distintas acciones que puede ejecutar un usuario administrador de la aplicación.

Al hacer clic sobre alguno de los botones saltará una página con una lista de elementos, los cuales pertenecerán a alguna de las secciones que ahí se presentan. Por ejemplo, al hacer clic sobre el botón “Abrir” para la sección “Agregar Carrera”, se mostrará una pantalla con la lista de todas las categorías que se encuentran disponibles en el sistema. Cada elemento de esta lista tendrá un botón de editar y otro de eliminar. En la parte inferior de la pantalla de lista se tendrá un botón con un símbolo de “+”, el cual al pulsarlo llevará a la pantalla de nueva publicación.

Figura 21. Vista inicial del panel de administración



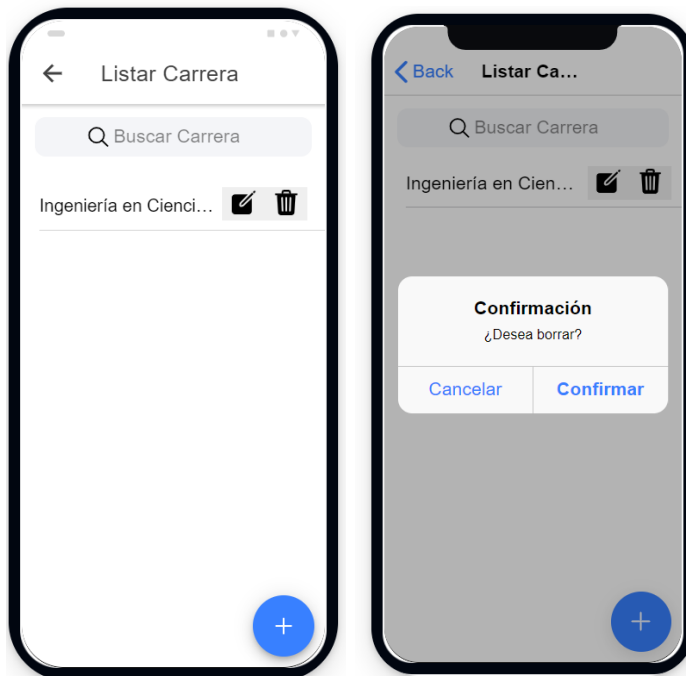
Fuente: elaboración propia.

5.12. Página del listado de categorías

Esta pantalla servirá para enlistar, crear, editar y eliminar los tipos de categorías del sistema. Habrá una lista de categorías con dos botones, el primero para editar la categoría y el segundo para eliminar por completo el contenido de la categoría.

Cuando se intente eliminar la categoría, saltará una pantalla de confirmación. En la parte superior habrá una barra donde se busca por título de la categoría; en la parte inferior derecha habrá un botón para acceder a la pantalla de nueva publicación.

Figura 22. Vista de los listados de categorías



Fuente: elaboración propia.

5.13. Página de edición de categorías

Esta pantalla servirá para editar los diferentes campos de información de una publicación. Cada tipo de publicación tendrá sus campos, algunos con una imagen de icono y otras una imagen con relación 4:3 o modo paisaje. Para manejar estos dos tipos de imagen habrá un módulo de recorte de imagen con el cual el usuario podrá acercar o mover el área de recorte para que la imagen cumpla con las características de tamaño y relación de aspecto.

Además de los campos de imagen habrá de texto, los cuales se deberán llenar con la información de la categoría que se está creando en el sistema. Si alguno de los campos es inválido o está vacío, el sistema indicará con un mensaje en letras rojas, el área de texto que se debe revisar y llenar nuevamente.

Figura 23. Vista de la edición de categorías

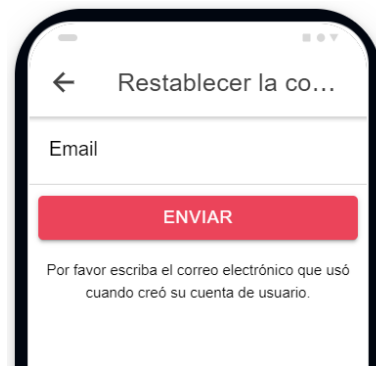


Fuente: elaboración propia.

