

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN PROTOTIPO DE UNA APLICACIÓN WEB PARA LA ASIGNACIÓN Y SEGUIMIENTO DE PRÁCTICAS PREPROFESIONALES EN LA FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA DE LA EPN

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERA EN ELECTRÓNICA Y REDES DE INFORMACIÓN**

MAYTÉ SOLEDAD BÁEZ BARRAGÁN

DIRECTOR: ING. PABLO WILIAN HIDALGO LASCANO, MSc.

Quito, noviembre 2021

AVAL

Certifico que el presente trabajo fue desarrollado por Mayté Soledad Báez Barragán, bajo mi supervisión.

MSc. PABLO WILIAN HIDALGO LASCANO
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Mayté Soledad Báez Barragán declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

MAYTÉ SOLEDAD BÁEZ BARRAGÁN

DEDICATORIA

A mi familia que significa todo para mí: mis padres y hermanos.

AGRADECIMIENTO

En primer lugar, quiero agradecer a mis padres por la dedicación con la que me han criado, porque me han inculcado grandes valores y por ellos he llegado a ser lo que soy hoy. Y a mis hermanos por ser mis amigos y compañeros de aventuras desde el momento en que nací.

Gracias a mis amigos, a Nohelia por ser mi mejor amiga y estar presente siempre. A mis amigos Darío y Felipe que han sido mis compañeros a lo largo de esta carrera. Y a Jorge, que en los últimos meses se ha convertido en un gran amigo.

A mis amigos del colegio por tantos años de amistad incondicional: Renata, Nicole, Stefy y Pablo.

Por último, quiero agradecer a mi directora de tesis, la PhD. Priscilla, que ha sido una gran guía en este camino y desde el inicio mostró dedicación para lograr un excelente trabajo. Y al MSc. Pablo Hidalgo me ha brindado su ayuda a lo largo de la carrera universitaria.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS.....	X
ÍNDICE DE CÓDIGO.....	XI
RESUMEN.....	XIII
ABSTRACT.....	XIV
1 INTRODUCCIÓN.....	1
1.1 OBJETIVOS.....	2
1.2 ALCANCE.....	2
1.3 MARCO TEÓRICO	6
1.3.1 FRONTEND	6
1.3.2 BACKEND	14
2 METODOLOGÍA.....	17
2.1 LEVANTAMIENTO DE INFORMACIÓN.....	17
2.1.1 LEVANTAMIENTO DEL PROCESO.....	17
2.1.2 REQUERIMIENTOS.....	25
2.2 DISEÑO DE LA APLICACIÓN	33
2.2.1 ARQUITECTURA DE LA APLICACIÓN.....	33
2.2.2 ESQUEMA DE DATOS	35
2.2.3 DIAGRAMA DE ESTADOS.....	37
2.2.4 DISEÑO DE LA INTERFAZ GRÁFICA	38
2.2.5 DIAGRAMAS DE NAVEGACIÓN	46
2.3 IMPLEMENTACIÓN DEL PROTOTIPO	50
2.3.1 CREACIÓN DEL PROYECTO.....	50
2.3.2 IMPLEMENTACIÓN BACKEND	52
2.3.3 IMPLEMENTACIÓN FRONTEND.....	59
3 RESULTADOS Y DISCUSIÓN.....	85
3.1 PRUEBAS DE FUNCIONAMIENTO.....	85

3.1.1	CREACIÓN Y CONFIGURACIÓN DE USUARIOS.....	85
3.1.2	GESTIÓN DE ANUNCIOS.....	90
3.1.3	GESTIÓN DE PRÁCTICAS PREPROFESIONALES.....	94
3.2	PRUEBAS DE USUARIO.....	98
3.2.1	RESULTADOS.....	100
4	CONCLUSIONES Y RECOMENDACIONES.....	105
4.1	CONCLUSIONES.....	105
4.2	RECOMENDACIONES.....	107
5	REFERENCIAS BIBLIOGRÁFICAS.....	109
	ANEXOS.....	113

ÍNDICE DE FIGURAS

Figura 1.1. Logotipo de EXPAPP.....	3
Figura 1.2. Diagrama de infraestructura de EXPAPP.....	3
Figura 1.3. a) HTML usando directiva <i>ngFor</i> . b) Resultado del uso de la directiva <i>ngFor</i>	11
Figura 2.1. Diagrama de actividad del proceso de prácticas preprofesionales en la FIEE.....	26
Figura 2.2. Diagrama de contexto de la Aplicación Web para obtención y manejo de prácticas preprofesionales.....	27
Figura 2.3. Diagrama de casos de uso para el rol Administrador Técnico.....	29
Figura 2.4. Diagrama de casos de uso para el rol Tutor.....	29
Figura 2.5. Diagrama de casos de uso para el rol Administrador.....	30
Figura 2.6. Diagrama de casos de uso para el rol Supervisor Empresa.....	30
Figura 2.7. Diagrama de casos de uso para el rol Estudiante.....	31
Figura 2.8. Diagrama de arquitectura de EXPAPP.....	34
Figura 2.9. Diseño de los módulos de la aplicación web.....	35
Figura 2.10. Diagrama de modelos de EXPAPP.....	36
Figura 2.11. Diagrama de estados de EXPAPP.....	37
Figura 2.12. Diseño de la interfaz de usuario para rol Administrador Técnico.....	39
Figura 2.13. Diseño de la interfaz de usuario para rol Administrador – Estadísticas.....	39
Figura 2.14. Diseño de la interfaz de usuario para rol Administrador – Anuncios Pendientes.....	41
Figura 2.15. Diseño de la interfaz de usuario para rol Administrador – Detalles de anuncio.....	41
Figura 2.16. Diseño de la interfaz de usuario para rol Supervisor Empresa – Creación de anuncio.....	42
Figura 2.17. Diseño de la interfaz de usuario para rol Supervisor Empresa – Detalles de práctica preprofesional.....	43
Figura 2.18. Diseño de la interfaz de usuario para rol Estudiante – Anuncios publicados.....	44
Figura 2.19. Diseño de la interfaz de usuario para rol Estudiante – Creación de CV.....	44
Figura 2.20. Diseño de la interfaz de usuario para rol Tutor – Detalles de práctica preprofesional.....	45
Figura 2.21. Diseño de la interfaz de usuario para rol Tutor – Formulario de informe.....	46
Figura 2.22. Simbología de los diagramas de navegación.....	46
Figura 2.23. Diagrama de navegación del Tutor.....	47

Figura 2.24. Diagrama de navegación del Administrador.	47
Figura 2.25. Diagrama de navegación del Administrador Técnico.....	48
Figura 2.26. Diagrama de navegación del Estudiante.....	49
Figura 2.27. Diagrama de navegación del Supervisor Empresa.....	50
Figura 2.28. Componentes del <i>backend</i>	53
Figura 2.29. Estructura del <i>backend</i> de EXPAPP.	54
Figura 2.30. EXPAPP-FRONTEND: Resultado de la alerta creada en el Código 2.27....	76
Figura 2.31. EXPAPP-FRONTEND: Resultado de la alerta creada en el Código 2.28....	76
Figura 3.1. ADMINISTRADOR TÉCNICO: Creación de usuario Administrador.	86
Figura 3.2. ADMINISTRADOR TÉCNICO: Mensaje obtenido tras crear el usuario Administrador.	86
Figura 3.3. ADMINISTRADOR: Correo electrónico recibido al crearse la cuenta de usuario.....	87
Figura 3.4. ADMINISTRADOR TÉCNICO: Listado de estudiantes.....	87
Figura 3.5. ESTUDIANTE: Formulario de registro de Estudiante en EXPAPP.	88
Figura 3.6. ESTUDIANTE: a) Mensaje de error en la creación de la cuenta Estudiante cuando no se ha cargado los datos a la base. b) Mensaje de error cuando el correo electrónico ya se encuentra en uso.....	88
Figura 3.7. ESTUDIANTE: Creación del CV del estudiante.....	89
Figura 3.8. ESTUDIANTE: Resultado del CV generado en formato PDF.....	89
Figura 3.9. SUPERVISOR EMPRESA: Formulario para la creación de un anuncio.	90
Figura 3.10. ADMINISTRADOR: Pestaña de anuncios pendientes de aprobar.	90
Figura 3.11. ESTUDIANTE: Pantalla de anuncios.	91
Figura 3.12. ESTUDIANTE: Anuncios en los que ha postulado el estudiante.....	92
Figura 3.13. ADMINISTRADOR: Pantalla de detalles del anuncio.	92
Figura 3.14. ADMINISTRADOR: Confirmación de envío de estudiante a empresa.	93
Figura 3.15. ADMINISTRADOR: Ingreso de periodo académico para la práctica preprofesional.....	93
Figura 3.16. ADMINISTRADOR: Mensaje de envío de datos de estudiante exitoso.	93
Figura 3.17. SUPERVISOR EMPRESA: Detalles del anuncio de prueba y el estudiante seleccionado.	94
Figura 3.18. SUPERVISOR EMPRESA: Mensaje de confirmación para el inicio de práctica preprofesional.....	94
Figura 3.19. SUPERVISOR EMPRESA: Pantalla de datos de práctica preprofesional. ..	95
Figura 3.20. ADMINISTRADOR: Pantalla de resumen de prácticas preprofesionales. ...	95
Figura 3.21. ADMINISTRADOR: Asignación de Tutor para la práctica preprofesional de prueba.	96

Figura 3.22. TUTOR: Correo recibido al ser asignado el Tutor.....	96
Figura 3.23. TUTOR: Formulario para completar el Informe.....	97
Figura 3.24. TUTOR: Informe generado (Inf3) tras presionar “ Descargar ” o “ Ver ” de la Figura 3.23.	97
Figura 3.25. ADMINISTRADOR: Estadísticas.	98
Figura 3.26. Interfaz con nueva paleta de colores.....	104

ÍNDICE DE TABLAS

Tabla 1.1. Resumen de funcionalidades de EXPAPP para cada rol de usuario.....	4
Tabla 1.2. Herramientas de desarrollo de Firebase.....	16
Tabla 2.1. Resultados de entrevista al Decano de la FIEE.	18
Tabla 2.2. Resultados de entrevistas a estudiantes de la FIEE.	21
Tabla 2.3. Resultados de entrevistas a miembros de dos empresas.....	23
Tabla 2.4. Detalles de Informes de seguimiento de práctica preprofesional y responsables en la FIEE.....	28
Tabla 2.5. Requerimientos funcionales de EXPAPP.	32
Tabla 2.6. Requerimientos no funcionales de EXPAPP.	33
Tabla 2.7. Descripción del diseño de la interfaz gráfica para rol Administrador Técnico.....	38
Tabla 2.8. Descripción del diseño de la interfaz gráfica para rol Administrador.	40
Tabla 2.9. Descripción del diseño de la interfaz gráfica para rol Supervisor Empresa.....	42
Tabla 2.10. Descripción del diseño de la interfaz gráfica para rol Estudiante.....	43
Tabla 2.11. Descripción del diseño de la interfaz gráfica para rol Tutor.	45
Tabla 2.12. Pasos para el levantamiento del <i>frontend</i>	51
Tabla 2.13. Pasos para el levantamiento del <i>backend</i>	52
Tabla 2.14. Funciones implementadas en el <i>backend</i>	55
Tabla 2.15. Descripción de los atributos de la clase <i>Task</i>	57
Tabla 2.16. Componentes para la autenticación de usuarios en la aplicación web.....	60
Tabla 2.17. Componentes en común para todos los usuarios de la aplicación web.....	60
Tabla 2.18. Componentes para usuario Administrador técnico.....	61
Tabla 2.19. Componentes para usuario Tutor.....	61
Tabla 2.20. Componentes para Administrador.....	62
Tabla 2.21. Componentes para Empresa.....	63
Tabla 2.22. Componentes para estudiante.	64
Tabla 3.1. Descripción del proceso de pruebas de usuario de EXPAPP.....	99
Tabla 3.2. Resultado del cuestionario aplicado al rol Administrador Técnico.	101
Tabla 3.3. Resultado del cuestionario aplicado a los roles participantes en el proceso de prácticas preprofesionales.....	102
Tabla 3.4. Errores reportados en las pruebas de usuario de EXPAPP.....	103
Tabla 3.5. Cambios sugeridos por los usuarios.	104

ÍNDICE DE CÓDIGO

Código 1.1. Ejemplo de declaraciones y exportaciones en un módulo de Angular (líneas 7 y 8).....	9
Código 1.2. Ejemplo de declaraciones y proveedores en un módulo de Angular (línea 9).	10
Código 1.3. Ejemplo de importaciones en módulos de Angular (línea 8).	10
Código 1.4. Importaciones necesarias en <i>AppRoutingModule</i>	12
Código 1.5. Configuración de rutas en <i>AppRoutingModule</i>	13
Código 1.6. Configuración de <i>imports</i> y <i>exports</i> de <i>AppRoutingModule</i>	13
Código 1.7. Uso de directiva <i>RouterOutlet</i>	14
Código 2.1. EXPAPP-BACKEND: Declaración de módulos necesarios en <i>index.js</i>	54
Código 2.2. EXPAPP-BACKEND: Configuración del transportador de correo electrónico.	56
Código 2.3. EXPAPP-BACKEND: Configuración de la plantilla del correo electrónico que se enviará cuando se crea la cuenta de un usuario.	56
Código 2.4. EXPAPP-BACKEND: Función para enviar correo electrónico a un usuario. 56	
Código 2.5. EXPAPP-BACKEND: Clase <i>Task</i>	57
Código 2.6. EXPAPP-BACKEND: Definición de <i>workers</i>	58
Código 2.7. EXPAPP-BACKEND: Función <i>tasks</i> implementada.	59
Código 2.8. EXPAPP-BACKEND: Método <i>taskConverter</i>	59
Código 2.9. EXPAPP-FRONTEND: Declaración de rutas en <i>AppRoutingModule</i>	65
Código 2.10. EXPAPP-FRONTEND: Configuración de <i>RoleGuard</i>	66
Código 2.11. EXPAPP-FRONTEND: Declaración de rutas hijas para el usuario tutor. ...	66
Código 2.12. EXPAPP-FRONTEND: Ejemplo de cómo importar componentes de Angular Material en un módulo de Angular.	67
Código 2.13. EXPAPP-FRONTEND: Arreglo que contiene los elementos que se importan desde Angular Material.	68
Código 2.14. EXPAPP-FRONTEND: Definición de <i>imports</i> , <i>exports</i> y <i>providers</i> de <i>AngularMaterialModule</i> (líneas 98-100).	68
Código 2.15. EXPAPP-FRONTEND: Clase <i>MatPaginatorIntlSpanish</i>	69
Código 2.16. EXPAPP-FRONTEND: Importación de <i>AngularMaterialModule</i> en <i>AppModule</i>	69
Código 2.17. EXPAPP-FRONTEND: Implementación de botón de <i>Angular Material</i>	70
Código 2.18. EXPAPP-FRONTEND: Ejemplo de importación de <i>FlexLayoutModule</i> en <i>TutorModule</i> (línea 15 y 32).	70
Código 2.19. EXPAPP-FRONTEND: Ejemplo de uso de <i>FlexLayout</i> (línea 4).	71

Código 2.20.	EXPAPP-FRONTEND: Declaración de variables para la configuración de Ng2-charts.....	71
Código 2.21.	EXPAPP-FRONTEND: Código HTML requerido para la implementación de Ng2-charts.....	72
Código 2.22.	EXPAPP-FRONTEND: Importación de Pdfmake en el componente que se va a utilizar.	72
Código 2.23.	EXPAPP-FRONTEND: Método para abrir documento PDF en otra pestaña del navegador (línea 203-206) y método para descargar documento PDF (líneas 209-211).....	73
Código 2.24.	EXPAPP-FRONTEND: Función <i>getDocumentDefinition</i>	74
Código 2.25.	EXPAPP-FRONTEND: Continuación de la función <i>getDocumentDefinition</i>	74
Código 2.26.	EXPAPP-FRONTEND: Importación de <i>Swal</i> en el componente LoginComponent.....	75
Código 2.27.	EXPAPP-FRONTEND: Ejemplo de uso de <i>Swal</i> para mostrar alertas.....	75
Código 2.28.	EXPAPP-FRONTEND: Uso de alerta de <i>Swal</i> con botón de confirmación.....	76
Código 2.29.	EXPAPP-FRONTEND: Configuración del archivo <i>environment.prod.ts</i> para uso de Firebase en el proyecto de Angular.	77
Código 2.30.	EXPAPP-FRONTEND: Módulos importados en AppModule para uso de <i>AngularFire</i>	77
Código 2.31.	EXPAPP-FRONTEND: Inyección de dependencias de <i>AngularFire</i> en los servicios.....	78
Código 2.32.	EXPAPP-FRONTEND: Importación de todos los módulos de Firebase.....	78
Código 2.33.	EXPAPP-FRONTEND: Uso de <i>AngularFireAuth</i>	80
Código 2.34.	EXPAPP-FRONTEND: Función para cambio de contraseña.....	81
Código 2.35.	EXPAPP-FRONTEND: Uso de <i>Firebase storage</i>	82
Código 2.36.	EXPAPP-FRONTEND: Uso de <i>AngularFirestore</i> para traer registros de la base de datos.....	83
Código 2.37.	EXPAPP-FRONTEND: Uso de <i>AngularFirestore</i> para actualizar registros de la base de datos.....	83
Código 2.38.	EXPAPP-FRONTEND: Uso de <i>AngularFireFunctions</i>	84

RESUMEN

En este documento se presenta el trabajo de titulación que consiste en el diseño y la implementación del prototipo de una aplicación web para la obtención y seguimiento de prácticas preprofesionales en la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional. El prototipo se llama EXPAPP y se compone de *frontend*, que fue desarrollado con Angular 9; y, *backend*, que fue desarrollado sobre Firebase.

El presente documento se divide en cuatro capítulos:

En el primer capítulo se presentan los objetivos del trabajo y su alcance, además, se exponen los conceptos principales que representan la base teórica del desarrollo del prototipo EXPAPP.

El segundo capítulo se divide en tres secciones. En la primera sección se describe el levantamiento del proceso de prácticas preprofesionales que involucra la obtención de los requerimientos de EXPAPP. En la segunda sección se presenta el diseño del prototipo y en la tercera sección se describe su implementación.

En el tercer capítulo se presentan las pruebas de funcionamiento y las pruebas de usuario ejecutadas; y los resultados obtenidos tras aplicar entrevistas de satisfacción a los usuarios. Adicionalmente, se detallan los cambios realizados en el prototipo por errores encontrados durante la fase de pruebas y sugerencias que realizaron los usuarios.

En el cuarto capítulo se presentan las conclusiones y recomendaciones del presente trabajo de titulación.

Como anexos de este trabajo se tiene el Manual de creación de un proyecto de Firebase, el Código del *backend*, el Código del *frontend* y el Manual de usuario de EXPAPP.

PALABRAS CLAVE: Prácticas preprofesionales, Angular, *Frontend*, Firebase, *Backend*.

ABSTRACT

The following degree work presents the prototype design of an online app for obtaining and monitoring professional internships at the Faculty of Electrical and Electronic Engineering of the Escuela Politécnica Nacional. The prototype name is EXPAPP; it consists of a frontend developed with Angular 9, and a backend developed on Firebase.

This document is divided into four chapters:

The first chapter describes this work's objectives and scope, along with the main concepts and the theoretical basis of the EXPAPP prototype development.

The second chapter is divided into three sections. The first section describes the requirement gathering process for the EXPAPP prototype. The second section explains the prototype design, and the third section describes its implementation.

The third chapter presents the EXPAPP functional tests and user tests results. Additionally, the changes implemented to the EXPAPP prototype resulted from the errors reported during the testing phase.

Finally, the fourth chapter presents the conclusions and recommendations of this degree work.

As annexes of this degree work, there are the manual with instructions for creating a Firebase project, backend code, and frontend code, and the EXPAPP User Manual.

KEYWORDS: Internship, Angular, Frontend, Firebase, Backend.

1 INTRODUCCIÓN

Actualmente en la Facultad de Ingeniería Eléctrica y Electrónica (FIEE) de la Escuela Politécnica Nacional uno de los requisitos para la obtención del título de grado es cumplir una cantidad específica de horas de prácticas preprofesionales. Muchos estudiantes tardan en culminar las horas requeridas debido a que presentan dificultades en la búsqueda de empresas en las que puedan realizar prácticas preprofesionales, esto puede provocar que el estudiante extienda el tiempo de permanencia en la universidad.

Una vez que el estudiante ha sido aceptado para realizar prácticas preprofesionales se debe llevar a cabo un seguimiento; para esto, la comisión de prácticas preprofesionales de la FIEE asigna un tutor. Hoy en día el seguimiento se lo realiza de manera manual, mediante llamadas telefónicas o correo electrónico entre el tutor asignado y la empresa. En algunas ocasiones la asignación de tutor y los informes que se requieren son realizados de manera tardía. Todo el proceso de seguimiento se podría automatizar llenando formularios en el prototipo de aplicación web desarrollada en este trabajo de titulación, en la cual también se envíen notificaciones para que tanto la empresa, estudiante y el tutor recuerden llenar dichos informes a tiempo.

Por estas razones se propone la implementación de un prototipo de aplicación web que permita a los estudiantes buscar prácticas preprofesionales que estén directamente relacionadas con su carrera, y para aplicar a un puesto podrán crear su *Currículum Vitae* (CV) en la página llenando un formulario. Además, permitirá el seguimiento de las prácticas preprofesionales tanto a estudiantes, empresa y personal administrativo, así como elaborar de manera automática el registro de horas de cumplidas. Cada usuario recibirá notificaciones al correo electrónico como recordatorio de llenar los informes dentro de los plazos correspondientes.

El prototipo propuesto tiene el fin de automatizar el proceso de prácticas preprofesionales de la FIEE, lo cual es beneficioso para la coordinación del proceso de prácticas preprofesionales, así como para estudiantes y empresas. El personal asignado para el seguimiento de prácticas preprofesionales hoy en día realiza todo este trabajo de manera manual. La automatización de este proceso podría reducir el tiempo que toma normalmente efectuar estas tareas, y mejorar el control de las prácticas preprofesionales que realizan los estudiantes, pues tendrán acceso a toda la información del estudiante y de la empresa.

Las empresas también se beneficiarán pues se disminuiría el trabajo que realizan actualmente, ya que con el uso de esta aplicación no tendrán que seleccionar entre los candidatos, sino que se asignará al estudiante que cumpla con el perfil que requieren.

En este capítulo se presentan los objetivos que tiene el presente trabajo y su alcance. También se describen a detalle los conceptos principales necesarios para la implementación del prototipo de aplicación web.