5.14. Página de recuperación de la contraseña

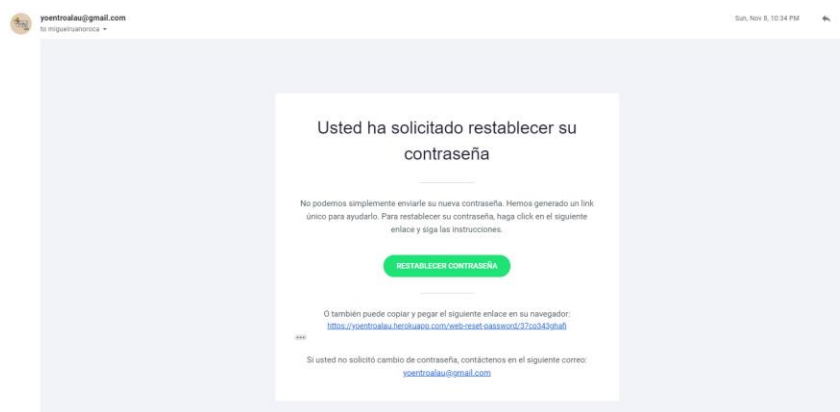
La página de recuperación de contraseña servirá para enviar un correo al usuario de la aplicación, el cual contendrá un enlace que redirigirá a una página de restablecimiento de contraseña.

Figura 24. Vista de la página de restablecimiento de contraseña



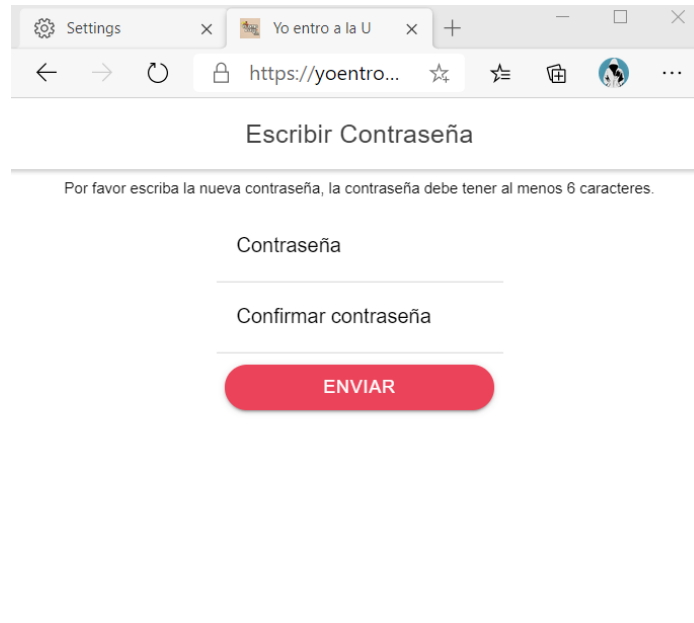
Fuente: elaboración propia.

Figura 25. Vista de correo de restablecimiento de contraseña



Fuente: elaboración propia.

Figura 26. **Vista de la página de restablecimiento de contraseña**



Fuente: elaboración propia.

6. ACCIONES DE USUARIO

Las acciones de usuario describen lo que puede y lo que no puede hacer un usuario dentro de la aplicación. Cada tipo de usuario tendrá sus permisos correspondientes. La plataforma tendrá dos tipos de usuarios.

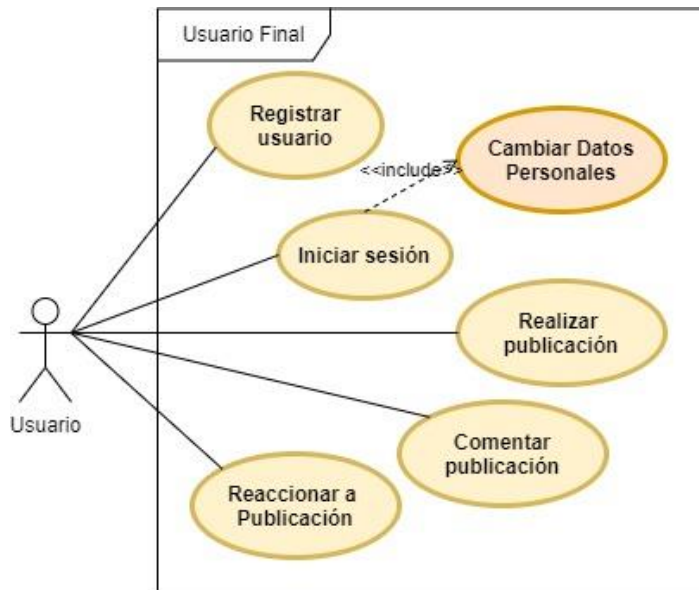
- Usuario final: la persona que consulta y publica información en la plataforma.
- Usuario administrador: la persona que puede revisar y borrar cualquier tipo de información de la plataforma.

6.1. Punto de vista del usuario final de la aplicación

Un usuario final es aquel que pueda consultar, publicar y modificar información, pero no puede ejecutar cambios sustanciales en la estructura de la aplicación.

A continuación, se describe las actividades que podrá ejecutar el usuario final de la plataforma.

Figura 27. **Casos de uso del usuario final**



Fuente: elaboración propia, empleando diagrams.net.

Tabla II. **CDU Registrar usuario**

Código	Caso de Uso 01.
Nombre	Registrar un usuario en la plataforma.
Objetivo	Un usuario final de la aplicación puede crear un perfil de usuario para que sus acciones dentro de la plataforma queden asociadas a dicho perfil.
Descripción	El usuario se dirige a la pantalla de registro y llena los datos requeridos para crear un perfil de usuario.
Actores	Usuario Final.
Condiciones	<ul style="list-style-type: none"> El usuario debe acceder a la aplicación.

Continuación tabla II.

Escenario principal	<ol style="list-style-type: none"> 1. El usuario ingresa a la aplicación y selecciona la pestaña perfil. 2. El usuario selecciona el botón “Registrarse”. 3. El usuario ingresa su información en los campos correspondientes. 4. El sistema guarda el nuevo perfil de usuario.
Escenario de excepción	<ol style="list-style-type: none"> 3.a. Si el usuario ingresa los datos de forma errónea, el sistema solicita al usuario que ingrese de nuevo el campo requerido, con la información correcta.

Fuente: elaboración propia.

Tabla III. **CDU Iniciar sesión**

Código	Caso de Uso 02.
Nombre	Autenticar un usuario en la plataforma
Objetivo	Un usuario final de la aplicación puede acceder a su perfil de usuario para que sus acciones dentro de la plataforma queden registradas con este perfil.
Descripción	El usuario se dirige a la pantalla de usuario y llena los datos del formulario para acceder a las funciones del perfil.
Actores	Usuario Final.
Condiciones	<ul style="list-style-type: none"> • El usuario debe acceder a la aplicación. • El usuario debe tener una cuenta registrada.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario ingresa a la aplicación y selecciona la pestaña perfil. 2. El usuario pulsa el botón “Iniciar sesión”. 3. El usuario ingresa las credenciales correspondientes. 4. Si las credenciales coinciden con algún registro de la base de datos, la plataforma permite el ingreso al perfil de usuario.
Escenario de excepción	<ol style="list-style-type: none"> 3.a. Si el usuario ingresa los datos de forma errónea, el sistema solicita al usuario que ingrese de nuevo el campo requerido, con la información correcta.

Fuente: elaboración propia.

Tabla IV. **CDU Cambiar datos personales**

Código	Caso de Uso 03
Nombre	Cambiar datos de un usuario en la plataforma
Objetivo	Un usuario final de la aplicación puede acceder a la página perfil de usuario para cambiar sus datos personales.
Descripción	El usuario se dirige a la pantalla de cambio de datos de usuario y llena los datos del formulario para modificar su información.
Actores	Usuario Final
Condiciones	<ul style="list-style-type: none"> • El usuario debe acceder a la aplicación. • El usuario debe tener una cuenta registrada.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario ingresa a la aplicación y selecciona la pestaña perfil. 2. El usuario pulsa el botón “Iniciar sesión”. 3. El usuario ingresa las credenciales correspondientes 4. Si las credenciales coinciden con algún registro de la base de datos, la plataforma permite el ingreso al perfil de usuario 5. El usuario pulsa el botón “Editar”. 6. El usuario ingresa su información en los campos correspondientes. 7. El usuario pulsa el botón “Actualizar” 8. El sistema actualiza los datos.
Escenario de excepción	<p>3.a. Si el usuario ingresa los datos de forma errónea, el sistema solicita al usuario que ingrese de nuevo el campo requerido, con la información correcta.</p> <p>6.a. Si el usuario ingresa los datos de forma errónea, el sistema solicita al usuario que ingrese de nuevo el campo requerido, con la información correcta.</p>

Fuente: elaboración propia.

Tabla V. **CDU Realizar publicación**

Código	Caso de Uso 04
Nombre	Realizar una publicación en la plataforma

Continuación tabla V.