1.1 OBJETIVOS

El objetivo general de este Trabajo de Titulación es:

Desarrollar un prototipo de una aplicación web para la asignación y seguimiento de prácticas preprofesionales en la Facultad de Ingeniería Eléctrica y Electrónica de la EPN.

Los objetivos específicos son:

- Diagnosticar el proceso de prácticas preprofesionales en la FIEE con una muestra de los involucrados en el proceso.
- Diseñar los componentes que tendrá la aplicación web para cada uno de los roles: administrador técnico, administrador, estudiante, empresa y tutor.
- Implementar los componentes previamente diseñados.
- Validar el correcto funcionamiento del prototipo utilizando dos usuarios para cada uno de los roles propuestos.

1.2 ALCANCE

En este trabajo se presenta el diseño y la implementación del prototipo de una aplicación web para la obtención y seguimiento de prácticas preprofesionales o pasantías en la FIEE de la Escuela Politécnica Nacional (EPN).

La diferencia entre una pasantía y una práctica preprofesional es que la primera es remunerada y la segunda no. En este documento se nombrará a ambas como prácticas preprofesionales debido a que en la FIEE se lo maneja de esta manera.

En la Figura 1.1 se puede observar el logotipo diseñado para la aplicación, a la cual se le dio el nombre de EXPAPP. En EXPAPP se tiene la posibilidad de que los estudiantes de la FIEE encuentren prácticas preprofesionales relacionadas directamente a su carrera y se realice su seguimiento.



Figura 1.1. Logotipo de EXPAPP.

Para que se lleve a cabo el seguimiento de la práctica preprofesional en EXPAPP debe haber la participación de personal administrativo de la FIEE y profesores que realizan las tutorías. Las empresas también forman parte de este proceso. Por estas razones EXPAPP tiene cinco diferentes roles de usuario: Administrador, Administrador técnico, Supervisor Empresa, Estudiante y Tutor.

La interfaz de cada rol de usuario, *frontend*, se desarrolló en el framework Angular 9 y el *backend* se lo implementó con Firebase de Google, del cual se utilizan servicios como base de datos, autenticación y almacenamiento de archivos. Entonces cada usuario interactúa con EXPAPP mediante Internet como se observa en la Figura 1.2.

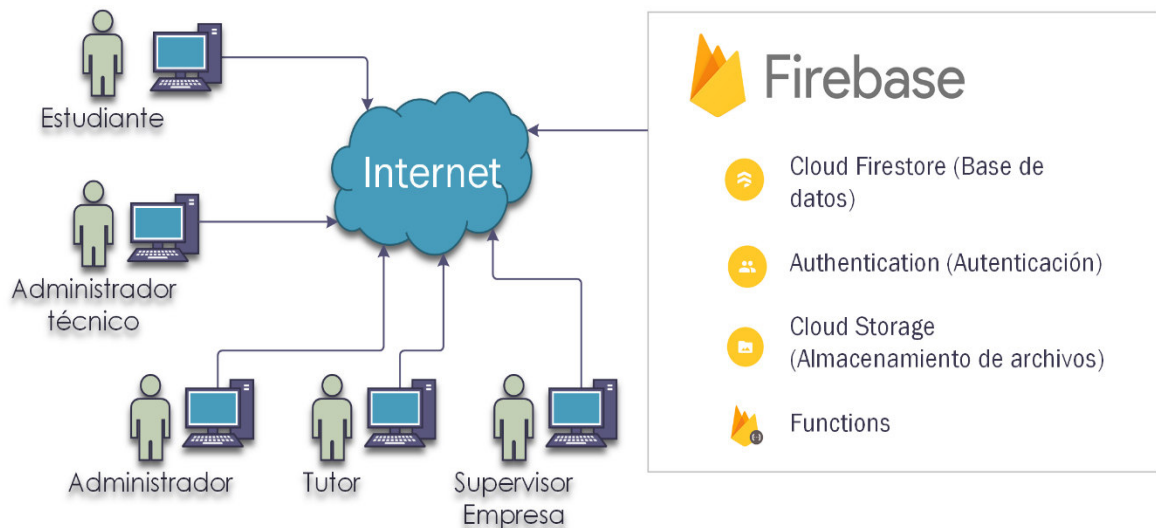


Figura 1.2. Diagrama de infraestructura de EXPAPP.

Desde EXPAPP se lleva a cabo el proceso de prácticas preprofesionales de la FIEE de principio a fin. En la Tabla 1.1 se presentan las acciones que cada rol de usuario puede realizar en la aplicación web.

Tabla 1.1. Resumen de funcionalidades de EXPAPP para cada rol de usuario.

Rol	Funcionalidad
Administrador técnico	<ul style="list-style-type: none"> • Realizar el CRUD de usuarios, carreras, materias y periodos académicos. • Ver información de prácticas preprofesionales.
Administrador	<ul style="list-style-type: none"> • Ingresar o cambiar sus datos. • Aprobar o descartar el anuncio de prácticas preprofesionales que sea creado por una empresa. • Crear un anuncio para una empresa. • Visualizar la lista de estudiantes que han aplicado una práctica preprofesional. • Seleccionar un estudiante o estudiantes que cumplan con los requisitos de la empresa. • Asignar o cambiar tutor de una práctica preprofesional en curso. • Enviar una invitación para que empresas realicen la solicitud de practicantes. • Visualizar información completa de estudiantes, tutores y empresas. • Visualizar estadísticas relacionadas a estudiantes, tutores y empresas.
Supervisor Empresa	<ul style="list-style-type: none"> • Ingresar o cambiar sus datos. • Realizar anuncios de prácticas preprofesionales con información detallada de los requerimientos del cargo. • Aceptar o rechazar el practicante que el usuario administrador le envía. • Asignar fechas de inicio y final de prácticas preprofesionales. • Modificar el informe de control de asistencia autogenerado. • Llenar el formulario para informe de final de práctica preprofesional y generar el informe. • Finalizar el proceso de práctica preprofesional.
Estudiante	<ul style="list-style-type: none"> • Crear cuenta. • Crear su CV llenando un formulario. • Visualizar la información sobre sus prácticas preprofesionales: horas cumplidas y horas por cumplir. • Ver anuncios de prácticas preprofesionales que han sido publicados. • Aplicar a una práctica preprofesional. • Modificar el informe de control de asistencia autogenerado. • Revisar el historial de las prácticas preprofesionales que ha realizado. • Llenar encuesta al final del proceso de práctica preprofesional y generar el documento. • Ver informes asociados al proceso de práctica preprofesional.
Tutor	<ul style="list-style-type: none"> • Ingresar o cambiar sus datos. • Recibir notificaciones al correo electrónico y en la aplicación web cuando el usuario administrador le asigne un estudiante para seguimiento. • Revisar la información del estudiante, de la empresa y de las actividades que el estudiante realiza en la empresa. • Recibir notificaciones al correo electrónico y en la aplicación web cuando la empresa haya llenado el informe de final de periodo. • Recibir recordatorios mediante correo electrónico y en la aplicación web en el caso de que no haya completado algún informe a tiempo. • Llenar el informe de mitad de práctica preprofesional.

En un inicio se planteó que los usuarios reciban una invitación del Administrador Técnico para crear su cuenta, pero se llegó a la conclusión de que se puede afectar a la seguridad

y correcto funcionamiento del prototipo, debido a que los usuarios tendrían libertad de acceder a la creación de cuentas y existen roles que manejan información sensible.

Por esta razón, para los roles: Administrador, Administrador Técnico, Supervisor Empresa y Tutor se crea la cuenta mediante el Administrador Técnico, quien crea los usuarios y estos reciben sus credenciales al correo electrónico registrado.

El único usuario que puede crear su cuenta es el usuario Estudiante, sin embargo, se valida que sus datos se encuentren registrados en la base de datos, esto para controlar que los estudiantes registrados sean de la FIEE únicamente.

Todos los usuarios tendrán la posibilidad de cambiar su contraseña o recuperarla en caso de haberla olvidado.

1.3 MARCO TEÓRICO

En esta sección se exponen los conceptos principales que representan la base teórica del desarrollo del prototipo de EXPAPP. En el apartado 1.3.1 se detallan los conceptos relacionados al *frontend* de EXPAPP, los que incluyen los principales elementos de Angular, el *framework* usado para el desarrollo del prototipo. En la sección 1.3.2 se explican los elementos de Firebase que se usaron para formar el *backend* de EXPAPP.

1.3.1 FRONTEND

El *frontend* en una aplicación web es todo lo que el usuario ve y con lo que interactúa a través de un navegador web. Para crear el *frontend* de una aplicación se combinan tres principales elementos; los dos primeros son dos lenguajes de maquetación y el último es un lenguaje de programación [1]:

- HTML (*HyperText Markup Language*): se lo utiliza para establecer la estructura y el contenido básico de la página.
- CSS (*Cascading Style Sheets*): se lo utiliza para definir cómo se presentan los elementos HTML en términos de diseño y disposición.
- JavaScript: utilizado para definir la lógica de la aplicación y hacer que los sitios web sean interactivos. Recibe solicitudes de los usuarios y las envía al *backend*.

Cuando la página web se carga, el navegador lee el contenido del archivo o archivos HTML, incluyendo CSS y JavaScript, los procesa para generar un esquema virtual de los elementos que componen la página web; el esquema tiene la apariencia de un árbol de objetos y es llamado Modelo de Objetos de Documento o DOM, en la sección 1.3.1.1 se profundiza este concepto.

Adicionalmente, en el desarrollo del *frontend* se puede hacer uso de *frameworks* y/o librerías para facilitar la programación o añadir funcionalidad adicional para necesidades específicas.

1.3.1.1 Modelo de objetos de documento

El Modelo de Objetos de Documento, o DOM por sus siglas en inglés, es generado por el navegador web a partir de archivos HTML y CSS, y modificaciones realizadas en JavaScript; no es mostrado en ninguna parte, sino que sirve para ofrecer propiedades, métodos y eventos para crear una página web organizada en objetos y poder modificarla [2].

El DOM es un estándar definido por el W3C (World Wide Web Consortium) y dice [3]:

"El Modelo de Objetos de Documento (DOM) del W3C es una plataforma e interfaz independiente del lenguaje que permite a los programas y scripts acceder y actualizar dinámicamente el contenido, la estructura y el estilo de un documento."

El estándar DOM W3C está separado en tres partes: 1. Núcleo DOM que define un modelo estándar para todo tipo de documento estructurado, 2. DOM XML y 3. DOM HTML. El DOM HTML es el más común y define: los elementos HTML como objetos, las propiedades de todos los elementos HTML, los métodos para acceder a todos los elementos HTML, y los eventos para todos los elementos HTML [2].

En otras palabras, el DOM HTML es un estándar que indica cómo obtener, cambiar, agregar o eliminar elementos HTML.

1.3.1.2 Framework

De manera general un marco de trabajo o *framework* es una estructura compuesta por componentes que pueden ser personalizados o intercambiados para desarrollar una aplicación de cualquier tipo. Es decir, es una aplicación genérica incompleta que se configura para formar una aplicación concreta [4].

Un *framework* permite el uso de bibliotecas y acceso a soporte, además, mediante la reutilización de código ya existente facilita y acelera el desarrollo de aplicaciones, promoviendo buenas prácticas de desarrollo [5].

En conclusión, un *framework* no es más que un conjunto completo de herramientas para formar y organizar un sitio web o una aplicación web [1].

1.3.1.3 Programación Reactiva

Conocida como *Reactive Programming* es un paradigma de programación que está basado en el manejo de datos mediante *streaming* y en la propagación de cambios en tiempo real y de forma asíncrona. Su arquitectura está basada en Observables [6].

Un Observable es un objeto que tiene un método *subscribe*, que tiene como parámetro un observador y retorna una suscripción. Cuando el observador se suscribe al Observable emite un valor y el observador es notificado [7].

También se puede describir un Observable como algo que se quiere observar mediante un evento o valor futuro, por ejemplo, una llamada HTTP. Los Observables, a diferencia de

las funciones, son capaces de retornar un valor múltiples veces y soportan múltiples tipos de datos [8].

Para la implementación de programación reactiva existe una API llamada *Reactive extensions* (Rx) que provee documentación para implementar *reactive programming*, para poder integrarla en la tecnología Angular y/o JavaScript la librería es RxJS [6]. RxJS es una librería realizada con funciones puras y su principal función es permitir producir, consumir y manipular *streams* de datos [6].

1.3.1.4 Angular

Angular es una plataforma y *framework* de código abierto impulsado por Google, se lo utiliza para el diseño y desarrollo de aplicaciones web de tipo SPA (*Single Page Application*). En una aplicación de tipo SPA se realiza la carga de la página completa al inicio, es decir, cuando se abre la página por primera vez el proceso de carga tarda más, sin embargo, la navegación posterior es instantánea [9].

El lenguaje de programación utilizado en Angular es TypeScript, el cual es un lenguaje de código abierto que se fundamenta en la programación orientada a objetos. Fue creado en el año 2012 por Microsoft y es una extensión de JavaScript. Al compilar TypeScript se genera código JavaScript para que el navegador pueda interpretarlo; a este proceso se lo conoce como transpilar, es decir, generar código en un lenguaje de programación a partir de otro [9].

1.3.1.5 Angular CLI

Angular CLI (*Command-Line Interface*) es una interfaz de línea de comandos que ayuda al desarrollador a inicializar, administrar y modificar proyectos de Angular. Además, también permite construir y probar las aplicaciones de manera local mediante un servidor de desarrollo. La URL por defecto en la cual se prueba la aplicación es `http://localhost:4200/`; si se produce un cambio en los archivos fuente, el servidor automáticamente reconstruye la aplicación y recarga la página [10].

Mediante la línea de comandos es posible crear el esqueleto de una aplicación, es decir, una aplicación básica que contiene los archivos principales de una aplicación Angular, en la cual se crearán nuevos elementos para otorgarle la funcionalidad deseada [11].

1.3.1.6 Arquitectura de Angular

Los elementos que componen este *framework* son: módulos, componentes, servicios y directivas. A continuación, se describe a detalle cada uno de los elementos de Angular:

1.3.1.6.1 Módulos o *NgModules*

Este elemento proporciona un contexto o dominio de aplicación para componentes y servicios, el propósito de un módulo es declarar todos los elementos creados y agruparlos. Los módulos permiten organizar el código para gestionar aplicaciones complejas y que puedan ser reutilizadas [9].

Toda aplicación tiene un módulo raíz llamado *AppModule*, el cual es el encargado de proveer el mecanismo de arranque que inicia la aplicación web; también existen módulos nativos, como por ejemplo el *HttpClientModule* [12].

Dentro de los módulos se pueden manejar los elementos de cuatro diferentes maneras [13]:

- **Declaraciones (*Declarations*):** principalmente son componentes que serán utilizados en el ámbito local. Es decir, que los componentes declarados se pueden utilizar únicamente en el módulo actual (Ejemplo en Código 1.1).
- **Exportaciones (*Exports*):** se emplean cuando se requiere usar los componentes en otros módulos (Ejemplo en Código 1.1).

```
1  import { NgModule } from '@angular/core';
2
3  import { SomeComponent } from './some.component';
4  import { SomeDirective } from './some.directive';
5
6  @NgModule({
7    declarations: [SomeComponent, SomeDirective],
8    exports: [SomeComponent, SomeDirective]
9  })
10
11  export class SomeModule{}
```

Código 1.1. Ejemplo de declaraciones y exportaciones en un módulo de Angular (líneas 7 y 8) [13].

- **Proveedores (*Providers*):** se usa para manejar el sistema de inyección de dependencias, es decir, los servicios que requieren ser utilizados. Si un servicio se registra en el *AppModule* podrá usarse en toda la aplicación (Ejemplo en Código 1.2) [14].


```

1  import { NgModule } from '@angular/core';
2
3  import { SomeComponent } from './some.component';
4  import { SomeDirective } from './some.directive';
5  import { SomeService } from './shared/some.service';
6
7  @NgModule({
8    declarations: [SomeComponent, SomeDirective],
9    providers: [SomeService]
10 })
11
12 export class SomeModule{}

```

Código 1.2. Ejemplo de declaraciones y proveedores en un módulo de Angular (línea 9) [13].

- **Importaciones (*Imports*):** se colocan los módulos que contienen componentes que se requiere utilizar, pero no se encuentran dentro del módulo actual (Ejemplo en Código 1.3).

```

1  import { NgModule } from '@angular/core';
2
3  import { CommonModule } from '@angular/common';
4  import { HttpClientModule } from '@angular/common/http';
5  import { OtherModule } from './other.module';
6
7  @NgModule({
8    imports: [CommonModule, HttpClientModule, OtherModule]
9  })
10
11 export class SomeModule{}

```

Código 1.3. Ejemplo de importaciones en módulos de Angular (línea 8) [13].

1.3.1.6.2 Componentes

Un componente es una porción de la aplicación web y debe existir al menos uno, el cual es el componente raíz que conecta una jerarquía de componentes con el DOM. Un componente es una clase que contiene datos y lógica para cumplir tareas específicas [15].

Un componente está compuesto por [15]:

- Un archivo HTML el cual declara lo que se presenta en la página.
- Una clase TypeScript que define el comportamiento.
- Un selector CSS que define cómo se usará el componente.
- Un archivo CSS que define el estilo del archivo HTML.

1.3.1.6.3 Servicios

Es un elemento que contiene datos o lógica que no están asociados a una vista, pero se requiere utilizar en diferentes componentes. Facilita la reutilización de código y proveen funcionalidad extra a los componentes. [9]

Los servicios se inyectan en los componentes como dependencias, lo que permite el manejo de clases de manera eficiente y ligera. La inyección se la realiza en el constructor del componente. [12]

1.3.1.6.4 Directivas

Son instrucciones que se representan como un atributo más de una etiqueta HTML [9]. Permiten a los desarrolladores asignar comportamientos especiales al DOM, lo que permite crear contenido dinámico extendiendo la funcionalidad de HTML. [16]

Las directivas más comunes en Angular son: *ngFor* que permite repetir elementos mediante la iteración de una lista; y *ngIf* que muestra u oculta elementos del DOM con respecto a una condición dada [17]. En la Figura 1.3 se tiene un ejemplo de uso de la directiva *ngFor*, en la Figura 1.3a se tiene el código en donde se especifica que un elemento HTML, en este caso un botón, va a iterar en un *array* de 5 números, en la Figura 1.3b se tiene el resultado.

<pre><button *ngFor="let item of [1,2,3,4,5]"> Botón # {{item}} </button></pre>	<table border="1"><tr><td>Botón # 1</td></tr><tr><td>Botón # 2</td></tr><tr><td>Botón # 3</td></tr><tr><td>Botón # 4</td></tr><tr><td>Botón # 5</td></tr></table>	Botón # 1	Botón # 2	Botón # 3	Botón # 4	Botón # 5
Botón # 1						
Botón # 2						
Botón # 3						
Botón # 4						
Botón # 5						

Figura 1.3. a) HTML usando directiva *ngFor*. b) Resultado del uso de la directiva *ngFor*.

Todos los elementos mencionados anteriormente interactúan entre sí en base a una arquitectura MVC (Modelo - Vista - Controlador) modificada, debido a que en Angular el modelo tiene una relación fuerte con la Vista, esto quiere decir que desde la Vista se puede realizar modificaciones en el Modelo y desde el Modelo se puede modificar la Vista [18].

En Angular la arquitectura MCV cambia de nombre a MVVM (Modelo - Vista - Vista - Modelo) [19]. A continuación, se explica qué elementos de Angular participan en esta arquitectura:

- Modelo: gestionan los datos de la aplicación [20]. Por lo tanto, se incluye: interfaces y servicios, que brindan atributos a los datos y la funcionalidad respectivamente. [19]
- Vista: son las páginas en donde el usuario interactúa con la aplicación [20]. Se integra por archivos HTML y hojas de estilo CSS [19].
- Vista Modelo: es el reemplazo del Controlador, se compone por los archivos .ts, que forman parte de los componentes, mediante el cual interactúan el Modelo y la Vista [20].

1.3.1.6.5 Guards

Los *Guards* no son un elemento necesario en la arquitectura de Angular, pero son de gran ayuda para agregarle seguridad a la aplicación web. Son interfaces que una clase puede implementar para tener un control sobre las rutas de la aplicación web. Se controla qué usuario accede a la ruta mediante un valor de retorno (verdadero o falso) [21].

La interfaz más utilizada se llama *CanActivate*, si la clase que la implementa devuelve verdadero la navegación continúa, si devuelve falso se cancela la navegación y se redirige a una página predeterminada por la clase [22].

1.3.1.7 Rutas en Angular

Para la navegación en Angular se tiene un módulo padre llamado por convención *AppRoutingModule*, el cual se dedica al enrutamiento y es importado por *AppModule*. Si se crea un proyecto de Angular mediante la línea de comandos el archivo *app-routing-module.ts* se crea por defecto [23].

En el archivo *app-routing-module.ts* primero se deben importar *RouterModule* y *Routes*, como se observa en la Código 1.4, para tener la funcionalidad de ruteo y posteriormente se importan los componentes necesarios, los cuales especifican a donde se va a navegar [23].

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { HeroesComponent } from './heroes/heroes.component';
```

Código 1.4. Importaciones necesarias en *AppRoutingModule* [23].

En el Código 1.5 se definen las rutas que indican al enrutador qué vista mostrar cuando un usuario hace clic en un enlace o pega una URL en la barra de direcciones del navegador. Una ruta de Angular tiene dos partes principalmente: el *path* que es una cadena que coincide con la URL en la barra de direcciones del navegador y el *component* que es el componente que debe crear el enrutador al navegar a esta ruta [23].

```
const routes: Routes = [  
  { path: 'heroes', component: HeroesComponent }  
];
```

Código 1.5. Configuración de rutas en *AppRoutingModule* [23].

Lo siguiente que se debe realizar es importar *RouterModule* y configurar las rutas en *AppRoutingModule*. Esto se lo realiza como se muestra en el Código 1.6 con el método *forRoot()*, el cual instancia las rutas como raíz y de manera global [24].

```
@NgModule({  
  imports: [RouterModule.forRoot(routes)],  
  exports: [RouterModule]  
})  
export class AppRoutingModule { }
```

Código 1.6. Configuración de *imports* y *exports* de *AppRoutingModule* [23].

Es posible la configuración de rutas hijas con el método *forChild()*, el cual especifica que ya existe un método *forRoute* e instancia las rutas debajo de éste de la siguiente manera: *forRoute/forChild* [24].

En los archivos HTML de los componentes se puede colocar la directiva *router-outlet*, la cual es reemplazada con el elemento que haya sido configurado en la ruta como se muestra en el Código 1.7 [23].