Objetivo	Un usuario final de la aplicación puede crear una publicación en cualquier categoría de la aplicación.
Descripción	El usuario se dirige a la a una pantalla que contenga de publicaciones y presiona el botón con un "+". Seguidamente, rellena los campos requeridos y presiona publicar
Actores	Usuario Final
Condiciones	<ul style="list-style-type: none"> • El usuario debe acceder a la aplicación. • El usuario debe tener una cuenta registrada. • El usuario debe ingresar a un listado de publicaciones
Escenario principal	<ol style="list-style-type: none"> 1. El usuario ingresa al listado de publicaciones y presiona el botón con el símbolo "+". 2. El sistema muestra la pantalla de nueva publicación. 3. El usuario llena la información en los campos correspondientes. 4. El usuario presiona el botón "Publicar". 5. El sistema guarda la nueva publicación.
Escenario alternativo	<ol style="list-style-type: none"> 3.a. El usuario selecciona un archivo 3.b. El sistema sube el archivo al servidor y guarda una referencia local en la página de nueva publicación
Escenario de excepción	<ol style="list-style-type: none"> 3.a. Si el usuario ingresa los datos de forma errónea, el sistema solicita al usuario que ingrese de nuevo el campo requerido, con la información correcta.

Fuente: elaboración propia.

Tabla VI. **CDU Comentar publicación**

Código	Caso de Uso 05
Nombre	Comentar una publicación de la plataforma
Objetivo	Un usuario final de la aplicación puede publicar un comentario en una publicación hecha en el sistema.
Descripción	El usuario selecciona una publicación, llena el cuadro de texto con su comentario y presiona el botón "Enviar".
Actores	Usuario Final

Continuación tabla VI.

Condiciones	<ul style="list-style-type: none"> • El usuario debe acceder a la aplicación.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario ingresa a la aplicación y selecciona una publicación. 2. El usuario ingresa el contenido del comentario en la cajilla de texto correspondiente. 3. El usuario presiona “Enviar”. 4. El sistema asocia el comentario a la publicación y lo muestra al usuario.
Escenario de excepción	3.a. Si el usuario no ingresa contenido en la publicación el botón enviar no funcionará.

Fuente: elaboración propia.

Tabla VII. **CDU Reaccionar a publicación**

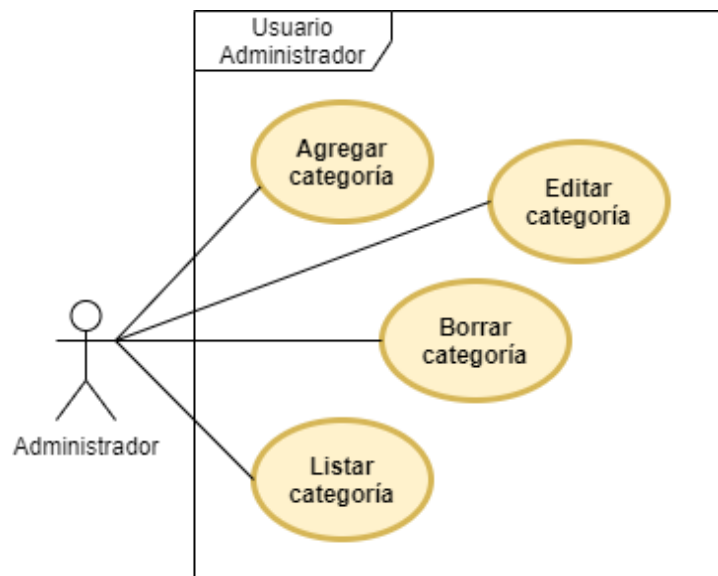
Código	Caso de Uso 06
Nombre	Reaccionar a una publicación de la plataforma
Objetivo	Un usuario final de la aplicación puede acceder a un listado de publicaciones o a un detalle de publicación y reaccionar usando uno de los 5 emoticones que ofrece el sistema.
Descripción	El usuario se dirige a un listado de publicaciones, presiona el botón de reacción y este despliega otra serie de botones que se tendrán que pulsar para registrar la reacción.
Actores	Usuario Final
Condiciones	<ul style="list-style-type: none"> • El usuario debe acceder a la aplicación. • El usuario debe acceder a un listado de publicaciones.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario ingresa a la aplicación y selecciona algún listado de publicaciones. 2. El usuario pulsa el botón con un emoticono de <i>like</i>. 3. El sistema muestra la serie de botones de reacción. 4. El usuario presiona uno de los botones. 5. El sistema registra la reacción y la asocia a la publicación.
Escenario alternativo	4.a. El usuario no presiona ningún botón del área de botones, sino afuera, ocasionando que se cierre la misma.

Fuente: elaboración propia.

6.2. Punto de vista del usuario administrador de la aplicación

El usuario administrador podrá ejecutar las mismas acciones que un usuario final y acciones adicionales. Por categoría se entienden las carreras, universidades, páginas oficiales y páginas de pènsum. A continuación, se describen las actividades que podrá ejecutar el usuario administrador de la plataforma.

Figura 28. Casos de uso del usuario administrador



Fuente: elaboración propia, empleando diagrams.net.