```
<h1>{{title}}</h1>  
<router-outlet></router-outlet>
```

Código 1.7. Uso de directiva *RouterOutlet* [23].

1.3.1.8 Angular Material

Es una librería de componentes web basada en *Material Design*, que es una guía de diseño principalmente para Android, pero que se ha extendido también a aplicaciones web. *Material Design* combina colores, sombras, profundidad y bordes para tener objetos de tres dimensiones [25].

Angular Material contiene bloques de construcción llamados componentes que sirven para crear la interfaz de usuario de manera rápida y amigable con el usuario. Estos bloques pueden ser tablas, botones, barras de navegación, entre otros [26].

Para usar la librería debe instalarse en el proyecto de Angular y posteriormente importar los componentes que se requiere usar en el módulo que se está trabajando. Es recomendable crear un Módulo (archivo de tipo *module.ts*) específicamente para Angular Material, en el cual se importará todos los componentes necesarios, esto mantiene una mejor organización en el proyecto. Una vez hecho esto se puede usar el componente y darle estilo cambiando los colores, tamaño, letra, entre otros.

1.3.2 BACKEND

El *backend* es la parte de la aplicación web que contiene la lógica que es transparente para el usuario. Entre las funciones que se pueden gestionar desde el *backend* están [27]:

- Conexión con bases de datos
- Utilizar librerías de servidores web
- Mantener las páginas web seguras
- Optimizar recursos para desarrollar aplicaciones más ligeras

1.3.2.1 Bases de datos no relacionales

Las bases de datos no relacionales o bases de datos *NoSQL* están diseñadas para modelos de datos variados. Son usadas principalmente en aplicaciones que manejan volúmenes grandes de datos, requieren baja latencia y modelos de datos flexibles [28].

Una de las principales ventajas de utilizar bases de datos no relacionales es la flexibilidad debido a que son ideales para datos semiestructurados y no estructurados, lo que permite tener un esquema flexible que da paso a un desarrollo rápido y más iterativo. Otra ventaja es que presentan alta escalabilidad pues se tiene un crecimiento horizontal, se añaden clústeres distribuidos de hardware y no se tiene la necesidad de añadir servidores caros [28].

Los datos en una base de datos de tipo NoSQL usan comúnmente los datos como documentos de tipo JSON (*JavaScript Object Notation*), debido a que es un modelo de datos intuitivo y eficiente. El uso de documentos facilita el desarrollo porque la consulta a la base de datos se vuelve más sencilla al usar el mismo formato del modelo que se emplea en la aplicación [28].

1.3.2.2 Firebase

Firebase es una plataforma alojada en la nube que permite el desarrollo de aplicaciones web y móviles (iOS y Android). Ayuda a que el desarrollo sea sencillo y rápido, sin privarse de la calidad requerida. Tiene herramientas variadas que permiten desarrollo, crecimiento, monetización y análisis, lo que implica menor tiempo invertido en el *backend* tanto en desarrollo como en mantenimiento [29].

En la Tabla 1.2 se describen algunas de las herramientas más utilizadas.

1.3.2.3 AngularFire

Es un paquete oficial de Google que tiene como objetivo facilitar la integración de Firebase a un proyecto de Angular. La implementación de Firebase puede resultar complicado para desarrolladores que están acostumbrados a la programación con Angular y esta librería permite inyectar servicios de Firebase a los componentes de la aplicación en base a la programación de Angular [35].

Está basado en objetos de tipo observable mediante la librería RxJS para manejar los *streams* de datos en tiempo real [36].

Tabla 1.2. Herramientas de desarrollo de Firebase.

Nombre	Función
<i>Authentication</i>	Firestore Authentication ayuda a que la creación de sistemas de autenticación sea fácil, rápida y segura. Proporciona inicio de sesión mediante correo electrónico, autenticación telefónica y también mediante plataformas como Google, GitHub, Twitter, Facebook, entre otras [30].
<i>Cloud Firestore</i>	Cloud Firestore es una base de datos <i>NoSQL</i> que almacena los datos en documentos que se encuentran organizados en colecciones y permite actualizaciones en tiempo real. Permite realizar consultas que devuelvan un documento en específico o recuperar todos los documentos de una colección que cumpla con la condición de la búsqueda [31].
<i>Cloud Storage</i>	Esta herramienta permite almacenar en la nube de manera fácil y rápida archivos de usuarios como fotos y videos, a los cuales se podrá acceder posteriormente mediante una URL [32].
<i>Functions</i>	Con Functions de Firebase se pueden crear funciones que activen herramientas de Firebase como <i>Authentication</i> , <i>Firestore</i> , <i>NodeMailer</i> , <i>Cloud Messaging</i> , entre otras. Estas funciones son ejecutadas con Node.js y se ejecutan únicamente cuando ocurre un evento específico que se encuentra en observación y se dispara con una condición dada [33].
<i>Hosting</i>	Firestore permite alojar una aplicación móvil o web para que puedan ser visitados desde cualquier dispositivo que se encuentre conectado a Internet. La aplicación se publica rápidamente y se configura de manera automática un certificado SSL [34].

2 METODOLOGÍA

Este capítulo está conformado por tres secciones. En la sección 2.1 se tiene el levantamiento de la información, los resultados de las entrevistas realizadas y los requerimientos obtenidos tras el análisis de las entrevistas. En la sección 2.2 se tiene el diseño de la aplicación, incluye diagramas de la aplicación y diseño de cada uno de los módulos de los roles de usuario. En la sección 2.3 se explica detalladamente cómo se realizó la implementación del prototipo.

2.1 LEVANTAMIENTO DE INFORMACIÓN

2.1.1 LEVANTAMIENTO DEL PROCESO

En esta sección se realiza un análisis sobre las entrevistas realizadas a los involucrados en el proceso de prácticas preprofesionales en la facultad: Decano de la FIEE, estudiantes y empresas. Mediante las entrevistas se busca la obtención de los requerimientos del prototipo de la aplicación web.

2.1.1.1 Entrevista al Decano de la Facultad de Ingeniería Eléctrica y Electrónica

La entrevista al Decano de la FIEE se la realizó con el objetivo de determinar cómo se lleva a cabo el proceso de prácticas preprofesionales en la actualidad, qué papel cumplen los miembros de la Comisión de Prácticas Preprofesionales (CPP), cómo se manejan los convenios con empresas, entre otros aspectos importantes.

En la Tabla 2.1 se presentan las respuestas obtenidas tras la entrevista al Decano de la FIEE, MSc. Fabio González.

A partir de la Tabla 2.1 se concluye que en la FIEE se requiere brindar mayor facilidad de encontrar prácticas preprofesionales a los estudiantes, brindándoles la oportunidad de realizar prácticas preprofesionales mediante convenios con empresas y de esta manera asegurar la culminación de las horas que son requisito para la titulación.

Actualmente, las empresas con convenio recurren a correos electrónicos o llamadas telefónicas para solicitar practicantes. Este proceso puede tener ciertas demoras debido a que primero se comunican con autoridades de la FIEE, quienes deben buscar estudiantes y seleccionarlos para la práctica.

Tabla 2.1. Resultados de entrevista al Decano de la FIEE.

Pregunta	Análisis
<i>¿Es usted miembro de la Comisión de Prácticas Preprofesionales de la FIEE?</i>	El Decano de la FIEE indicó que no es miembro de la CPP, pues, en la Guía de Prácticas Preprofesionales aprobada por el Consejo de Docencia se lo prohíbe; esto debido a que el Decano de la facultad debe aprobar ciertos trámites que se realiza en la CPP.
<i>¿Cuáles son las principales actividades que realiza un miembro de la Comisión de Prácticas Preprofesionales?</i>	El Decano de la FIEE es el encargado de la firma de cartas de prácticas preprofesionales, en el periodo de abril a octubre del 2020 (6 meses) se emitió aproximadamente 80 cartas.
<i>¿Cuántas cartas de prácticas preprofesionales dirigidas a empresas firma aproximadamente por semestre?</i>	El Decano de la FIEE es el encargado de la firma de cartas de prácticas preprofesionales, en el periodo de abril a octubre del 2020 (6 meses) se emitió aproximadamente 80 cartas.
<i>¿Con cuántas empresas se tiene convenio en la FIEE?</i>	El Decano de la FIEE indicó que hasta enero del 2020 se tenía 28 empresas con convenio en la FIEE, sin embargo, es probable que dicho valor haya aumentado.
<i>¿Cuál es el proceso que se sigue cuando una empresa con la que se tiene convenio solicita practicantes?</i>	La empresa realiza la solicitud de practicantes y en la FIEE se realiza un proceso de selección en base a los requerimientos del puesto. Se notifica a la empresa sobre el o los estudiantes seleccionados. En este caso ya no se requiere la carta de solicitud dirigida a la empresa, porque ya existe un convenio.
<i>¿Con qué frecuencia una empresa con la que se tiene convenio solicita practicantes? ¿Por cuál medio lo realiza? Ejemplo: Correo, llamada telefónica.</i>	El Decano de la FIEE indicó que existen empresas con las que se tiene una relación más fluida y éstas solicitan estudiantes para prácticas preprofesionales cada 6 meses aproximadamente, conociendo cuándo los estudiantes están de vacaciones. Las empresas realizan el pedido mediante correo electrónico, QUIPUX Gubernamental (Empresas Públicas) y llamada telefónica.
<i>¿Qué requisitos se toma en cuenta para seleccionar un estudiante para una práctica preprofesional?</i>	El Decano de la FIEE afirmó que se toma en cuenta dos principales aspectos. El primero es que el estudiante cumpla con las habilidades y conocimientos que la empresa solicitó para el puesto.

	<p>El segundo aspecto considerado es el número de créditos que tiene aprobados el estudiante, se da prioridad a estudiantes egresados que no poseen prácticas preprofesionales realizadas.</p>
<p><i>¿Cree que el proceso actual que siguen los estudiantes para realizar prácticas preprofesionales puede mejorarse? ¿De qué manera?</i></p>	<p>El Decano de la FIEE opina que sería importante conseguir más convenios para brindar a los estudiantes más oportunidades de conseguir prácticas preprofesionales y que no tengan que conseguir prácticas preprofesionales de manera independiente.</p> <p>Otro aspecto que considera importante mejorar es el cumplimiento de plazos en la entrega de informes por demoras de los involucrados en el proceso.</p> <p>Por último, los informes entregados suelen tener errores de formato y ortografía o datos erróneos, por lo que se requiere volver a llenar los informes, lo que involucra la movilización del estudiante hasta la empresa.</p>
<p><i>¿Considera que la facultad está lista o estaría dispuesta a una transformación digital o digitalización del proceso de prácticas preprofesionales? ¿Por qué?</i></p>	<p>El Decano de la FIEE considera importante contar con una herramienta que permita facilitar y acelerar los procesos involucrados en las prácticas preprofesionales; y mediante la misma se pueda dar a conocer a los estudiantes oportunidades de prácticas preprofesionales.</p> <p>Otro aspecto importante es que las empresas lo podrían usar como una bolsa de empleo enfocado a prácticas preprofesionales.</p> <p>Finalmente, se tendría un mejor control del proceso y se tendría la información involucrada en un repositorio, lo cual la hace accesible para los usuarios interesados.</p>

El registro de culminación de prácticas preprofesionales puede llegar a retrasarse, debido a que el supervisor de la empresa tiene que llenar y firmar el informe final, después el estudiante entrega el informe al tutor para la firma y debe entregarlo en Subdecanato. En ocasiones los informes tienen formato inadecuado o datos erróneos, por lo que, el estudiante debe movilizarse hasta la empresa para corregir dichos errores.

El Decano de la FIEE considera importante contar con una herramienta que permita la obtención y manejo de prácticas preprofesionales; debido a que se ayudaría desde la FIEE a los estudiantes a encontrar prácticas preprofesionales de manera rápida. Asimismo, las empresas usarían la herramienta como bolsa de prácticas, lo cual podría reducir el tiempo que les toma conseguir practicantes hoy en día.