Tabla VIII. CDU Agregar categoría

Código	Caso de Uso 07
Nombre	Agregar categoría a la plataforma
Objetivo	Un usuario administrador de la aplicación puede acceder a la página de creación de categoría y agregar un nuevo registro.

Continuación tabla VIII.

Descripción	El usuario se dirige a la pantalla de creación de categoría y llena los datos del formulario para crear un nuevo registro de categoría en la aplicación.
Actores	Administrador
Condiciones	<ul style="list-style-type: none"> • El administrador debe tener una cuenta registrada. • El administrador debe acceder a la aplicación. • El administrador debe iniciar sesión.
Escenario principal	<ol style="list-style-type: none"> 1. El administrador ingresa a la aplicación y selecciona la pestaña perfil. 2. El administrador selecciona el botón “Panel de administración”. 3. El administrador pulsa el botón “Abrir” de la categoría. 4. El administrador pulsa el botón con símbolo “+”. 5. El administrador ingresa la información en los campos correspondientes. 6. El administrador pulsa el botón “Publicar”. 7. El sistema publica los datos correspondientes.
Escenario de excepción	4.a. Si el administrador ingresa los datos de forma errónea, el sistema solicita al usuario que ingrese de nuevo el campo requerido, con la información correcta.

Fuente: elaboración propia.

Tabla IX. **CDU Editar categoría**

Código	Caso de Uso 08
Nombre	Editar categoría de la plataforma
Objetivo	Un usuario administrador de la aplicación puede acceder a la pantalla de edición de categoría y modificar su contenido.
Descripción	El administrador se dirige a la pantalla de edición de categoría, modifica los datos y el sistema actualiza la información.
Actores	Administrador

Continuación tabla IX.

Condiciones	<ul style="list-style-type: none"> • El administrador debe tener una cuenta registrada. • El administrador debe acceder a la aplicación. • El administrador debe iniciar sesión.
Escenario principal	<ol style="list-style-type: none"> 1. El administrador ingresa a la aplicación y selecciona la pestaña perfil. 2. El administrador selecciona el botón “Panel de administración”. 3. El administrador pulsa el botón “Abrir” de la categoría. 4. El administrador busca en el listado de categorías disponibles y presiona el botón de editar. 5. El administrador ingresa la información en los campos correspondientes. 6. El administrador pulsa el botón “Publicar”. 7. El sistema actualiza los datos correspondientes.
Escenario de excepción	<ol style="list-style-type: none"> 5.a. Si el administrador ingresa los datos de forma errónea, el sistema solicita al usuario que ingrese de nuevo el campo requerido, con la información correcta.

Fuente: elaboración propia.

Tabla X. **CDU Borrar categoría**

Código	Caso de Uso 09
Nombre	Borrar categoría de la plataforma
Objetivo	Un usuario administrador de la aplicación puede acceder a la pantalla de eliminación de categoría y eliminar una categoría.
Descripción	El administrador se dirige a la pantalla de eliminación de categoría, selecciona una categoría y presiona el botón eliminar.
Actores	Administrador
Condiciones	<ul style="list-style-type: none"> • El administrador debe tener una cuenta registrada. • El administrador debe acceder a la aplicación. • El administrador debe iniciar sesión.

Continuación tabla X.

Escenario principal	<ol style="list-style-type: none"> 1. El administrador ingresa a la aplicación y selecciona la pestaña perfil. 2. El administrador selecciona el botón “Panel de administración”. 3. El administrador pulsa el botón “Abrir” de la categoría. 4. El administrador busca en el listado de categorías disponibles y presiona el botón de eliminar. 5. El administrador confirma la eliminación. 6. El sistema elimina los datos correspondientes.
Escenario de excepción	<ol style="list-style-type: none"> 5.a. El administrador no confirma la eliminación y el sistema no modifica registro alguno.

Fuente: elaboración propia.

Tabla XI. **CDU Listar categoría**

Código	Caso de Uso 10
Nombre	Listar categorías de la plataforma
Objetivo	Un usuario administrador de la aplicación puede acceder a las pantallas donde se listan las diferentes categorías.
Descripción	El administrador se dirige a la pantalla del listado de categorías y puede observar la información de cada una.
Actores	Administrador
Condiciones	<ul style="list-style-type: none"> • El usuario debe tener una cuenta registrada. • El administrador debe acceder a la aplicación. • El usuario debe iniciar sesión.
Escenario principal	<ol style="list-style-type: none"> 7. El administrador ingresa a la aplicación y selecciona la pestaña perfil. 8. El administrador selecciona el botón “Panel de administración”. 9. El administrador pulsa el botón “Abrir” de la categoría. 10. El sistema muestra el listado de categorías.

Fuente: elaboración propia.

7. DOCUMENTACIÓN

7.1. Requisitos

Estos son los requisitos necesarios para un funcionamiento óptimo de la aplicación. Si no se cumple alguno de ellos, la aplicación funcionará de manera inesperada.

7.1.1. Instalación de la aplicación desarrollada

- Smartphone con Android 5.0 (*Lollipop*, nivel de API 21) o superior.
- Permisos de almacenamiento y de internet.
- Espacio de 10 MB para la instalación.

7.2. Herramientas para el desarrollo de la aplicación

Estas son las herramientas necesarias para diseñar, visualizar y construir la aplicación.

7.2.1. Visual Studio Code

Es un editor de código gratuito, multiplataforma, rápido y liviano desarrollado por Microsoft para Windows, Linux y OS X.

7.2.2. Android SDK

El SDK (Software Development Kit) de Android provee herramientas necesarias para el desarrollo de una aplicación basada en el lenguaje Java o Kotlin, que son los motores de una aplicación para teléfonos Android.

7.2.3. Google USB Driver

Es un controlador que se utiliza para depurar dispositivos con sistema operativo Android. Es necesario para probar la aplicación en un entorno real.

7.2.4. MongoDB Atlas

Es un servicio de MongoDB que provee un servidor de base de datos en la nube; es gratuito si al usarlo no se sobrepasa de ciertos límites mensuales. Provee una interfaz intuitiva y fácil de implementar para conectar una aplicación móvil a una base de datos de MongoDB. La aplicación “Yo entro a la U” hace uso de estos servidores y de esta base de datos.

7.2.5. Capacitor

Es un entorno de desarrollo nativo y de código abierto usado para construir aplicaciones web nativas. Utiliza HTML, CSS y JavaScript para crear aplicaciones multiplataforma (iOS, Android y Aplicaciones Progresivas).

7.3. Hardware

Ionic depende de que estas herramientas de línea de comandos estén disponibles en su entorno y de una computadora con los siguientes requerimientos.

- Windows
 - Microsoft® Windows® 7/8/10 (32 o 64 bits).
 - 3 GB de RAM, 8 GB de RAM recomendado (además de 1 GB para el Emulador de Android).
 - 2 GB de espacio mínimo, 4 GB de RAM recomendado.
 - Resolución mínima de 1 280 x 800 píxeles.
 - Java 8.

- Mac
 - Mac® OS X® 10.10 (Yosemite) o superior.
 - 3 GB de RAM, 8 GB de RAM recomendado (además de 1 GB para el Emulador de Android).
 - 2 GB de espacio mínimo, 4 GB recomendado.
 - Resolución mínima de 1 280 x 800.
 - Java 7.

- Linux
 - GNOME o KDE Desktop.
 - Ubuntu® 14.04 LTS (Trusty Tahr).
 - 64 bits / 32 bits.
 - GNU C (glibc) 2.1 o superior.
 - 2 GB de RAM, 8 GB de RAM recomendado (además de 1 GB para el Emulador de Android).

- Resolución mínima de 1 280 x 800 píxeles.
- 2 GB de espacio mínimo, 4 GB recomendado.
- Java 8.

7.4. Software

Serán todos los programas de computadora que se usarán para establecer un entorno de desarrollo para la aplicación.

- Visual Studio Code
- Android SDK
- Windows / Linux / MacOS
- Capacitor SDK

7.5. Tutorial de instalación

Consiste en la serie de pasos que se deben seguir para establecer el entorno de desarrollo que permite ejecutar, modificar y visualizar la aplicación.

7.5.1. Instalación en Windows

Serie de pasos que se deben seguir para ejecutar el entorno de desarrollo en una computadora que ejecute el sistema operativo Windows.

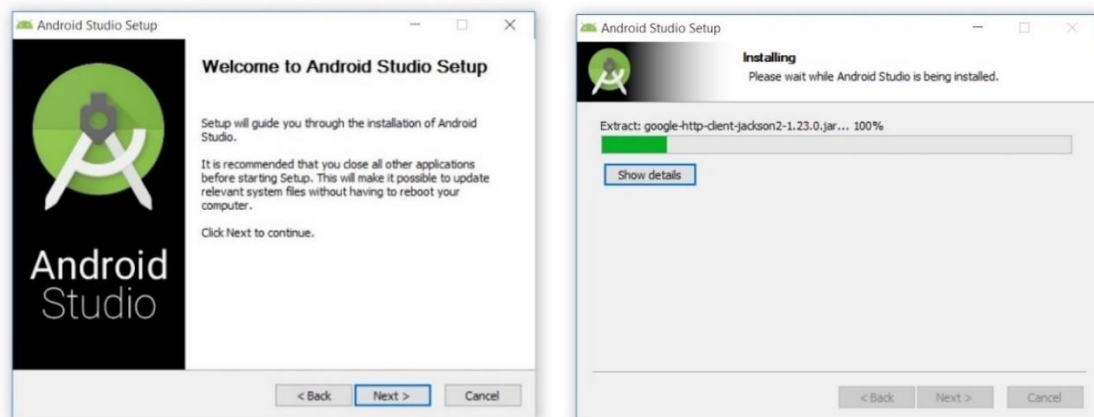
7.5.1.1. Descargar e instalar Android Studio