Por último, el seguimiento de prácticas preprofesionales se daría de manera más controlada y ágil. Los informes y documentos del proceso estarían disponibles en un repositorio, lo que los hace fácilmente accesibles para el personal administrativo, que en el caso de requerir cambios o verificación de la información lo puede realizar rápidamente.

2.1.1.2 Entrevista a estudiantes de la Facultad de Ingeniería Eléctrica y Electrónica

Para determinar el grado de satisfacción de los estudiantes con el proceso actual de prácticas preprofesionales se realizó una entrevista estructurada a diez estudiantes distribuidos en las cuatro carreras de la FIEE. Las entrevistas estuvieron enfocadas a encontrar los principales problemas que tienen los estudiantes durante el proceso de prácticas preprofesionales y sus posibles soluciones.

En la Tabla 2.2 se muestran los resultados obtenidos tras la entrevista a diez estudiantes de la FIEE.

En conclusión, la mayoría de los estudiantes coincide que en la Facultad no existe una adecuada socialización de cómo se lleva a cabo el proceso de prácticas preprofesionales. Por otro lado, a los estudiantes les resulta difícil encontrar empresas o instituciones dónde puedan realizar sus prácticas preprofesionales.

Una alternativa para solucionar los problemas antes mencionados es crear un sitio web en donde los estudiantes puedan encontrar la información relacionada al proceso de prácticas preprofesionales, además, puedan conseguir un lugar en dónde realizar prácticas y darles un seguimiento rápido.

Tabla 2.2. Resultados de entrevistas a estudiantes de la FIEE.

Pregunta	Análisis
<i>¿En qué semestre estás? ¿Cuántos créditos aprobados tienes?</i>	El 30% de los estudiantes entrevistados se encuentran egresados y el 60% en último semestre (novenio).
<i>¿Ya has realizado prácticas preprofesionales? ¿En qué semestre fue tu primera práctica preprofesional?</i>	El 10% de los estudiantes entrevistados aún no ha realizado prácticas preprofesionales y ya ha egresado. El 90% ya ha empezado sus prácticas. Los estudiantes generalmente empiezan sus prácticas preprofesionales en el sexto semestre.
<i>¿Te ha resultado difícil conseguir dichas prácticas preprofesionales? ¿Por qué?</i>	El 80% de estudiantes entrevistados les ha resultado difícil conseguir prácticas preprofesionales. El 20% consiguió fácilmente práctica preprofesional con la ayuda de un contacto en alguna empresa.
<i>Crear tu curriculum vitae (CV) es un requisito para aplicar a una práctica preprofesional. ¿Te ha resultado difícil crear tu CV?</i>	El 90% de los estudiantes entrevistados opinaron que les resultó difícil elaborar su CV. Las principales razones son: el formato, qué información poner, piensan que no tienen nada que poner porque no han trabajado.
<i>De los siguientes pasos del proceso de prácticas preprofesionales, ¿cuál crees que es el más complicado o se demora más?</i> <ul style="list-style-type: none"> - Buscar prácticas preprofesionales - La carta para empresa - Asignación de tutor - Informes 	El 80% de estudiantes opina que el paso más complicado es buscar dónde realizar prácticas preprofesionales.
<i>¿Cuántos días aproximadamente se demora la asignación del tutor?</i>	En promedio la asignación del tutor tarda dos semanas desde que se entrega el oficio.
<i>¿Una vez terminada la práctica preprofesional cuánto tiempo transcurre aproximadamente hasta que se registren las horas en el SAEw?</i>	El 100% de los estudiantes entrevistados coincide que este tiempo depende de cuánto tiempo ellos tardan en entregar los informes firmados por el supervisor y tutor de la práctica

	<p>preprofesional al subdecanato para su posterior aceptación y registro en el sistema SAEw.</p> <p>Sin embargo, una de las razones por la que se incurre en demora en el registro de las horas es la forma en la que se ha llenado o respondido al reporte.</p> <p>Los estudiantes expresaron que en algunas ocasiones han tenido que solicitar por segunda vez el llenado del reporte final de la práctica preprofesional para corregir el formato en que se han contestado las preguntas, por ejemplo, fechas, nombre del estudiante, entre otros.</p>
<p><i>¿Has usado plataformas como LinkedIn, Multitrabajos, etc. para buscar prácticas preprofesionales? ¿Han sido efectivas?</i></p>	<p>Únicamente el 20% de los diez estudiantes han buscado prácticas preprofesionales en alguna de estas plataformas, sin embargo, no han tenido éxito. Opinan que no existen anuncios para prácticas preprofesionales.</p>
<p><i>¿Crees que en la facultad se debería brindar acceso a prácticas preprofesionales? ¿Qué métodos serían útiles? Ej. Sitio web.</i></p>	<p>El 100% de los estudiantes entrevistados opinaron que en la Facultad se debería brindar ayuda a los estudiantes para que encuentren prácticas preprofesionales.</p> <p>La mejor opción para brindar este servicio es crear un sitio web exclusivamente para prácticas preprofesionales y que en la Facultad se cree una oficina que maneje la página web y los convenios con las empresas. Otra opción es que los estudiantes reciban anuncios mediante el correo institucional.</p>

2.1.1.3 Entrevista a empresas

Para determinar cuál es el proceso que siguen las empresas al momento de buscar practicantes en la actualidad y los problemas que este proceso representa, se realizó entrevistas a dos personas encargadas de recursos humanos en dos empresas diferentes.

La primera empresa, de ahora en adelante **Empresa A**, es una multinacional de manufacturas, en esta empresa contratan estudiantes de todo tipo en su mayoría contratan estudiantes de negocios o carreras afines, sin embargo, también requieren practicantes de

ingeniería en el área de sistemas para el desarrollo de sus páginas web o manejo de la intranet.

La segunda empresa, de ahora en adelante **Empresa B**, es una empresa enfocada a dar asesoría de ciberseguridad a otras empresas como bancos o empresas que requieran un alto nivel de seguridad en su red. En esta empresa únicamente contratan practicantes de ingeniería electrónica y redes o sistemas.

En la Tabla 2.3 se tienen los resultados de dichas entrevistas.

Tabla 2.3. Resultados de entrevistas a miembros de dos empresas.

Pregunta	Análisis
<i>¿Qué método utilizan para conseguir practicantes?</i>	Los principales métodos son contactarse directamente con el decano de la facultad o solicitar a amistades que trabajen en las universidades para que les ayuden a conseguir practicantes.
<i>¿Han tenido problemas con la búsqueda de practicantes?</i>	El 100% de los entrevistados presentan problemas en la búsqueda de practicantes porque muchas veces encuentran estudiantes que encajan en el perfil, pero sus horarios no.
<i>¿Han usado plataformas como LinkedIn, Multitabajos, etc. para buscar practicantes? ¿Han sido efectivas?</i>	El 100% de los entrevistados ha usado estas plataformas, pero no para la búsqueda de practicantes.
<p>Qué cree que tarda más tiempo:</p> <ul style="list-style-type: none"> - Contactar estudiantes - Revisar CVs - Entrevistas - Llenar informes finales para la universidad 	El 100% de los entrevistados coincide en que el paso más complicado es la búsqueda de estudiantes que se ajusten a su perfil, además, tengan disponibilidad de tiempo para realizar las 6 horas diarias.
<i>¿Ha visto que en el CV los estudiantes coloquen información irrelevante?</i>	La Empresa A contrata estudiantes de negocios y estudiantes de ingeniería y opina que los estudiantes de negocios presentan mejores CVs.

	<p>La Empresa B contrata únicamente estudiantes de ingeniería piensa que los estudiantes generalmente colocan información irrelevante en el CV.</p> <p>Otro factor por considerar es que los estudiantes ponen información extensa de cursos o certificados obtenidos, recomiendan colocar al inicio del CV un resumen con todas las habilidades y conocimientos del estudiante.</p>
<p><i>¿Qué toman en cuenta al momento de revisar el CV de un aspirante?</i></p>	<p>El 100% de entrevistados coincide en que lo más importante es que el candidato cumpla con las habilidades que requieren para el puesto de practicante. Por lo tanto, deberían resaltar este campo en el CV.</p> <p>También consideran que es importante la ubicación (lugar donde viven o estudian) y la formación (semestre en que se encuentran).</p>
<p><i>¿Qué información considera que los estudiantes frecuentemente omiten en su CV y usted considera importante colocar?</i></p>	<p>El 100% de entrevistados opinan que en muchas ocasiones los estudiantes olvidan de poner su ubicación o ciudad en que se encuentran.</p>
<p><i>¿Usaría una página web exclusivamente de una facultad para encontrar practicantes de forma rápida?</i></p>	<p>El 100% de entrevistados coincidieron que sería lo ideal tener una comunicación directa, en lugar de recurrir al decano de la facultad o a sus amistades para conseguir practicantes.</p>
<p><i>¿Cree usted que es beneficioso llenar los informes de final de práctica preprofesional de manera digital (formularios) y recibir recordatorios para hacerlo? ¿O prefiere llenar los informes de manera manual?</i></p>	<p>El 100% de entrevistados opina que llenar los informes de manera digital les parece más sencillo y práctico. Además, las notificaciones serían de gran ayuda porque constantemente olvidan llenar los informes a tiempo.</p>

Actualmente, las empresas recurren a contactos en universidades o se comunican directamente con el decano de la facultad para encontrar practicantes. Una vez que se ponen en contacto con los estudiantes tienen que revisar los CVs de los candidatos hasta

encontrar un estudiante que se ajuste al perfil deseado. Posteriormente, deben llenar los informes de prácticas preprofesionales.

Finalmente, los entrevistados de las empresas A y B coinciden en que las actividades involucradas en el proceso de prácticas preprofesionales implican una elevada inversión de tiempo. Adicionalmente, mostraron interés en utilizar un sitio web de una facultad que les ayude a buscar practicantes y llevar el seguimiento del proceso.

2.1.2 REQUERIMIENTOS

En la sección anterior se analizó las respuestas obtenidas tras entrevistar a los actores involucrados en el proceso de prácticas preprofesionales. A continuación, se describe el flujo de dicho proceso mediante un diagrama de actividad realizado en base a los documentos: “Procedimiento para seguimiento de prácticas preprofesionales” [37] y “Procedimiento para realización de prácticas preprofesionales” [38].

Un diagrama de actividad es un tipo de diagrama, en el cual sus transiciones son producidas por eventos internos. Los nodos representan acciones que suceden secuencial o concurrentemente, es decir, puede haber un canal para cada participante del proceso [39].

En la Figura 2.1 se tienen los actores involucrados en el proceso de prácticas preprofesionales y los pasos que se deben seguir cada uno de ellos. A partir de esta información se ha elaborado el diagrama de contexto de la Aplicación web para la obtención y manejo de prácticas preprofesionales en la FIEE.

Un diagrama de contexto es una representación del sistema como un único proceso y se tiene las interacciones del sistema con el resto de las entidades o actores, que se representan con rectángulos. Se utilizan flechas para indicar el sentido de la interacción [40].

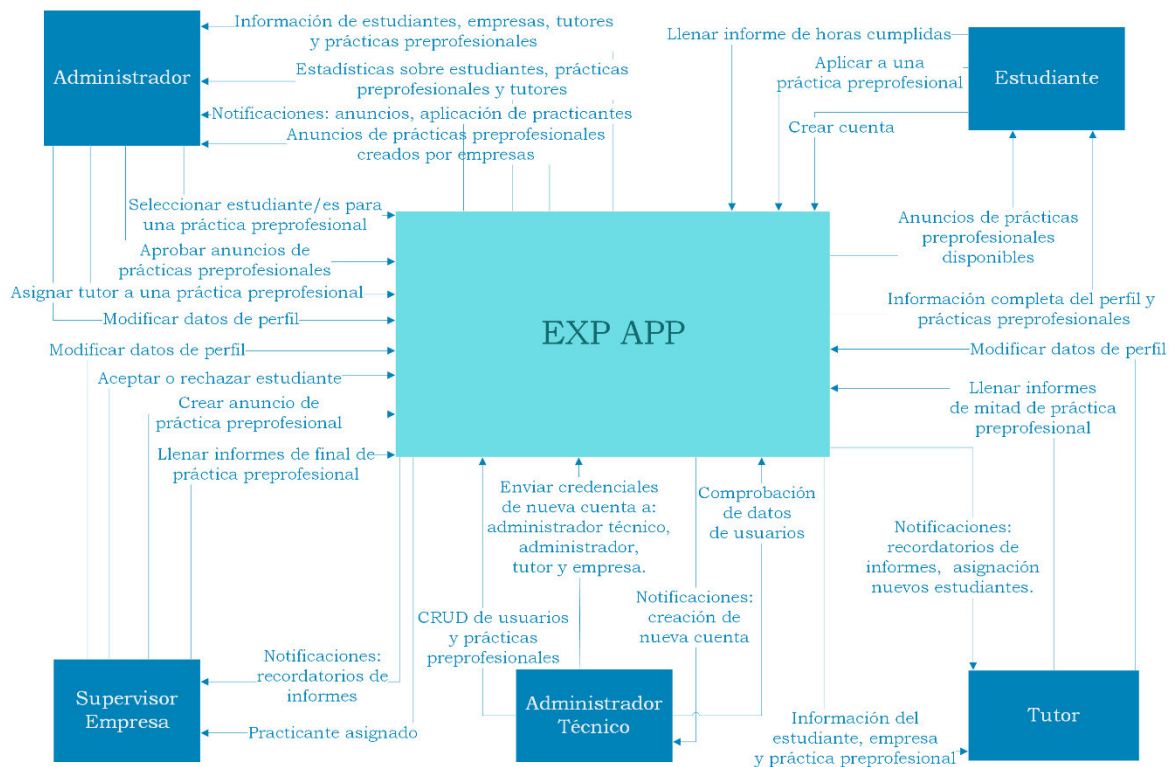


Figura 2.2. Diagrama de contexto de la Aplicación Web para obtención y manejo de prácticas preprofesionales.

Para la aplicación web se han considerado cinco diferentes roles de usuario, como se observa en la Figura 2.2, y se observan las interacciones que existe entre cada uno de ellos y la aplicación web, es decir, lo que cada usuario entrega a la aplicación y lo que recibe de ella. Los diferentes roles se han tomado en base a los actores que intervienen en el proceso de prácticas preprofesionales que se analizó previamente. A continuación, se describe brevemente cada uno de los roles.

- El usuario Administrador involucra a los miembros de la CPP y personal administrativo de la FIEE que forma parte del proceso de prácticas preprofesionales como: Decano/a, Subdecano/a, jefe y secretaria de departamento, entre otros.