Ionic necesita una instalación completa de Android Studio para procesar el código de JavaScript compilado desde el SDK de Capacitor. Android Studio se

usará para generar los compilados necesarios para subir la aplicación a Google Play Store.

- Descargar e instalar Android Studio usando el siguiente enlace: <https://developer.android.com/studio/#downloads>.
- Iniciar el instalador de Android Studio y seguir las instrucciones del asistente de instalación.

Figura 29. **Instalación de Android Studio**



Fuente: elaboración propia.

7.5.1.2. **Descargar e instalar Visual Studio Code**

VS Code es un editor ligero y de código abierto que posee una consola integrada que permite ejecutar, probar y depurar aplicaciones construidas con JavaScript.

- Descargar e instalar Visual Studio Code usando el siguiente enlace: <https://code.visualstudio.com/>.

- Seguir las instrucciones del instalador hasta finalizar con la instalación.

7.5.1.3. Descargar e instalar NodeJS

NodeJS es un entorno de tiempo de ejecución para JavaScript, que incluye todo lo necesario para ejecutar aplicaciones móviles escritas en lenguaje JavaScript.

- Descargar e instalar NodeJS usando el siguiente enlace: <https://nodejs.org/en/download/>.
- Hacer doble clic para iniciar el instalador, si el sistema operativo pregunta, hacer clic en “Run”.
- Cuando se presente la página de bienvenida, hacer clic en “Siguiente”.
- Aceptar los términos de licencia y hacer clic en “Siguiente”.
- Cuando se pregunte el destino de instalación, dejar la ruta por defecto y hacer clic en “Siguiente”.
- Esperar a que termine la instalación y por último cerrar la ventana haciendo clic en “Finalizar”.

7.5.2. Configurar dispositivo Android

Para probar y ejecutar la aplicación “Yo entro a la U” en un dispositivo móvil con Android, se necesitará por lo menos un dispositivo con Android 5.0, nivel de API 21 (también llamado Lollipop) o superior.

- En las configuraciones del dispositivo, habilitar las opciones de desarrollo y la depuración USB. Instrucciones más detalladas se pueden encontrar en la siguiente página: <https://developer.android.com/studio/debug/dev-options>.
- Instalación del controlador USB en la computadora de desarrollo. En Android Studio desde el SDK manager como se explica a continuación y en la página: <https://developer.android.com/studio/run/win-usb>.
 - Hacer clic en Tools > Android > SDK Manager
 - Hacer clic en la pestaña SDK Tools
 - Seleccionar Google USB Driver Package y hacer clic en instalar
- Haciendo uso de un cable USB, conectar el teléfono a la computadora, si se solicita, autorizar el acceso al dispositivo.

7.6. Acceso al repositorio y ejecución del código

El código fuente de la aplicación se encuentra alojado en dos repositorios de Github; por lo tanto, se versiona usando la herramienta Git. El código se encuentra separado de la siguiente manera:

7.6.1. Código del backend

El código fuente se encuentra alojado en el siguiente enlace: <https://github.com/mikeruano3/tesis-backend>. Una vez descargado el código, ejecutar los siguientes comandos en consola:

- *npm install* (para instalar las dependencias).
- *node index.js* (para iniciar la ejecución de los archivos de JavaScript).

7.6.2. Código del frontend

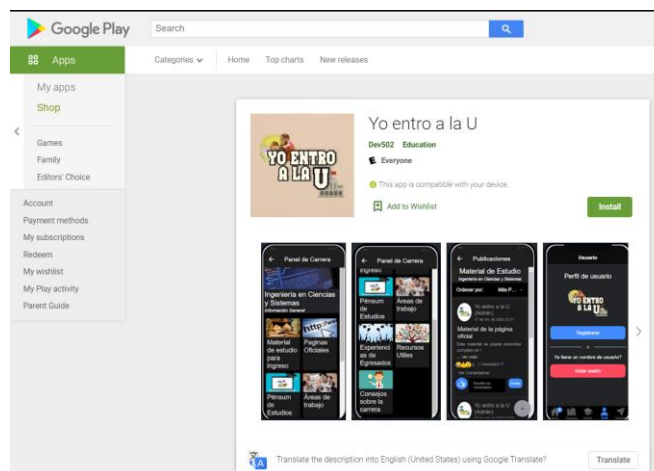
El código fuente del frontend se encuentra alojado en el siguiente enlace: <https://github.com/mikeruano3/tesis-frontend>. Una vez descargado el código, ejecutar los siguientes comandos en consola:

- `npm install` (para instalar las dependencias).
- `npm install -g ionic` (para instalar Ionic).
- `ionic serve -l` (para iniciar Ionic Lab).

7.7. Publicación en la tienda Google Play

La aplicación está disponible en el siguiente enlace: <https://play.google.com/store/apps/details?id=gcreators.yoentroatlau.app>.

Figura 30. Vista de la aplicación en Google Play



Fuente: elaboración propia.

CONCLUSIONES

1. La aplicación ofrece una guía para estudiantes que desean entrar a la universidad.
2. Se ofrece una aplicación cuyo contenido se puede utilizar para realizar exámenes básicos y específicos por carrera.
3. La aplicación es capaz de mostrar contactos de personas que prestan servicios educativos.
4. Las publicaciones dentro de la aplicación promueven el intercambio de material educativo.
5. Los usuarios pueden publicar sus experiencias dentro de la aplicación usando las áreas específicas para ello.
6. Con las publicaciones en la aplicación, se puede ayudar a los usuarios a tener un mejor conocimiento de la situación universitaria.
7. Los estudiantes de las carreras pueden publicar información sobre los trabajos en los que ejercen sus profesiones.
8. Las carreras proyectadas en la aplicación son explicadas desde un punto de vista estudiantil.

RECOMENDACIONES

1. Promover el uso de la aplicación entre los aspirantes a estudiantes universitarios.
2. Incitar a los estudiantes universitarios a que compartan en la aplicación las experiencias personales que tuvieron en sus respectivas carreras.
3. Invitar a algunos usuarios para que se vuelvan administradores de la aplicación y creen nuevos registros de carrera dentro de la aplicación.

BIBLIOGRAFÍA

1. AGUIAR, Edgar; PASTEN, Alejandro; ASCENCIO, Sandra. *Firestore en el desarrollo de aplicaciones móviles*. [en línea]. <<http://repositorio.upsin.edu.mx/formatos/TesinaEdgarUlisesAgueroAguiar6855.pdf>>. [Consulta: 30 de septiembre de 2020].
2. GRIFFITH, Chris. *Mobile App Development with Ionic, Revise Edition*. [en línea]. <<https://www.oreilly.com/library/view/mobile-app-development/9781491998113/ch01.html>>. [Consulta: 2 de noviembre de 2020].
3. MAZA, Jhonathan; AVILA, David; PESANTEZ, Luz. *Desarrollo de una aplicación móvil utilizando el framework MEAN Stack e IONIC: Un estudio de caso en una compañía de transporte*. [en línea]. <<https://journals.gdeon.org/index.php/esj/article/download/74/58>>. [Consulta: 3 de noviembre de 2020].
4. PANJUTA, Denis. *How to publish your Android App on Play Store 2020*. [en línea]. <<https://tutorials.eu/generate-a-signed-apk-in-android-studio-and-upload-it-on-google-play-store>>. [Consulta: 4 de noviembre de 2020].
5. PEÑA, Juan. *¿QUÉ ES FIREBASE Y QUÉ NOS APORTA?* [en línea]. <<https://arpentechnologies.com/es/blog/aplicaciones-movil/que-es-firebase-y-que-nos-aporta>>. [Consulta: 24 de octubre de 2020].

6. TOBAR, Luis. *La educación superior en Guatemala en la primera década del siglo XXI* [en línea]. <<https://www.redalyc.org/articulo.oa?id=179422350009>>. [Consulta: 23 de octubre de 2020].
7. USAOLA, Macario Polo. *MongoDB: gestión, administración y desarrollo de aplicaciones*. 1a ed. España: Universidad de Castilla-La Mancha, 2015. ISBN: 9781326504274. 193 p.
8. YAQ, Paula. *Los 7 errores más frecuentes a la hora de elegir carrera*. [en línea]. <<https://yaq.es/reportajes/elegir-carrera-nos-equivocamos>>. [Consulta: 24 de octubre de 2020].

APÉNDICES

Apéndice 1. Logotipo de la aplicación



Fuente: elaboración propia.

Apéndice 2. Logotipo de la cuenta de desarrollador de Google Play



Fuente: elaboración propia.