- El usuario Administrador Técnico se encarga del CRUD de usuarios, prácticas preprofesionales en la aplicación web. Este usuario también es encargado de verificar la información que los usuarios ingresan, es decir, lleva un control de la base de datos de usuarios.
- El usuario Supervisor Empresa es quien creará anuncios para prácticas preprofesionales, una vez que se le asigne el estudiante empezará un proceso de

práctica preprofesional y mediante la aplicación web podrá llevarse el seguimiento de esta mediante formularios que deben ser llenados con información del estudiante y la práctica preprofesional.

- El usuario Estudiante podrá visualizar anuncios de prácticas preprofesionales que estén relacionadas a su carrera y aplicar al puesto, en el caso de ser seleccionados se da inicio al proceso de práctica preprofesional.
- El usuario Tutor es quien lleva el seguimiento de las Prácticas Preprofesionales que se encuentren en curso. Su principal función es ponerse en contacto con el Supervisor de la Empresa y llenar la información requerida en los informes.

Los usuarios Supervisor Empresa, Tutor y Estudiante son los actores principales en el proceso de práctica preprofesionales. En la Tabla 2.4 se detalla cuáles son los informes requeridos en el proceso y quiénes son los encargados de realizarlos; se ha definido un identificador de cada informe, el cual se utilizará a partir de este punto para nombrar a los informes.

Una vez claros los roles de usuario necesarios y las actividades que cada uno realiza en el proceso de prácticas preprofesionales se procede con el diseño de la aplicación web, lo cual se detalla en la siguiente sección.

Tabla 2.4. Detalles de Informes de seguimiento de práctica preprofesional y responsables en la FIEE.

ID	Nombre	Usuario Asignado	Descripción
Inf1	Encuesta de Prácticas Preprofesionales	Estudiante	Se evalúa la experiencia del estudiante en la Práctica Preprofesional.
Inf2	Formulario de Prácticas Preprofesionales	Tutor y Supervisor de Empresa	Se registran todos los datos del estudiante y empresa, e información de la práctica preprofesional como horarios, horas cumplidas, actividades realizadas y desempeño del estudiante.
Inf3	Informe Mitad Periodo	Tutor	Se registran datos de la empresa e información sobre el desempeño del estudiante en la práctica preprofesional.
Inf4	Informe Final	Tutor	Se registran las materias del Pensum académico que tienen relación con la práctica preprofesional.
Inf5	Registro de Asistencia	Supervisor de Empresa y Estudiante	Se registran los días que el estudiante realizó la práctica preprofesional, con hora de entrada y de salida.

2.1.2.1 DIAGRAMA DE CASOS DE USO

El diagrama de casos de uso permitirá representar el sistema desde el punto de vista de los usuarios y definir los requerimientos funcionales de la aplicación web [39].

En base a la Figura 2.2 se han obtenido las actividades que cada rol de usuario debe realizar en la aplicación web y se han construido los diagramas de casos de uso para cada uno de ellos que se encuentran desde la Figura 2.3 hasta la Figura 2.7.

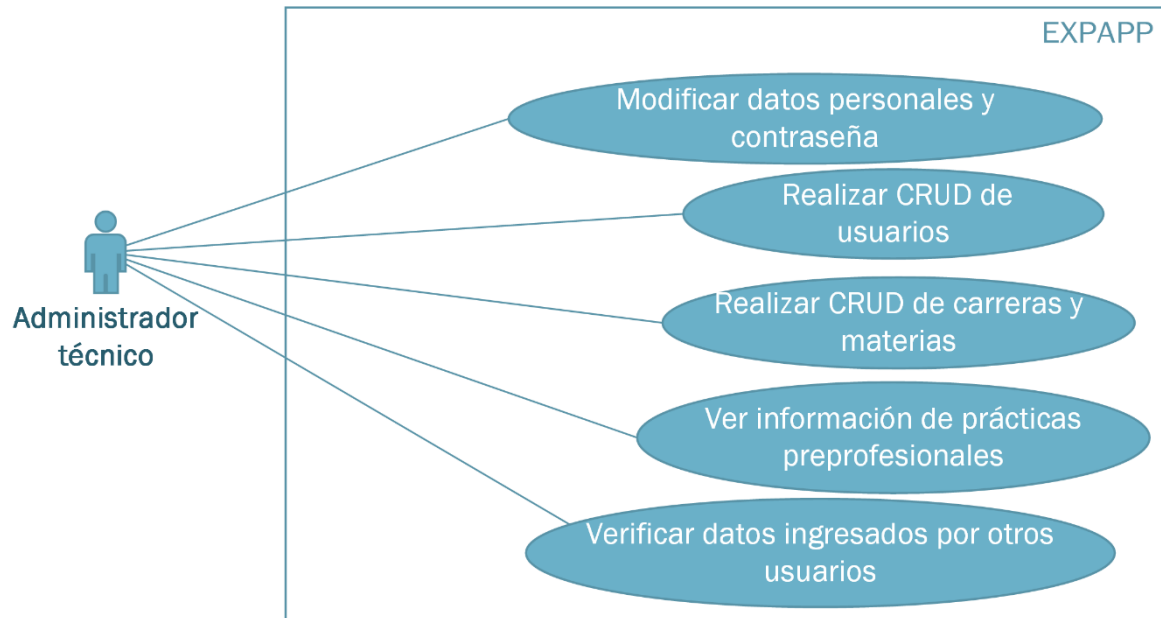


Figura 2.3. Diagrama de casos de uso para el rol Administrador Técnico.

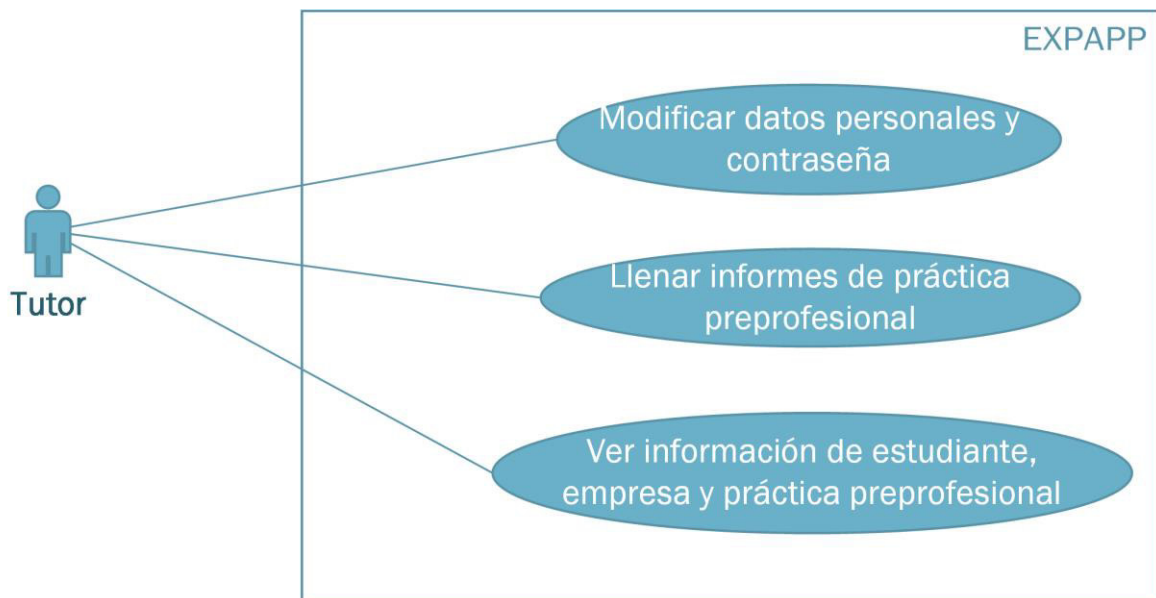


Figura 2.4. Diagrama de casos de uso para el rol Tutor.

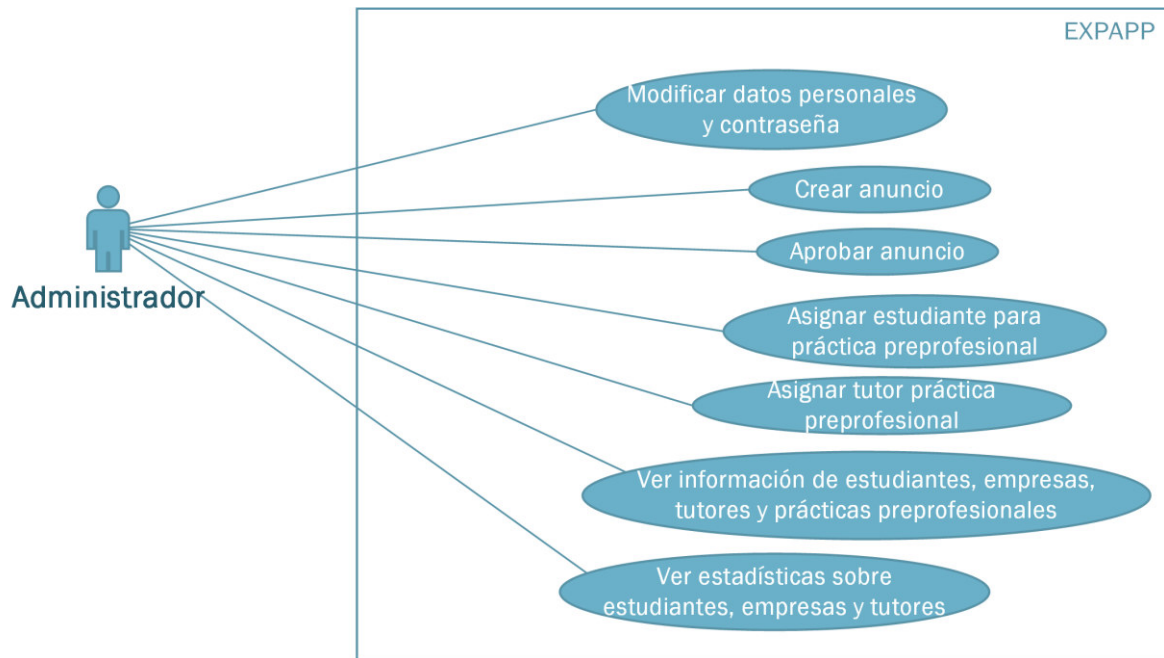


Figura 2.5. Diagrama de casos de uso para el rol Administrador.

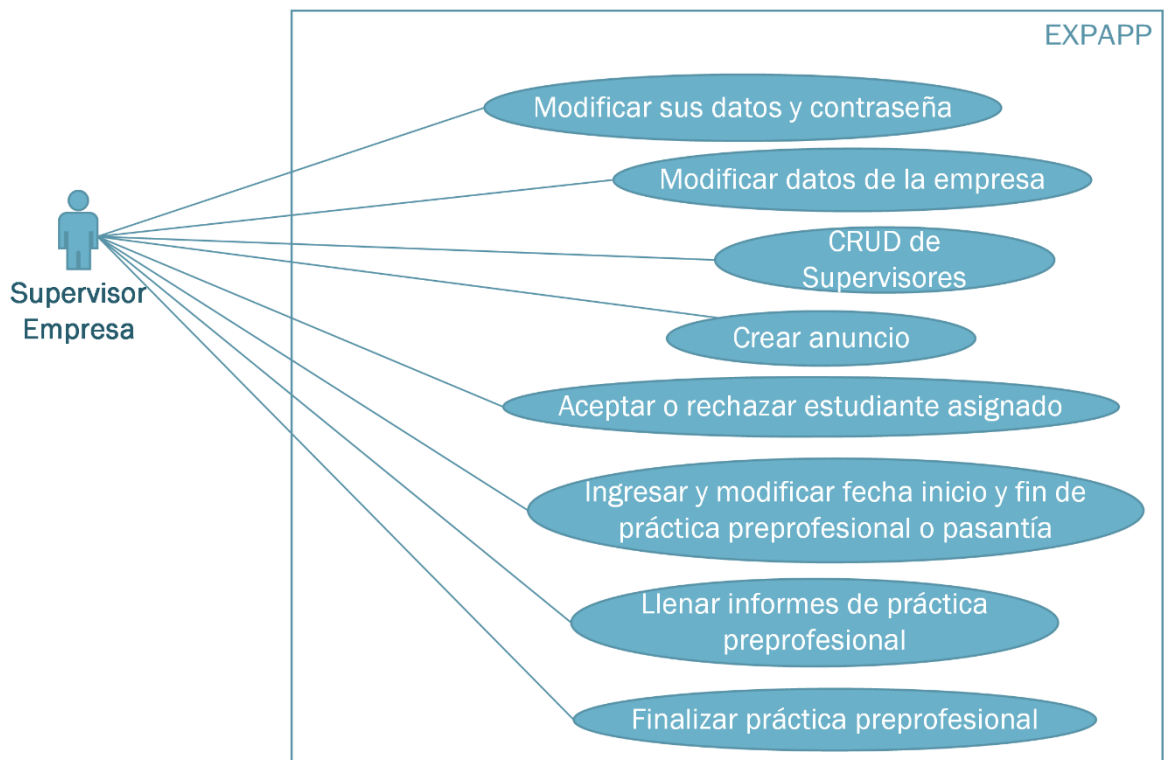


Figura 2.6. Diagrama de casos de uso para el rol Supervisor Empresa.

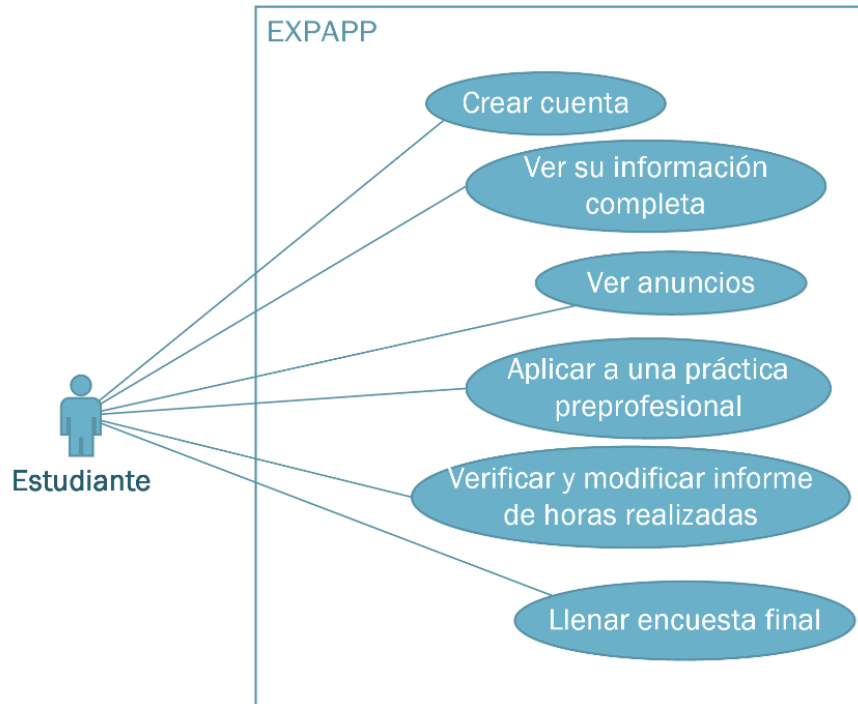


Figura 2.7. Diagrama de casos de uso para el rol Estudiante.

Los requerimientos funcionales son los servicios que brindará la aplicación al usuario y cómo reaccionará a acciones de los usuarios o entradas que sean resultado de la interacción con otros sistemas; en otras palabras, es lo que el sistema hará o no. Por otro lado, los requerimientos no funcionales son las propiedades del sistema, que pueden referirse a rendimiento, disponibilidad, seguridad, entre otros [41].

En la Tabla 2.5 se presentan los requerimientos funcionales mínimos de cada rol de la aplicación web en base a los diagramas de casos de uso presentados anteriormente. Y en la Tabla 2.6 se describen los requerimientos no funcionales del prototipo.

Se ha asignado un identificador a cada requerimiento con el cual será mencionado a lo largo del documento.

Tabla 2.5. Requerimientos funcionales de EXPAPP.

Identificador	Descripción	Rol
Rq1	Modificar datos personales	<ul style="list-style-type: none"> • Administrador Técnico • Administrador • Supervisor Empresa • Tutor
Rq2	Cambiar contraseña	<ul style="list-style-type: none"> • Todos
Rq3	CRUD de usuarios	<ul style="list-style-type: none"> • Administrador Técnico
Rq4	CRUD de carreras y materias	<ul style="list-style-type: none"> • Administrador Técnico
Rq5	Crear anuncio	<ul style="list-style-type: none"> • Administrador • Supervisor Empresa
Rq6	Aprobar anuncio	<ul style="list-style-type: none"> • Administrador
Rq7	Asignar Estudiante para práctica preprofesional	<ul style="list-style-type: none"> • Administrador
Rq8	Asignar Tutor a práctica preprofesional	<ul style="list-style-type: none"> • Administrador
Rq9	Ver información de Estudiantes	<ul style="list-style-type: none"> • Administrador • Supervisor Empresa • Tutor
Rq10	Ver información de Empresas	<ul style="list-style-type: none"> • Administrador • Tutor
Rq11	Ver información de Prácticas Preprofesionales	<ul style="list-style-type: none"> • Administrador • Estudiante • Supervisor Empresa • Tutor
Rq12	Ver estadísticas de Estudiantes, Empresas y Tutores.	<ul style="list-style-type: none"> • Administrador
Rq13	Aceptar o rechazar estudiante asignado a práctica preprofesional.	<ul style="list-style-type: none"> • Supervisor Empresa
Rq14	CRUD de Supervisores Empresa.	<ul style="list-style-type: none"> • Supervisor Empresa
Rq15	Definir fecha de inicio y fin de práctica preprofesional.	<ul style="list-style-type: none"> • Supervisor Empresa
Rq16	Llenar informes de práctica preprofesional.	<ul style="list-style-type: none"> • Estudiante • Supervisor Empresa • Tutor
Rq17	Finalizar práctica preprofesional.	<ul style="list-style-type: none"> • Supervisor Empresa
Rq18	Crear una cuenta	<ul style="list-style-type: none"> • Estudiante
Rq19	Crear CV	<ul style="list-style-type: none"> • Estudiante
Rq20	Ver anuncios activos	<ul style="list-style-type: none"> • Administrador • Supervisor Empresa • Estudiante
Rq21	Postular para una práctica preprofesional	<ul style="list-style-type: none"> • Estudiante

Tabla 2.6. Requerimientos no funcionales de EXPAPP.

Identificador	Descripción	Rol
Rq22	El prototipo debe tener un sistema de autenticación para restringir el acceso únicamente para usuarios registrados.	<ul style="list-style-type: none"> • Todos
Rq23	Se protegerá las rutas propias de cada rol, para que los usuarios tengan acceso únicamente a las interfaces que le corresponden.	<ul style="list-style-type: none"> • Todos
Rq24	Para la creación de usuarios en el prototipo se debe validar que el correo electrónico sea del dominio e pn.edu.ec .	<ul style="list-style-type: none"> • Administrador Técnico • Administrador • Estudiante • Tutor

2.2 DISEÑO DE LA APLICACIÓN

2.2.1 ARQUITECTURA DE LA APLICACIÓN

En la Figura 2.8 se observa cómo va a ser la estructura de la aplicación para que tenga un correcto funcionamiento. El usuario interactúa con el *frontend*, el cual será implementado en el *framework* Angular; como se analizó en la sección 1.3.1.6 la arquitectura de Angular está conformada por Módulos, Componentes, Servicios y Directivas. Los servicios tendrán métodos que permitirán la interacción con el *backend* de la aplicación.

El *backend* será implementado con Firebase de Google y tendrá cuatro principales servicios: base de datos *NoSql* (*Cloud Firestore*), autenticación (*Authentication*), almacenamiento de archivos (*Cloud Storage*) y *Functions*.

Para que la implementación sea de manera ordenada y escalable se realizó el diseño de los módulos para cada rol de usuario, esto con el fin separarlos en contextos diferentes y no sobrecargar el *AppModule*.

También se tendrá un módulo para los componentes que serán compartidos por todos los usuarios. En todos los componentes de la aplicación se puede inyectar servicios para aumentar la funcionalidad y tener funciones de manera accesible para todos los usuarios. En la Figura 2.9 se presenta un diagrama del diseño de los módulos de cada rol y el módulo

compartido por todos ellos, además, se presenta cuatro servicios necesarios en la aplicación.

Cada usuario tendrá un archivo de tipo *routing.module* que permite el manejo de rutas para la navegación y un conjunto de componentes según los requerimientos de cada rol. El módulo de rutas padres es el *app.routing.module* en el cual inyecta un *guard* para proteger las rutas de los usuarios.

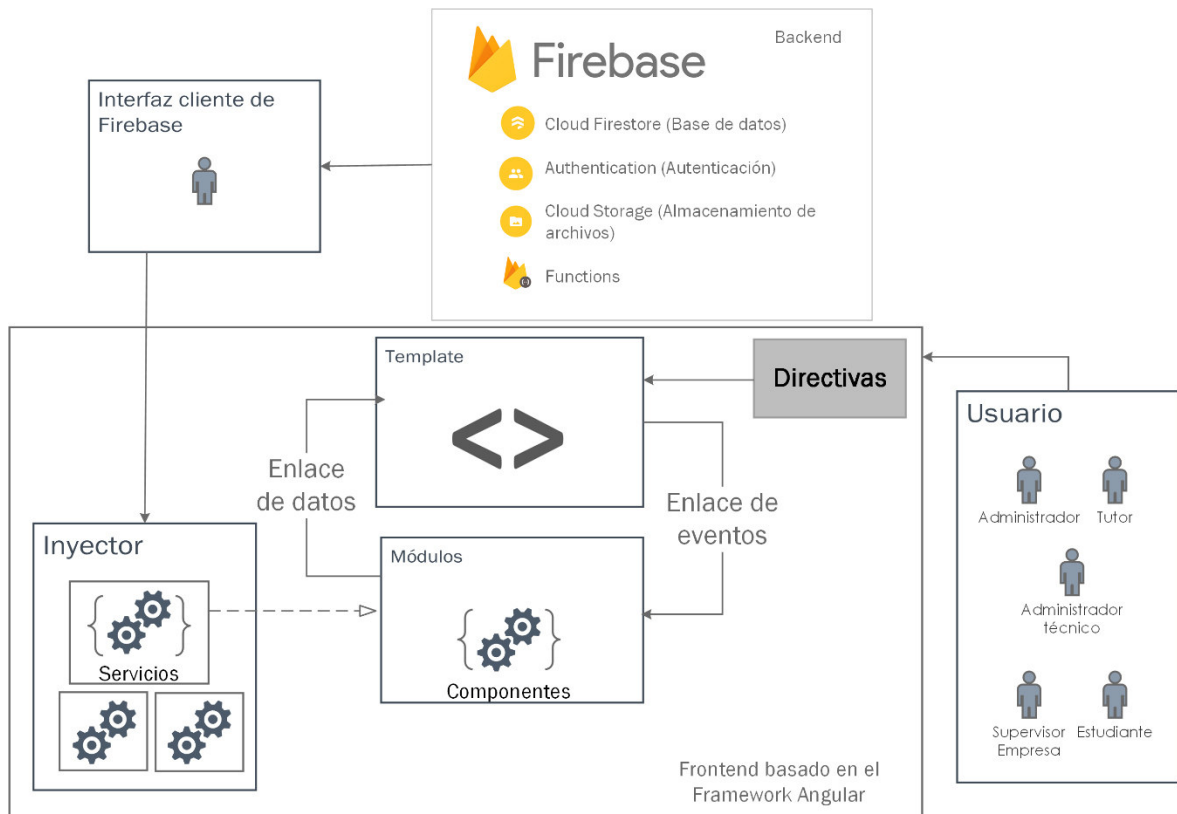


Figura 2.8. Diagrama de arquitectura de EXPAPP.

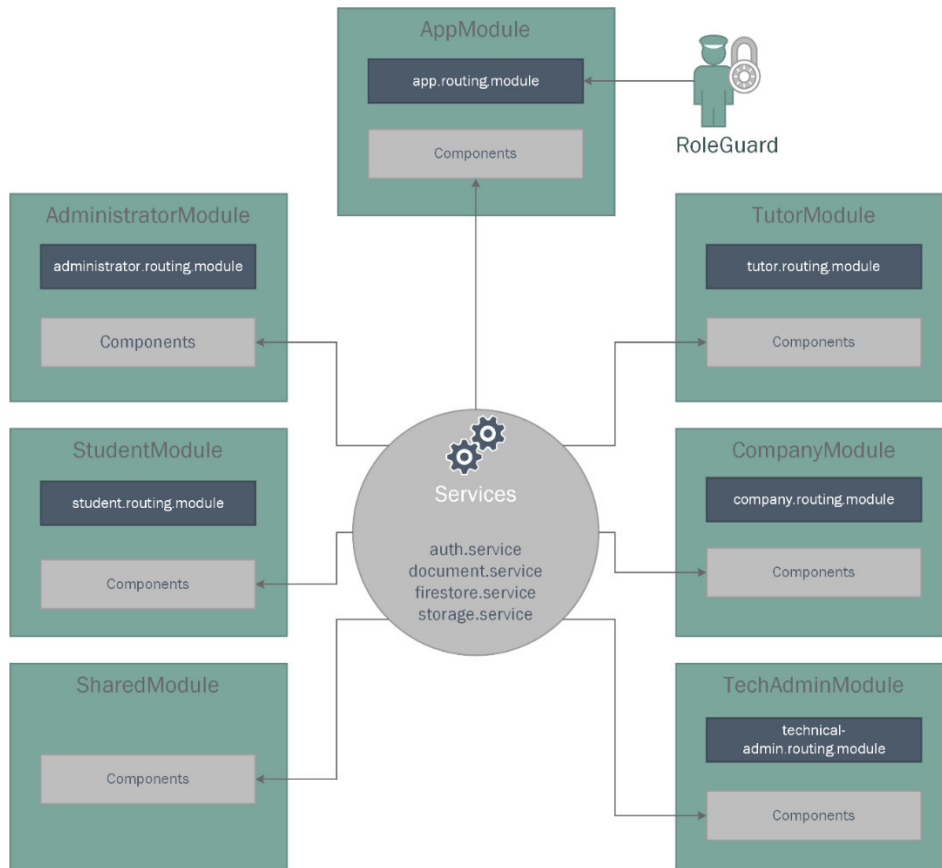


Figura 2.9. Diseño de los módulos de la aplicación web.

2.2.2 ESQUEMA DE DATOS

De acuerdo con la arquitectura de la aplicación web la base de datos es de tipo *NoSQL* al utilizar como motor de base de datos *Cloud Firestore* de Firebase que trabaja con colecciones de documentos.

La base de datos es flexible y cada documento puede tener diferentes atributos, aunque pertenezcan a la misma colección. Sin embargo, para que los datos tengan consistencia y sigan un mismo patrón se ha elaborado un diagrama de modelos que se encuentra en la Figura 2.10, los cuales darán la estructura que los diferentes documentos tendrán y para conocer cómo se relacionan entre sí, con lo cual no hace falta trabajar con un diagrama de clases.

2.2.3 DIAGRAMA DE ESTADOS

Una vez que se ha definidos los modelos que se va a tener en la aplicación, se toman Anuncio y Práctica Preprofesional para analizar su comportamiento y su ciclo de vida mediante un diagrama de estados. El ciclo de estos dos objetos describe el proceso de prácticas preprofesionales en la FIEE en EXPAPP.

Se colocó los estados de Anuncio y Práctica Preprofesional en un mismo diagrama porque dependen uno del otro para continuar con el proceso.

En el diagrama de la Figura 2.11 cada rectángulo representa el estado en que se encuentra el objeto y una flecha representa un cambio de estado, sobre la flecha se señala el rol de usuario y la acción que debe realizar para que se genere el cambio de estado.

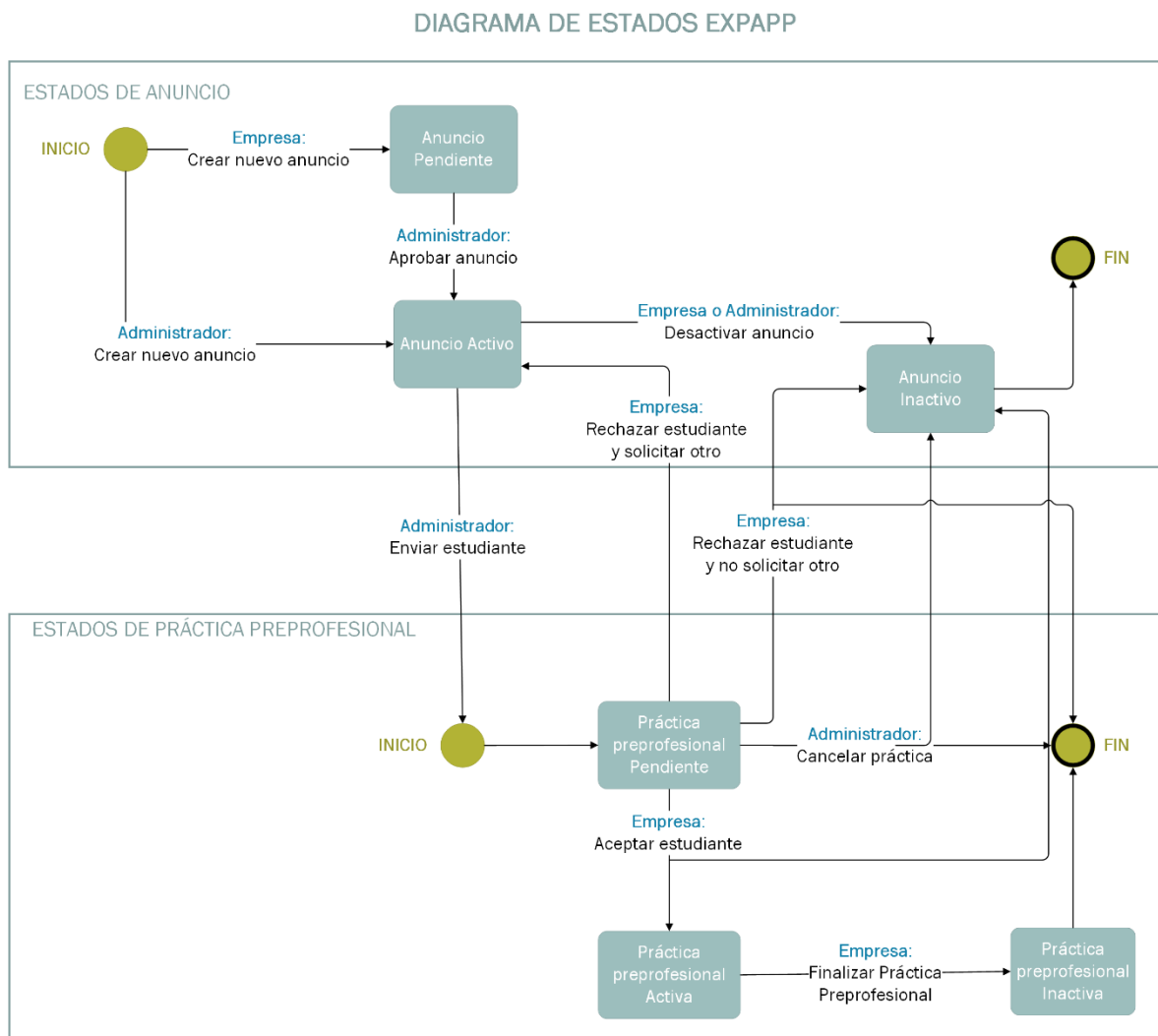


Figura 2.11. Diagrama de estados de EXPAPP.

2.2.4 DISEÑO DE LA INTERFAZ GRÁFICA

En base a los requerimientos funcionales de la Tabla 2.5 se realizó un boceto de las interfaces de usuario para cada rol de usuario con la finalidad de tener una guía del diseño que tendrá el prototipo EXPAPP.

Para la elaboración de las posibles pantallas de cada usuario se utilizó Figma, que es una aplicación que funciona en un navegador o en versión de escritorio. Es una de las herramientas más completas para el diseño y prototipado de aplicaciones web y móviles, pues permite la generación de código para realizar traspaso. En esta aplicación se tienen elementos de Angular Material y gracias a esto el diseño se vuelve más sencillo y rápido.

A continuación, se presenta el diseño para el usuario Administrador Técnico. En la Figura 2.12 se tiene la pantalla diseñada, la cual contiene rótulos con letras. En la Tabla 2.7 se tiene la explicación de los rótulos relacionados al requerimiento que cumplen.

Tabla 2.7. Descripción del diseño de la interfaz gráfica para rol Administrador Técnico.

Figura	Rótulo	Descripción	Requerimiento asociado
Figura 2.12	A	Al dar clic sobre este botón el usuario se dirige a la pantalla de su perfil.	Rq1, Rq2
	B	El usuario puede visualizar la información de los usuarios del rol "Administrador" distribuidos en una tabla. Así puede ver información para los usuarios de todos los roles.	Rq3
	C	Botón que permite al usuario crear un nuevo usuario con el rol "Administrador".	Rq3
	D	Botón que permite la edición del usuario que se encuentra en la fila seleccionada.	Rq3

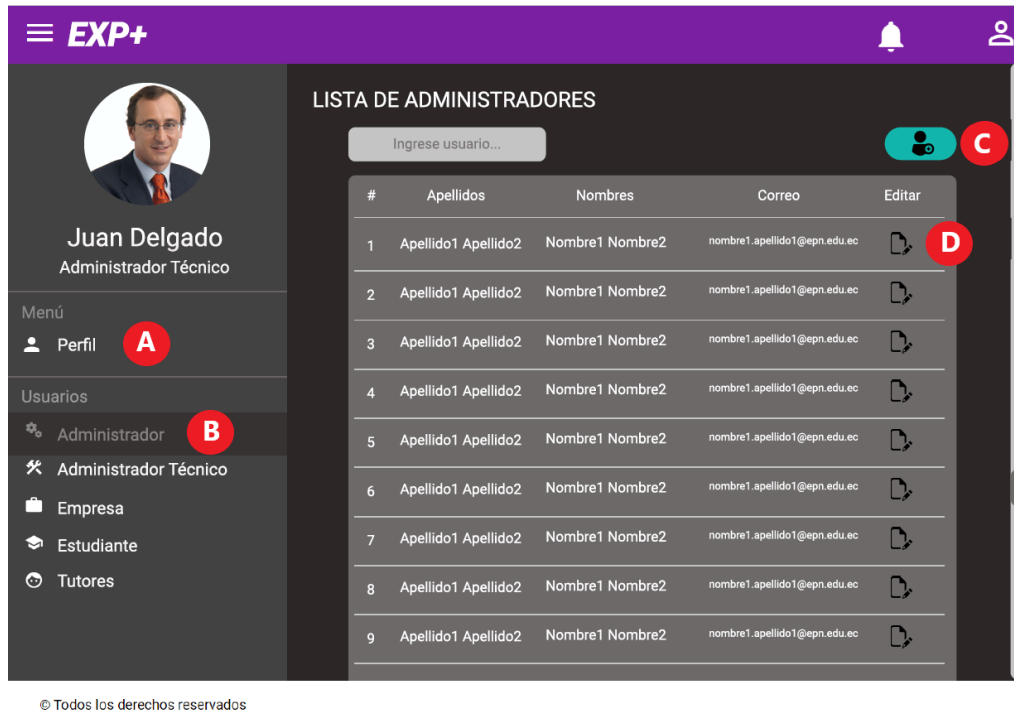


Figura 2.12. Diseño de la interfaz de usuario para rol Administrador Técnico.

En la Tabla 2.8 se tiene la descripción del diseño de la interfaz gráfica para el rol Administrador y desde la Figura 2.13 hasta la Figura 2.15 se tienen las pantallas diseñadas.



Figura 2.13. Diseño de la interfaz de usuario para rol Administrador – Estadísticas.

Tabla 2.8. Descripción del diseño de la interfaz gráfica para rol Administrador.

Figura	Rótulo	Descripción	Requerimiento asociado
Figura 2.13	A	Al dar clic sobre este botón el usuario se dirige a la pantalla de su perfil.	Rq1, Rq2
	B	El Administrador verá estadísticas de datos asociados a prácticas preprofesionales de la FIEE.	Rq12
Figura 2.14	C	En esta sección el Administrador verá anuncios creados por las empresas que están pendientes por aprobar.	Rq6
	D	Al dar clic sobre este botón el Administrador aprueba el anuncio.	Rq6
	E	En esta sección el Administrador tendrá información de Estudiantes, Empresas y Tutores.	Rq9, Rq10, Rq11
Figura 2.15	F	En esta sección el Administrador verá el detalle de los anuncios que se encuentran activos.	Rq20
	G	El Administrador tiene acceso a la información de estudiantes que han aplicado al anuncio y que se encuentran preseleccionados	Rq7, Rq9
	H	El Administrador tendrá la información del estudiante que ha aplicado al anuncio. Esta información sirve de ayuda para la selección.	Rq7, Rq9
	I	El administrador preselecciona un estudiante para la práctica preprofesional.	Rq7

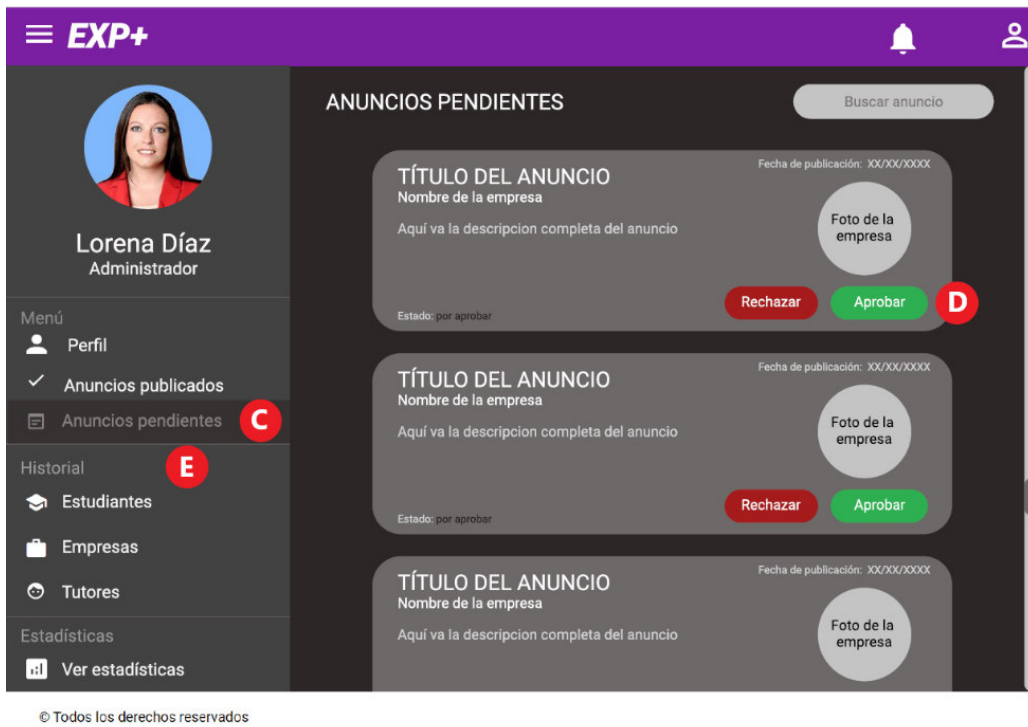


Figura 2.14. Diseño de la interfaz de usuario para rol Administrador – Anuncios Pendientes.

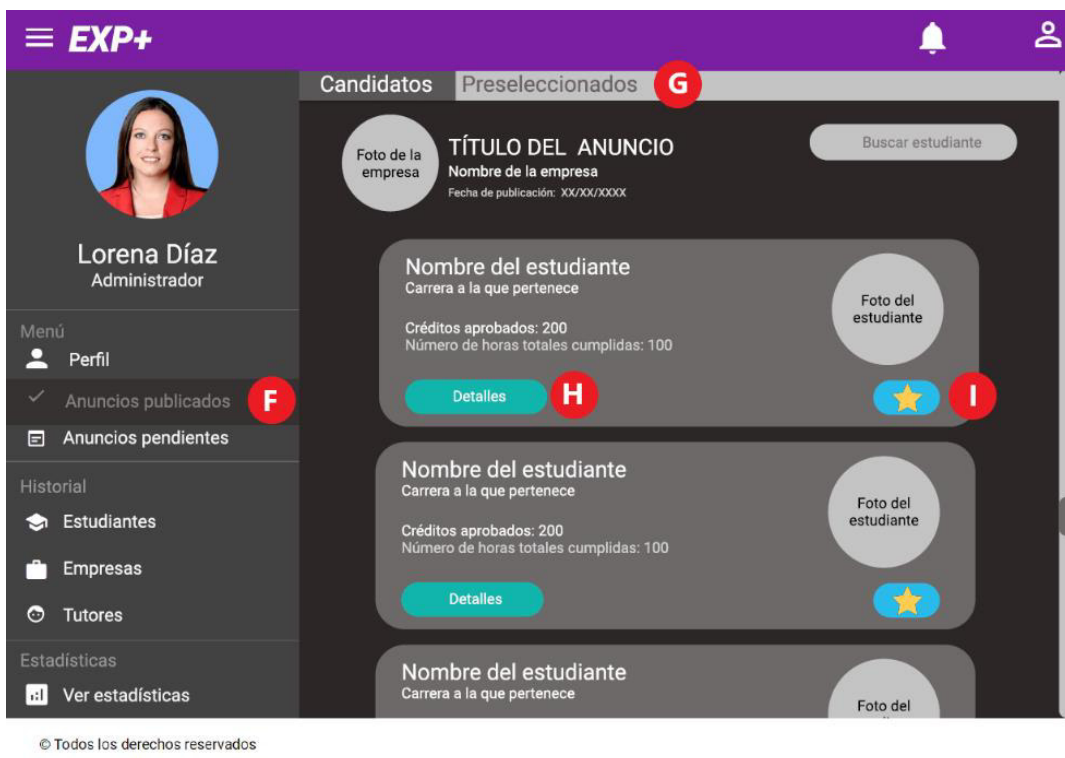


Figura 2.15. Diseño de la interfaz de usuario para rol Administrador – Detalles de anuncio.

En la Tabla 2.9 se tiene la descripción del diseño de la interfaz gráfica para el rol Supervisor Empresa y desde la Figura 2.16 hasta la Figura 2.17 se tienen las pantallas diseñadas.

Tabla 2.9. Descripción del diseño de la interfaz gráfica para rol Supervisor Empresa.

Figura	Rótulo	Descripción	Requerimiento asociado
Figura 2.16	A	Al dar clic sobre este botón el usuario se dirige a la pantalla de su perfil.	Rq1, Rq2
	B	El Supervisor Empresa puede ver los anuncios activos.	Rq20
	C	En esta pantalla se tiene el formulario de creación de un nuevo anuncio.	Rq5
Figura 2.17	D	El Supervisor Empresa tiene el detalle de las prácticas preprofesionales activas.	Rq11
	E	En el detalle de prácticas preprofesionales el Supervisor Empresa tendrá acceso a los informes.	Rq16

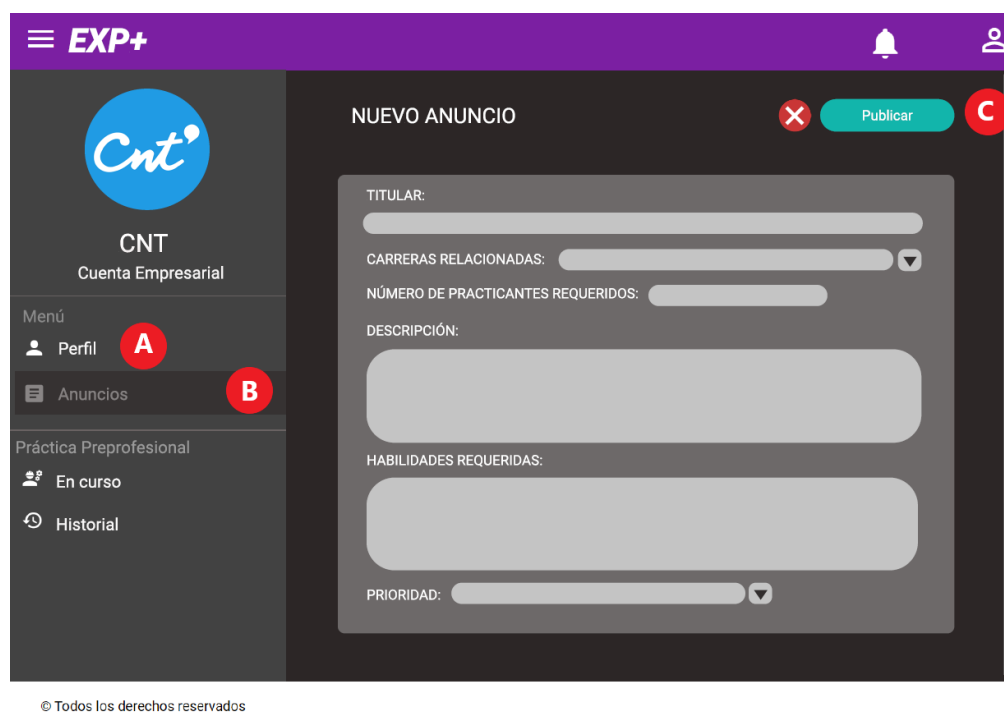


Figura 2.16. Diseño de la interfaz de usuario para rol Supervisor Empresa – Creación de anuncio.

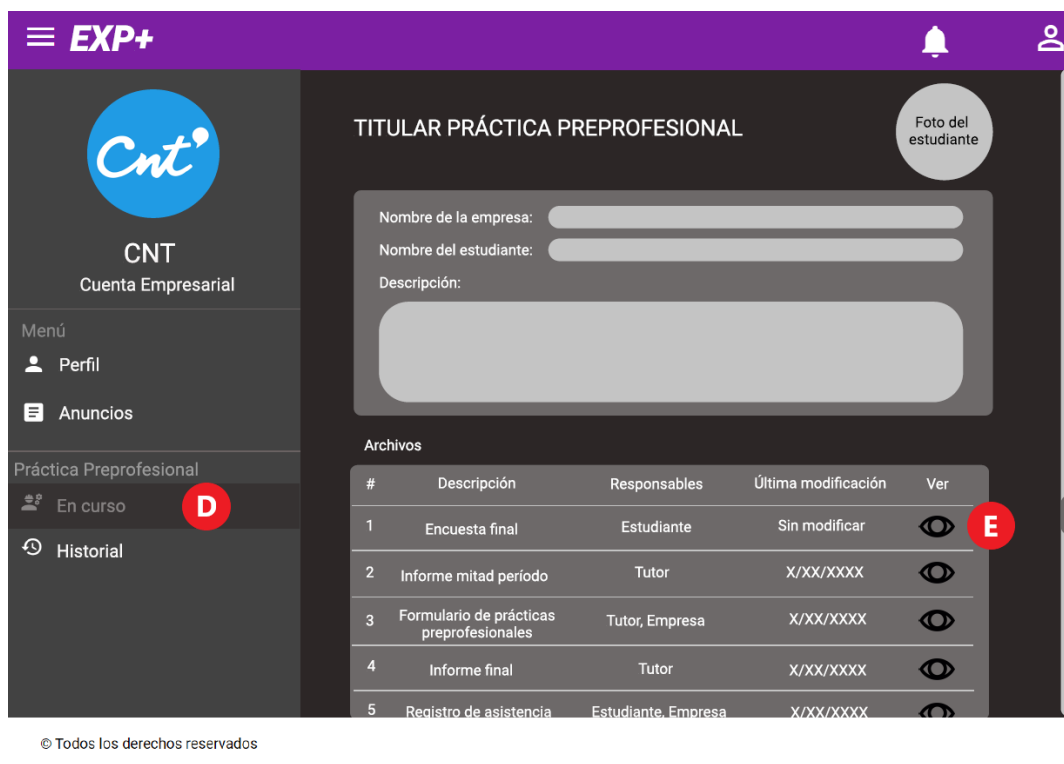
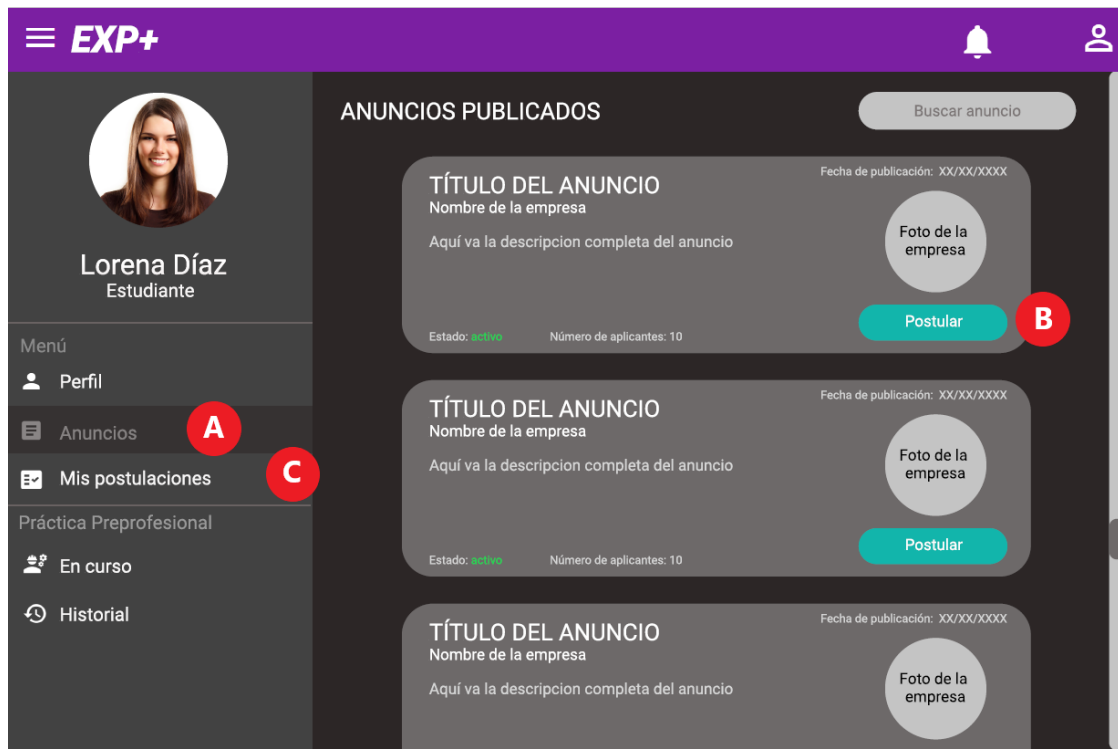


Figura 2.17. Diseño de la interfaz de usuario para rol Supervisor Empresa – Detalles de práctica preprofesional.

En la Tabla 2.10 se tiene la descripción del diseño de la interfaz gráfica para el rol Supervisor Empresa; en la Figura 2.18 y en la Figura 2.19 se tienen las pantallas diseñadas.

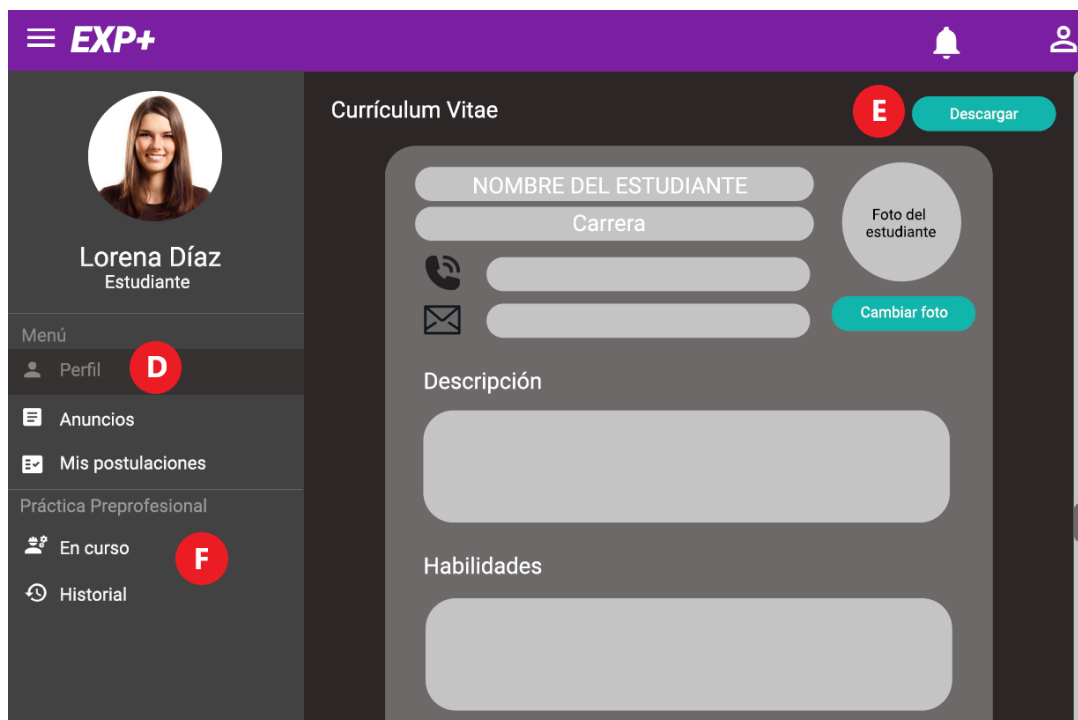
Tabla 2.10. Descripción del diseño de la interfaz gráfica para rol Estudiante.

Figura	Rótulo	Descripción	Requerimiento asociado
Figura 2.18	A	El estudiante puede ver anuncios activos.	Rq20
	B	El estudiante postula al anuncio de práctica preprofesional.	Rq21
	C	El estudiante puede ver los anuncios a los que ha postulado.	Rq21
Figura 2.19	D	Al dar clic sobre este botón el usuario se dirige a la pantalla de su perfil.	Rq1, Rq2
	E	El estudiante tiene el detalle de las prácticas preprofesionales.	Rq11
	F	El estudiante crea su CV y lo exporta como PDF.	Rq19



© Todos los derechos reservados

Figura 2.18. Diseño de la interfaz de usuario para rol Estudiante – Anuncios publicados.



© Todos los derechos reservados

Figura 2.19. Diseño de la interfaz de usuario para rol Estudiante – Creación de CV.

En la Tabla 2.11 se tiene la descripción del diseño de la interfaz gráfica para el rol Supervisor Empresa; en la Figura 2.20 y en la Figura 2.21 se tienen las pantallas diseñadas.

Tabla 2.11. Descripción del diseño de la interfaz gráfica para rol Tutor.

Imagen	Rótulo	Descripción	Requerimiento asociado
Figura 2.20	A	Al dar clic sobre este botón el usuario se dirige a la pantalla de su perfil.	Rq1, Rq2
	B	El Tutor tiene acceso al historial de prácticas preprofesionales al que ha sido asignado.	Rq11
	C	En el detalle de prácticas preprofesionales el Tutor tiene acceso a los informes.	Rq16
Figura 2.21	D	Formulario para llenar la información de un informe de práctica preprofesional.	Rq16
	E	El Tutor puede descargar el informe.	Rq16

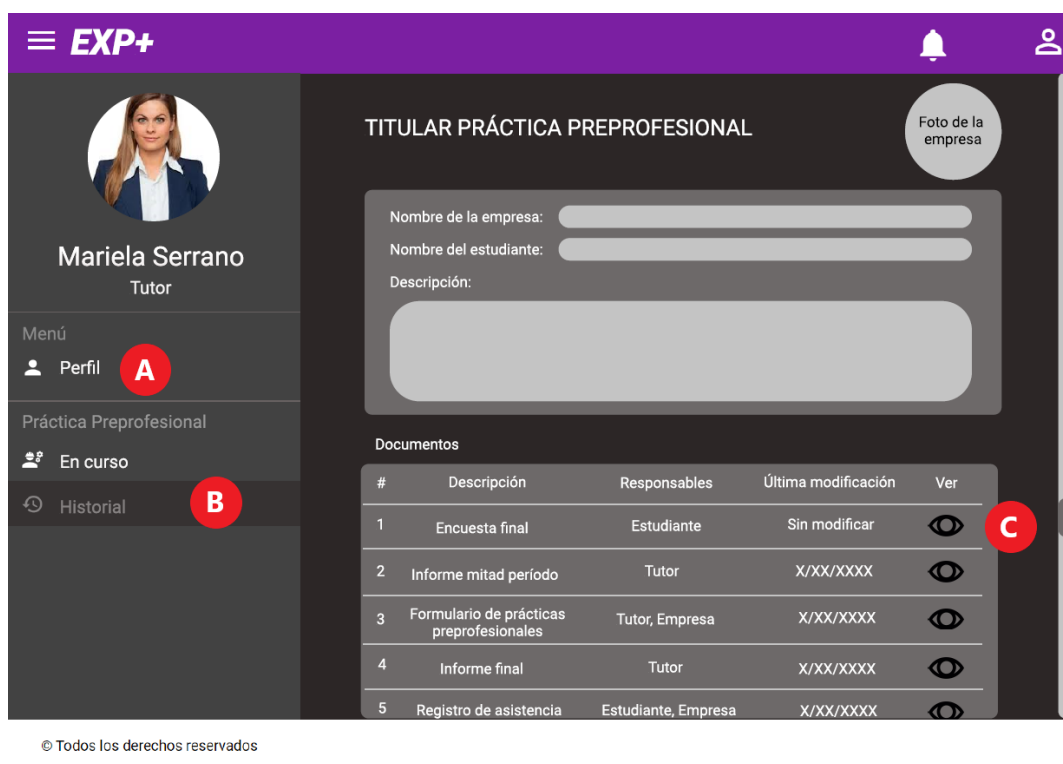


Figura 2.20. Diseño de la interfaz de usuario para rol Tutor – Detalles de práctica preprofesional.

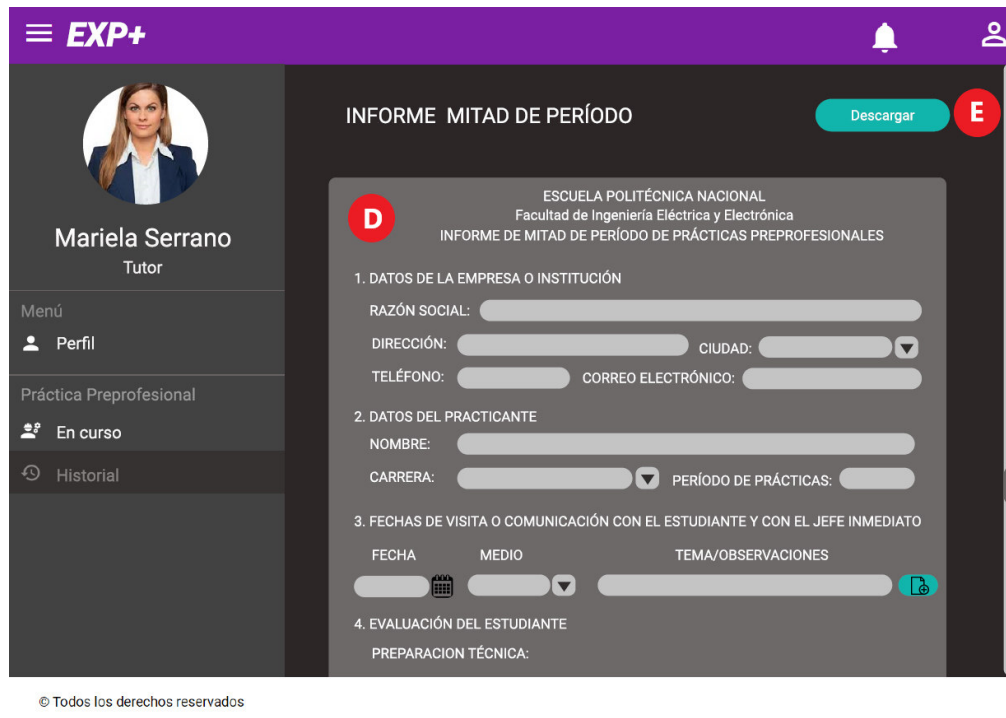


Figura 2.21. Diseño de la interfaz de usuario para rol Tutor – Formulario de informe.

2.2.5 DIAGRAMAS DE NAVEGACIÓN

En esta sección se presenta los diagramas de navegación de los cinco roles de usuario. Todos los roles tienen un Menú lateral el cual es la principal herramienta para la navegación a través de EXPAPP.

En el diagrama de navegación, cada flecha representa el cambio de una pantalla a otra, en el inicio o mitad de las flechas se identifica el botón o elemento del menú que debe ser presionado para el cambio. En la Figura 2.22 se presenta la simbología de los diagramas de navegación; y, desde la Figura 2.25 hasta la Figura 2.27 se presentan los diagramas de navegación.

SIMBOLOGÍA DIAGRAMA DE NAVEGACIÓN

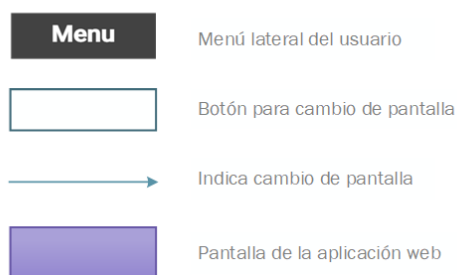


Figura 2.22. Simbología de los diagramas de navegación.

DIAGRAMA DE NAVEGACIÓN - TUTOR

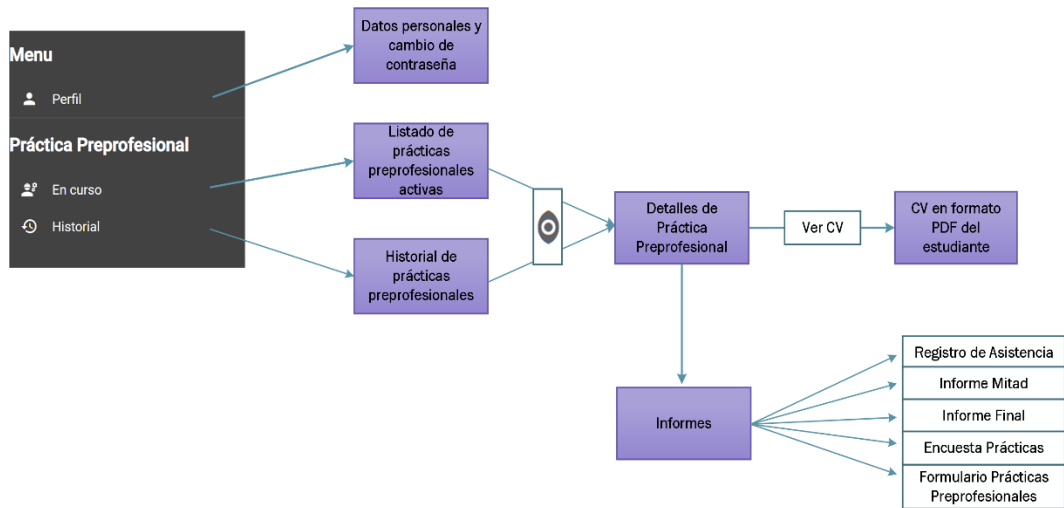


Figura 2.23. Diagrama de navegación del Tutor.

DIAGRAMA DE NAVEGACIÓN - ADMINISTRADOR

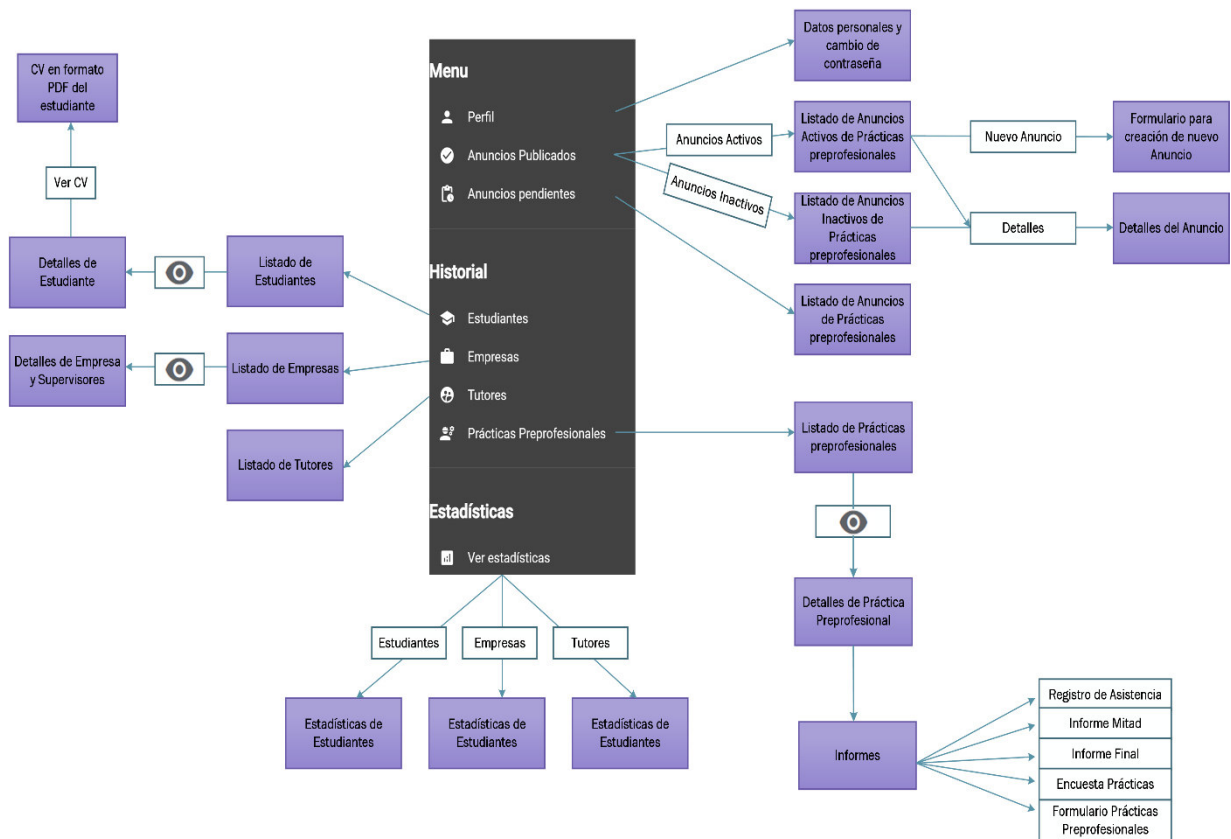


Figura 2.24. Diagrama de navegación del Administrador.

DIAGRAMA DE NAVEGACIÓN – ADMINISTRADOR TÉCNICO

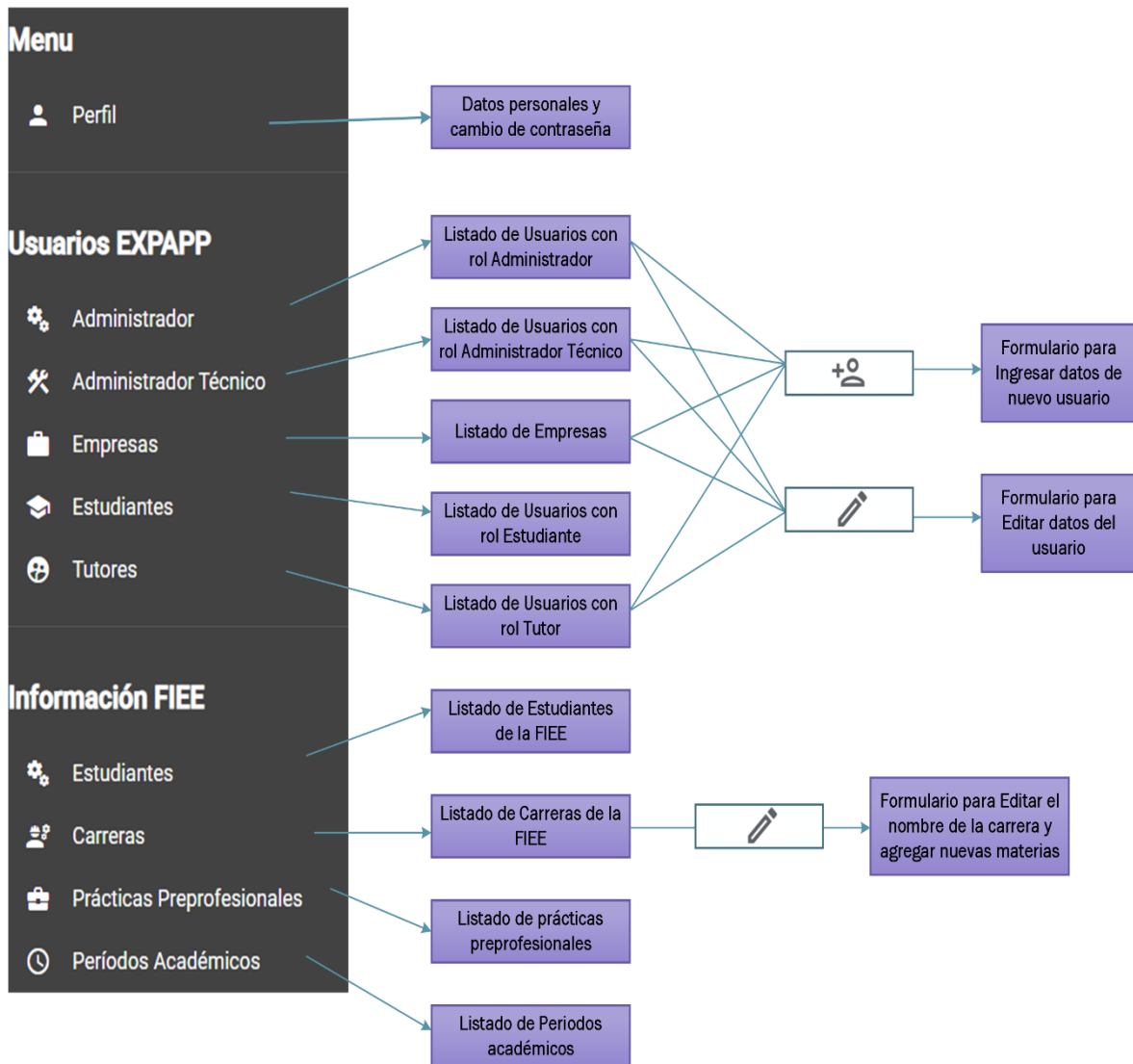


Figura 2.25. Diagrama de navegación del Administrador Técnico.

DIAGRAMA DE NAVEGACIÓN - ESTUDIANTE

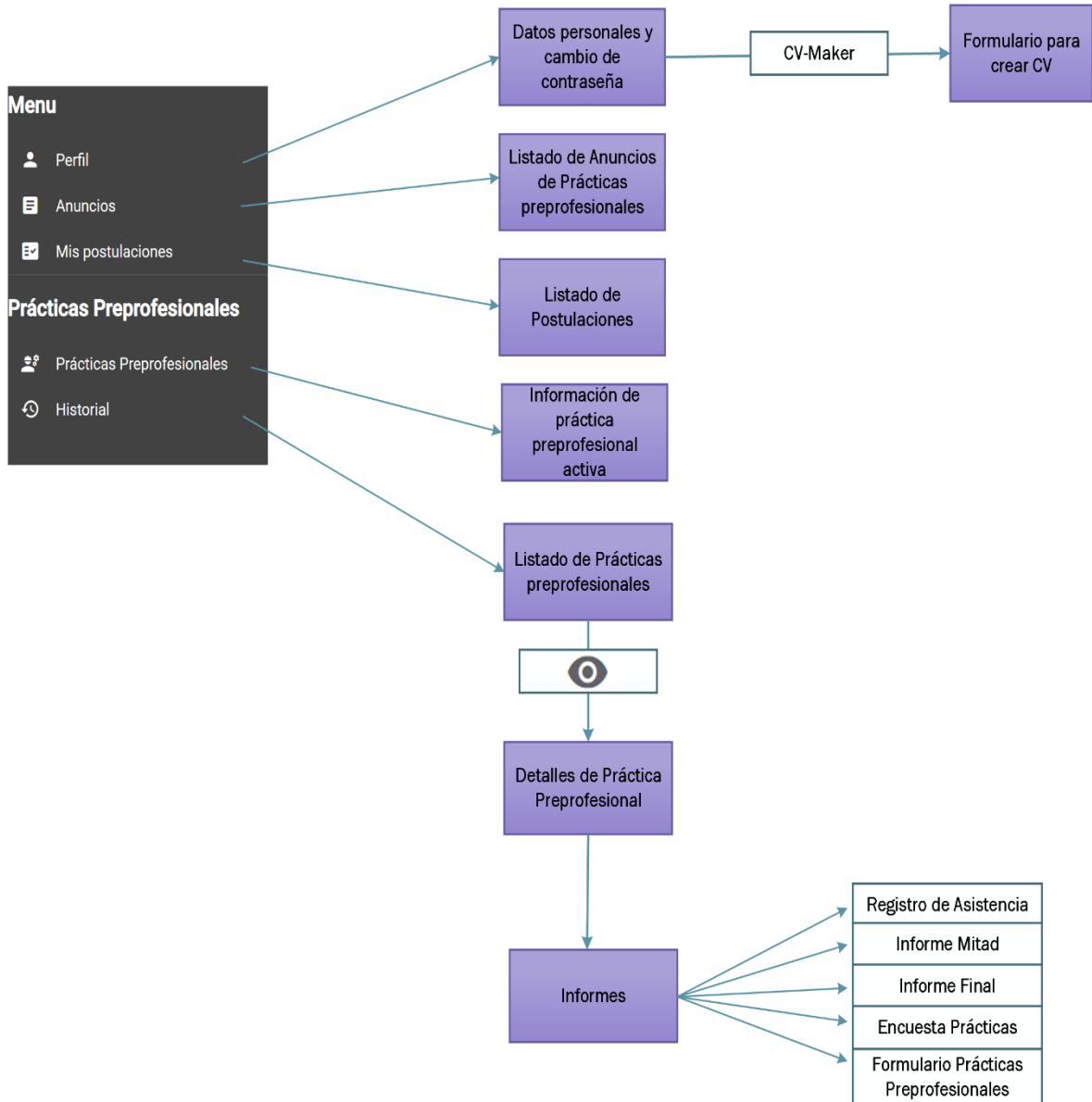


Figura 2.26. Diagrama de navegación del Estudiante.

DIAGRAMA DE NAVEGACIÓN - SUPERVISOREMPRESA

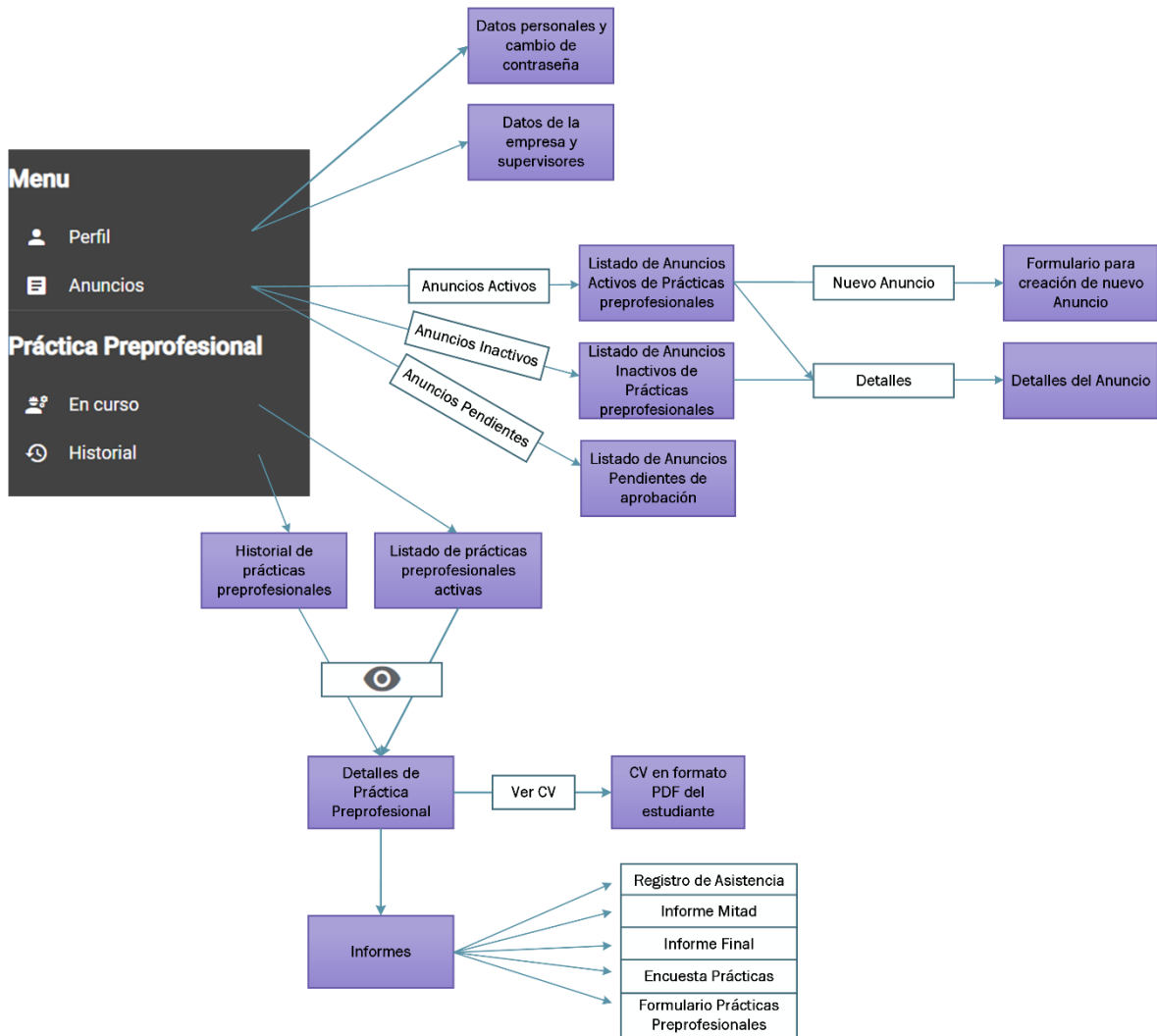


Figura 2.27. Diagrama de navegación del Supervisor Empresa.

2.3 IMPLEMENTACIÓN DEL PROTOTIPO

En este apartado se tienen tres secciones. En la primera sección se presenta las instalaciones necesarias para preparar el ambiente de desarrollo y la creación del proyecto. En la segunda sección se describe el desarrollo del *frontend* de la aplicación web y en la tercera sección el desarrollo del *backend*.

2.3.1 CREACIÓN DEL PROYECTO

El primer paso para empezar con el desarrollo del prototipo es la creación del proyecto en Firebase, para lo cual se requiere tener una cuenta de Google. Los pasos para la creación del proyecto se describen en el ANEXO A.

Para el desarrollo de la aplicación se tendrán dos ambientes (*frontend* y *backend*), los cuales se levantarán en una máquina con las siguientes características: Sistema Operativo Windows 10, Almacenamiento SSD 120GB, memoria RAM de 16GB.

En las Tabla 2.12 y Tabla 2.13 se muestran los pasos e instalaciones necesarias para crear el proyecto del *frontend* y del *backend* respectivamente.

Tabla 2.12. Pasos para el levantamiento del *frontend*.

Orden	Proceso	Aplicación	Versión
1	Ingresar a la página oficial y descargar el instalador de la última versión estable.	Node.js	14.16.1
2	En el <i>Command Prompt</i> de Windows se ejecuta el siguiente comando: <i>npm install -g @angular/cli</i>	Angular CLI	9.0.5
3	En el <i>Command Prompt</i> de Windows se cambia de ruta hacia donde se quiere crear el proyecto se ejecuta el siguiente comando: <i>ng new nombre_del_proyecto</i>	Creación del proyecto para el <i>frontend</i> con Angular	No aplica
4	A partir de este paso en el <i>Command Prompt</i> de Windows se ejecuta el comando en la ruta en donde se creó el proyecto: <i>npm install @angular/flex-layout</i>	Angular Flex-Layout	9.0.0-beta.31
6	<i>ng add @angular/material</i>	Angular Material	9.2.4
7	<i>ng add @angular/fire</i>	AngularFire	6.0.2
8	<i>npm install --save ng2-charts</i> <i>npm install --save chart.js</i>	ng2-charts	2.4.2
9	<i>npm install pdfmake</i>	pdfmake	0.1.70
10	<i>npm install sweetalert2</i>	sweetalert2	10.8.1
11	<i>ng generate module nombre_del_módulo</i>	Creación de módulos	No aplica
12	<i>ng generate service nombre_del_servicio</i>	Creación de servicios	No aplica
13	<i>ng generate component nombre_del_componente</i>	Creación de componentes	No aplica

Tabla 2.13. Pasos para el levantamiento del *backend*.

Orden	Instalación	Descripción	Versión
1	En el <i>Command Prompt</i> de Windows se ejecuta el siguiente comando: <i>npm install -g firebase-tools</i>	Firebase	3.11.0
2	En el <i>Command Prompt</i> de Windows se ejecuta el siguiente comando: <i>npm install firebase-functions@latest firebase-admin@latest --save</i>	Firebase Functions	9.2.0
3	En el <i>Command Prompt</i> de Windows se ejecuta el siguiente comando: <i>firebase login</i>	Iniciar sesión en Firebase	No aplica
4	En el <i>Command Prompt</i> de Windows se cambia de ruta hacia la carpeta en donde se quiere crear el proyecto se ejecuta el siguiente comando: <i>firebase init functions</i>	Creación del proyecto de Functions para el <i>backend</i>	No aplica
5	En el <i>Command Prompt</i> de Windows se ejecuta el comando en la ruta en donde se creó el proyecto: <i>npm install nodemailer</i>	NodeMailer	6.4.16

2.3.2 IMPLEMENTACIÓN BACKEND

En esta sección se presenta el desarrollo del *backend* de EXPAPP. En la primera sección, 2.3.2.1, se describe la estructura del proyecto y las configuraciones necesarias para inicializar la aplicación. En la segunda sección, 2.3.2.2, se describen las funciones implementadas y su despliegue en Firebase. El código del *backend* se encuentra en el ANEXO B.

2.3.2.1 Componentes del *backend*

En base al diagrama de arquitectura de la aplicación que se encuentra en la Figura 2.8, definido en la sección 2.2.1, se ha definido que el *backend* de EXPAPP se desarrollará sobre Firebase. En la Figura 2.28 se muestran los componentes de Firebase que serán parte del *backend* de la aplicación: *Authentication*, *Cloud Firestore*, *Cloud Storage* y *Functions*. En la Tabla 1.2 se tiene la descripción de los productos de Firebase.

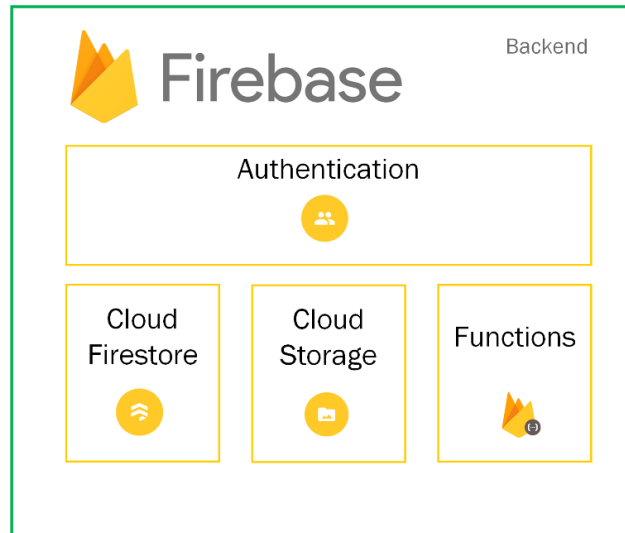


Figura 2.28. Componentes del *backend*.

Para el uso de Authentication en primer lugar se debe realizar la configuración del método de autenticación que se utilizará, para EXPAPP se utilizan credenciales (correo electrónico y contraseña) en la Tabla 1.2 se tiene la información de los métodos de autenticación disponibles. Para usar *Cloud Firestore* y *Cloud Storage* se configura las reglas de acceso, que permiten acceder desde el *frontend* a la base de datos y a los archivos almacenados. Estas configuraciones se las realizan mediante el navegador web en la aplicación creada en Firebase. En la página oficial de Firebase se tiene la documentación para el uso de sus productos [42].

EXPAPP tiene ciertos requerimientos que necesitan una configuración personalizada, como el envío de correos electrónicos y recordatorios. Para crear esta funcionalidad adicional que Firebase no tiene disponible, se utiliza *Firebase Functions*.

Firebase Functions requiere la creación de un proyecto, como se indicó en la Tabla 2.13. Al crear el proyecto JavaScript se genera una estructura de carpetas (Figura 2.29). La lógica del *backend* estará en el archivo `index.js`.

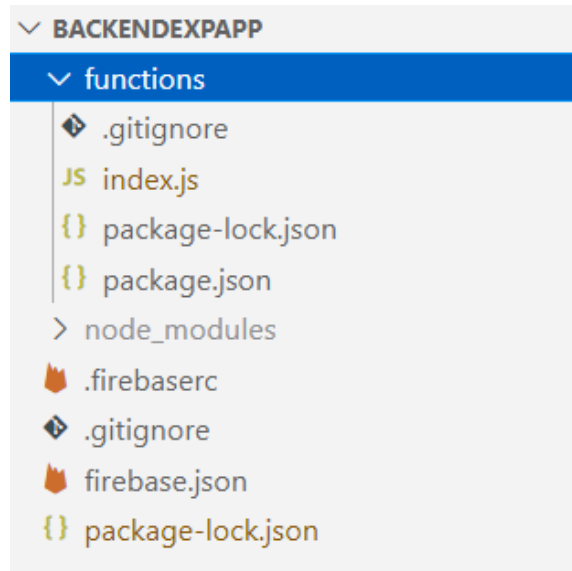


Figura 2.29. Estructura del *backend* de EXPAPP.

Para hacer uso de Firebase se requiere importar los siguientes módulos: *Cloud Functions* para exportar las funciones creadas a Firebase, *Firebase Admin* para acceder a otros productos de Firebase y *NodeMailer* para el envío de correo electrónico, como se muestra en las líneas 1 a la 3 del Código 2.1. En la línea 5 se crea una instancia de *Firebase Admin* que permite realizar cambios en Cloud Storage y Authentication del proyecto de Firebase [43].

```
1  const functions = require("firebase-functions");
2  const admin = require("firebase-admin");
3  const nodemailer = require("nodemailer");
4
5  admin.initializeApp();
6
```

Código 2.1. EXPAPP-BACKEND: Declaración de módulos necesarios en index.js.

2.3.2.2 Funciones

Para la creación de funciones se utiliza el método ***functions.https.onCall*** el cual crea funciones HTTPS que admite llamadas desde una aplicación, en otras palabras se crean *endpoints*, en este caso desde el frontend [44]. En la Tabla 2.14 se muestran las funciones creadas y una descripción de lo que hacen.

Tabla 2.14. Funciones implementadas en el *backend*.

Función	Descripción
exports.signUpUser	Primero se verifica que el email enviado no exista dentro de <i>Firestore Authentication</i> , si es que no existe crea el nuevo usuario con su email y contraseña.
exports.sendEmailInvitation	Mediante el transportador, descrito en el Código 2.2, se envía un correo electrónico predeterminado con una plantilla (Código 2.3). En la sección 2.3.2.2.1 se detalla esta función.
exports.sendNotification	Mediante el transportador de correo electrónico se envía una notificación y también guarda la notificación en la base de datos en una colección de notificaciones asociada a cada usuario.
exports.deleteUserAuth	Esta función recibe un email y trae los datos del usuario de <i>Firestore Authentication</i> , en el caso de existir el usuario lo elimina con la función de <i>Firestore Admin deleteUser()</i> la cual recibe como parámetro el Id del Usuario. En el caso de que no exista el usuario se retorna un mensaje de error.
exports.tasks	Esta función es programada, es decir, una vez configurada se ejecuta cada cierto tiempo según se especifique. Se tiene una colección llamada <i>Tasks</i> , en la cual se almacenan notificaciones pendientes por enviar, esto se utiliza para el envío de recordatorios a los usuarios. Esta función revisa cada minuto el estado de los documentos existentes en la colección <i>Tasks</i> , si encuentra tareas pendientes las ejecuta y cambia su estado para indicar que se han completado. En la sección 2.3.2.2.2 se detalla esta función.

Para subir las funciones a *Firestore Functions* se requiere ejecutar el comando en el *Command Prompt* de Windows ubicado en la carpeta del proyecto:

firebase deploy --only functions

2.3.2.2.1 Función exports.sendEmailInvitation

El envío de correo electrónico se lo realiza con *NodeMailer*, que es un módulo de Node.js para envío de correo de manera sencilla con el protocolo SMTP [45]. Una vez instalado se debe crear el transportador que es el objeto que envía el correo electrónico [46]. La configuración necesaria para el transportador está en el Código 2.2, y se requiere las credenciales de la cuenta de Gmail utilizada (líneas 34 y 35).

```

29   const mailTransport = nodemailer.createTransport({
30     host: "smtp.gmail.com",
31     port: 465,
32     secure: true,
33     auth: {
34       user: "",
35       pass: "",
36     },
37   });

```

Código 2.2. EXPAPP-BACKEND: Configuración del transportador de correo electrónico.

El siguiente paso es definir la plantilla del correo electrónico que se enviará como se observa en Código 2.3. Esta plantilla se usa en la función *sendEmailInvitation* del Código 2.4.

```

39   const configureEmail = (email, password, name, role) => ({
40     from: `${APP_NAME} <noreply@expapp>`,
41     to: email,
42     subject: `Se ha creado tu cuenta en ${APP_NAME}!`,
43     text: `Saludos ${ name || ""}`,
44
45     Te damos la bienvenida a ${APP_NAME}!
46
47     Tu usuario tiene el rol de: ${role} y tus credenciales son:
48
49     Email: ${email}
50     Contraseña: ${password}
51
52     Ingresa mediante el siguiente link: https://prototipoexp.web.app/iniciar-sesion.
53
54     Por seguridad cambia tu contraseña cuando inicies sesión!
55
56     Saludos,
57     EXPAPP Team`,
58   });

```

Código 2.3. EXPAPP-BACKEND: Configuración de la plantilla del correo electrónico que se enviará cuando se crea la cuenta de un usuario.

Finalmente, se crea la función que se cargará a *Firebase Functions*. La cual utiliza el transportador y envía la plantilla configurada con los datos del usuario que se reciben desde el *frontend* como se muestra en el Código 2.4.

```

60   exports.sendEmailInvitation = functions.https.onCall(
61     async({ email, password, name, role }) => {
62       await mailTransport.sendMail(configureEmail(email, password, name, role));
63       return { msg: "Correo enviado con éxito" };
64     });

```

Código 2.4. EXPAPP-BACKEND: Función para enviar correo electrónico a un usuario.

2.3.2.2.2 Función `exports.tasks`

Una función programada es una acción que se va a realizar periódicamente, en Firebase la configuración de estas funciones se la realiza con un Activador Pub/Sub. En esta sección se explica la configuración de una función programada para el envío de recordatorios mediante notificación y correo electrónico [47].

Para la configuración de recordatorios desde el *frontend* se guardan documentos contienen la información de tareas por realizar en una colección de Firestore llamada *Tasks*. En el *backend* se crea una clase que contiene los atributos de la tarea. En la Tabla 2.15 se tiene los atributos que pertenecen a la clase *Task* creada en el Código 2.5.

Tabla 2.15. Descripción de los atributos de la clase *Task*.

Atributo	Tipo	Descripción
options	JSON	Este campo tiene como atributos: <ul style="list-style-type: none">• <i>body</i>: descripción de la acción que se le pide realizar en el recordatorio que debe realizar el usuario.• <i>role</i>: rol del usuario al que se le va a enviar el recordatorio.• <i>route</i>: ruta a la que se le redirige al usuario para ver la acción que se le pide realizar en el recordatorio.• <i>title</i>: asunto del recordatorio.• <i>uid</i>: identificador del usuario al que se le enviará el recordatorio.
performAt	dateTime	Indica el momento en que debe activarse la función.
status	string	Indica el estado de la tarea. <ul style="list-style-type: none">• <i>scheduled</i>: si la tarea está programada y pendiente de realizar.• <i>completed</i>: indica si ya se ejecutó la tarea.• <i>error</i>: si al ejecutar la acción se genera algún problema.
worker	string	Es una función que se llama cuando se dispara el evento. En el Código 2.6 se tiene la definición del <i>worker</i> para el envío de recordatorios.

```
183 class Task {
184     constructor(options, performAt, status, worker) {
185         this.options = options;
186         this.performAt = performAt;
187         this.status = status;
188         this.worker = worker;
189     }
190     toString() {
191         return this.worker;
192     }
193 }
```

Código 2.5. EXPAPP-BACKEND: Clase *Task*.

En el Código 2.6 se muestra el proceso que realiza el *worker sendRecordatory*. Los *workers* son funciones asíncronas que definen tareas por realizar cuando la función se active, en este caso solo hay uno, que corresponde al envío de recordatorios. En el *worker sendRecordatory* primero se consulta de Firestore el usuario al que se le va a enviar el recordatorio (línea 167) y se verifica que exista (línea 169). Si el usuario existe se envía un correo electrónico (líneas 171 - 173) y se guarda la notificación (línea 174). Si no existe el usuario devuelve un mensaje de error (línea 177).

```
165  const workers = {
166    sendRecordatory: async() => {
167      await db.collection(optionsAux.role).doc(optionsAux.uid).get()
168      .then((doc) => {
169        if (doc.exists) {
170          const { uid, name, email, role } = doc.data();
171          mailTransport.sendMail(configureNotification(email, name,
172            optionsAux.title,
173            optionsAux.body, optionsAux.route));
174          saveNotification(uid, role, optionsAux.title, optionsAux.body, optionsAux.route);
175        }
176      }).catch((error) => {
177        console.log("Error getting user:", error);
178      });
179    }
180  };
```

Código 2.6. EXPAPP-BACKEND: Definición de *workers*.

En el Código 2.7 se tiene la implementación de la función que se ejecutará cada cierto tiempo definido por *pubsub.schedule('minutos hora díaMes mes díaSemana')*, si en lugar de un valor se coloca un "*" esto significa que ese campo toma cualquier valor. En la línea 139 se observa que se ha configurado con el valor ("* * * * *"), lo que significa que se ejecuta cada minuto.

En la línea 141 del Código 2.7 se guarda la fecha actual en una variable y se consulta los documentos de la colección de *Tasks* de Firestore que cumplan con dos condiciones (líneas 143 - 147). La primera condición es que el campo *performAt* sea menor o igual a la fecha actual con el fin de no traer tareas que deben ejecutarse en el futuro; y la segunda condición es que su estado sea "*scheduled*" para desde la base de datos se devuelva únicamente los registros que no hayan sido ejecutados.

En el Código 2.7 en la línea 146 se utiliza la función *taskConverter*, que se encuentra en el Código 2.8, la cual mapea los registros obtenidos de Firestore con la clase *Task*. En la línea 149 se define un array en donde se guardarán las tareas pendientes como promesas. Una promesa es un objeto que indica el término de la ejecución de una función asíncrona [48].

En la línea 151 del Código 2.7 se recorre el arreglo que tiene las tareas por realizar; y se guarda el *worker* y las opciones de la tarea en variables (línea 148). A continuación, se define una constante llamada *job* que contiene la promesa del *worker* especificado, se define que si la promesa se resuelve exitosamente se cambia el estado del documento de la tarea a “*complete*”, y si se genera algún error el estado cambiará a “*error*”. Finalmente, se agrega el *job* al array *jobs* (línea 159) y una vez que se haya recorrido todo el array de Tareas se retorna de la función el resultado de las promesas guardadas en *jobs* (línea 162).

```

139 exports.tasks = functions.runWith({ memory: '2GB' }).pubsub.schedule('* * * * *')
140   .onRun(async(context) => {
141     const now = db.Timestamp.now();
142
143     var tasks = await db.collection('Tasks')
144       .where('performAt', '<=', now)
145       .where('status', '!=', 'scheduled')
146       .withConverter(taskConverter)
147       .get();
148
149     const jobs = [];
150
151     tasks.forEach(snapshot => {
152       const { worker, options } = snapshot.data();
153       optionsAux = options;
154
155       const job = workers[worker]()
156         .then(() => snapshot.ref.update({ status: 'complete' }))
157         .catch((err) => snapshot.ref.update({ status: 'error' }));
158
159       jobs.push(job);
160     });
161
162     return await Promise.all(jobs);
163   });

```

Código 2.7. EXPAPP-BACKEND: Función *tasks* implementada.

```

132 var taskConverter = {
133   fromFirestore: function(snapshot, options) {
134     const data = snapshot.data(options);
135     return new Task(data.options, data.performAt, data.status, data.worker);
136   }
137 };

```

Código 2.8. EXPAPP-BACKEND: Método *taskConverter*.

2.3.3 IMPLEMENTACIÓN FRONTEND

En esta sección se describe el proceso de desarrollo del *frontend* de EXPAPP. Se detallan los componentes implementados y uso de librerías para el desarrollo web; y el código de los módulos, servicios y componentes más significativos en la aplicación. El código completo de la aplicación se encuentra en el ANEXO C.

2.3.3.1 Componentes

En esta sección se presentan todos los componentes creados durante la implementación de la aplicación web. En base a la Figura 2.9 (Diseño de los módulos de la aplicación web) se creó un conjunto de componentes propios de cada rol de usuario (Tabla 2.18 - Tabla 2.22) y también componentes que son compartidos (Tabla 2.16 y Tabla 2.17).

Tabla 2.16. Componentes para la autenticación de usuarios en la aplicación web.

Nombre	Requerimiento relacionado	Descripción
<i>login</i>	-	Formulario que permite ingresar correo y contraseña para el inicio de sesión.
<i>password-reset</i>	Rq2	Formulario que permite el ingreso de correo electrónico para cambiar la contraseña en el caso de olvido.
<i>register</i>	Rq18	Formulario que permite el registro en la aplicación web mediante correo y contraseña para estudiantes de la FIEE.

Tabla 2.17. Componentes en común para todos los usuarios de la aplicación web.

Nombre	Requerimiento relacionado	Descripción
<i>final-report</i>	Rq16	Permite la modificación de un formulario para la generación del Informe Inf4 .
<i>footer</i>	-	Barra ubicada en la parte inferior de la página web con información de la aplicación web.
<i>header</i>	-	Menú que se despliega en la parte superior de la página web que permite la visualización de notificaciones, el cierre de sesión y mostrar/ocultar el <i>sidebar</i> de cada usuario.
<i>hour-report</i>	Rq16	Permite la modificación de un formulario para la generación del Informe Inf5 .
<i>internship-form</i>	Rq16	Permite la modificación de un formulario para la generación del Informe Inf2 .
<i>internship-survey</i>	Rq16	Permite la modificación de un formulario para la generación del Informe Inf1 .
<i>midterm-report</i>	Rq16	Permite la modificación de un formulario para la generación del Informe Inf3 .
<i>not-found</i>	-	Despliega un mensaje con el error 404 en el caso de que el usuario quiera acceder a una ruta que no existe o se encuentra protegida con un <i>guard</i> .
<i>profile</i>	Rq1	Muestra los datos del usuario que inicia sesión y permite actualizarlos. También se tiene la opción de cambio de contraseña.
<i>reports</i>	Rq16	Agrupar en pestañas todos los informes correspondientes a una práctica preprofesional.

Tabla 2.18. Componentes para usuario Administrador técnico.

Nombre	Requerimiento relacionado	Descripción
<i>career-details</i>	Rq4	Se tiene un formulario para cambiar el nombre de una carrera y agregar nuevas materias.
<i>careers</i>	Rq4	Se tiene una lista con el resumen de las carreras que pertenecen a la FIEE.
<i>new-company</i>	Rq3	Consiste en un formulario para la creación de una nueva empresa y un usuario de tipo supervisor empresa.
<i>new-user</i>	Rq3	Formulario con datos básicos para la creación de usuarios de tipo Administrador, Administrador Técnico, Estudiante y Tutor.
<i>Sidebar-tech-admin</i>	-	Menú lateral que permite al administrador técnico la navegación por la página web.
<i>students-fiee</i>	Rq4	Es una lista de estudiantes de la FIEE, para cargar nuevos estudiantes o modificar alguno se requiere importar un documento en Excel de tipo CSV.
<i>update-company</i>	Rq4	Contiene un formulario en el cual se carga la información de una empresa y puede ser modificada. También se tiene la opción de agregar nuevos supervisores y actualizar los que ya están registrados.
<i>update-user</i>	Rq4	Contiene un formulario en el cual se carga la información de un usuario y puede ser modificada.
<i>view-company</i>	Rq4	Permite ver la lista de empresas activas o inactivas.
<i>view-internship</i>	Rq4	Permite ver la lista de todas prácticas preprofesionales estén activas o no.
<i>view-user</i>	Rq4	Permite ver una lista de usuarios activos o inactivos según el rol seleccionado.

Tabla 2.19. Componentes para usuario Tutor.

Nombre	Requerimiento relacionado	Descripción
<i>internship-details</i>	<ul style="list-style-type: none"> • Rq9 • Rq10 • Rq16 	Se presentan los datos de una práctica preprofesional, estudiante y empresa correspondientes. También se muestran los informes relacionados a la práctica.
<i>internships-active</i>	Rq11	Muestra una lista con el resumen de las prácticas preprofesionales que se encuentran activas para el tutor.
<i>internships-record</i>	Rq11	Muestra una lista con el resumen de todas las prácticas preprofesionales a las que ha sido asignado el tutor.
<i>sidebar-tutor</i>	-	Menú lateral que permite al tutor la navegación por la página web.

Tabla 2.20. Componentes para Administrador.

Nombre	Requerimiento relacionado	Descripción
<i>announcement-details</i>	<ul style="list-style-type: none"> • Rq6 • Rq7 • Rq8 	Se presentan los datos de un anuncio, y el administrador podrá ver los estudiantes que han aplicado al puesto, preseleccionarlos y finalmente seleccionar uno o varios según el número de vacantes.
<i>announcements</i>	-	Presenta una lista de anuncios de prácticas preprofesionales de todas las empresas. Se tiene anuncios activos o inactivos.
<i>announcements-no-approved</i>	Rq6	Presenta una lista de anuncios de prácticas preprofesionales creados por las empresas y se encuentran pendientes de aprobar para su publicación.
<i>company</i>	Rq10	Permite ver la lista de empresas activas o inactivas.
<i>company-details</i>	Rq10	Se presenta información de la empresa y sus supervisores.
<i>internship</i>	Rq11	Permite ver la lista de todas prácticas preprofesionales.
<i>internship-details</i>	Rq11	Se presentan los datos de una práctica preprofesional, estudiante y empresa correspondientes. Se le permite al administrador modificar datos como el tutor y periodo. También se muestran los informes relacionados a la práctica.
<i>new-announcement</i>	Rq5	Este componente tiene un formulario que solicita datos para la creación de un nuevo anuncio.
<i>sidebar-admin</i>	-	Menú lateral que permite al administrador la navegación por la página web.
<i>statistics</i>	Rq12	Este componente agrupa por pestañas las estadísticas de empresa, estudiante y tutor.
<i>statistics-company</i>	Rq12	Se tiene gráficas para el análisis de datos relacionados a las empresas.
<i>statistics-student</i>	Rq12	Se tiene gráficas para el análisis de datos relacionados a los estudiantes
<i>statistics-tutor</i>	Rq12	Se tiene gráficas para el análisis de datos relacionados a los tutores.
<i>student</i>	Rq9	Una lista que tiene información de los estudiantes que se encuentran registrados en la página web.
<i>student-details</i>	Rq9	Se muestra un resumen de los datos del estudiante. Se puede acceder a su CV.
<i>tutor</i>	Rq8	Es una lista de los tutores que tienen cuenta activa en la aplicación web y se muestra cuántas prácticas preprofesionales tienen activas.

Tabla 2.21. Componentes para Empresa.

Nombre	Requerimiento relacionado	Descripción
<i>announcement-details</i>	<ul style="list-style-type: none"> • Rq9 • Rq16 	Se tiene los detalles del anuncio, si el anuncio está activo hay un botón que permite desactivarlo o si esta desactivo se puede volver a publicar.
<i>announcements</i>	-	Presenta una lista de anuncios de prácticas preprofesionales de la empresa. Se tiene anuncios activos o inactivos.
<i>confirm-internship</i>	Rq13	Cuando se empieza una práctica preprofesional, el administrador selecciona un estudiante que debe ser aceptado o rechazado por la empresa. Si es aceptado, en este componente se le permite a la empresa asignar un supervisor, fecha de inicio y fecha de fin.
<i>internship-details</i>	<ul style="list-style-type: none"> • Rq11 • Rq15 • Rq16 • Rq17 	Se presentan los datos de una práctica preprofesional, estudiante y empresa correspondientes. Se le permite Modificar a la empresa la fecha de inicio y fin de la práctica. También se muestran los informes relacionados a la práctica y puede dar fin a la práctica.
<i>internships-active</i>	Rq11	Muestra una lista con el resumen de las prácticas preprofesionales que se encuentran activas para el supervisor de la empresa que inició sesión.
<i>internships-record</i>	Rq11	Muestra una lista con el resumen de todas las prácticas preprofesionales a las que ha sido asignado el para el supervisor de la empresa que inició sesión.
<i>new-announcement</i>	Rq5	Este componente tiene un formulario que solicita datos para la creación de un nuevo anuncio.
<i>profile</i>	<ul style="list-style-type: none"> • Rq1 • Rq2 • Rq14 	El supervisor de la empresa tendrá sus datos que puede actualizarlos, los datos de la empresa o de otros supervisores. También se le permite agregar nuevos supervisores.
<i>sidebar-company</i>	-	Menú lateral que permite al supervisor de la empresa la navegación por la página web.
<i>student-details</i>	Rq9	Se muestra un resumen de los datos del estudiante. Se puede acceder a su CV.

Tabla 2.22. Componentes para estudiante.

Nombre	Requerimiento relacionado	Descripción
<i>announcements</i>	Rq20	Presenta una lista de anuncios de prácticas preprofesionales de la carrera del estudiante y se puede aplicar al puesto.
<i>cv-builder</i>	Rq19	Es un formulario que le permite al estudiante ingresar datos para la creación de su CV. El estudiante puede exportar su CV como PDF.
<i>internship-details</i>	Rq16	Se presentan los datos de una práctica preprofesional, estudiante y empresa correspondientes. También se muestran los informes relacionados a la práctica.
<i>internships-record</i>	Rq11	Muestra una lista con el resumen de todas las prácticas preprofesionales que ha realizado el estudiante.
<i>my-applications</i>	Rq21	Presenta una lista de anuncios de prácticas preprofesionales a los que ha aplicado el estudiante, se le permite quitar la postulación.
<i>sidebar-student</i>	-	Menú lateral que permite al estudiante la navegación por la página web.
<i>student-profile</i>	Rq2	Se presenta los datos personales del estudiante, pero no puede modificarlos. También se tiene un apartado de actualización de la contraseña.

2.3.3.2 AppRoutingModule y RoleGuard

En el diseño de la aplicación web se especificó que cada rol de usuario tendrá su módulo de rutas (**Figura 2.9.** Diseño de los módulos de la aplicación web.). En el archivo *app.routing.module* se definen las rutas principales y en los archivos *routing.module* de cada usuario se tendrá las rutas hijas.

En el Código 2.9 se tiene la definición de rutas principales de la aplicación web. En las líneas 13 - 28 se definen rutas para componentes compartidos, la definición de estas rutas consiste en definir el nombre (*path*) y el componente al que va a dirigirse. En las líneas 28 - 33 se tiene la definición de la ruta padre para el rol Administrador. Se define el nombre de la ruta principal, y con la etiqueta *loadChildren* se especifica de qué módulo se extraerán las rutas hijas.

Para mantener las rutas de cada rol protegidas, con la etiqueta *canActivate* se define el *guard* que llevará este control y en la etiqueta *data* se especifica el rol que se requiere verificar en el *guard*. Con esto se controla que el acceso a las rutas hijas, definidas en

AdministratorModule, debe ser por un usuario haya iniciado sesión y tenga el rol Administrador.

En las líneas 34 a la 57 del Código 2.9 se tiene la definición de rutas para el resto de los roles de usuario, la cual es igual que para el usuario Administrador.

```
12 const routes: Routes = [  
13   { path: 'iniciar-sesion',  
14     component: LoginComponent  
15   },  
16   {  
17     path: 'registro',  
18     component: RegisterComponent  
19   },  
20   {  
21     path: 'olvide-mi-contraseña',  
22     component: PasswordResetComponent  
23   },  
24   {  
25     path: '404',  
26     component: NotFoundComponent  
27   },  
28   {  
29     path: 'admin',  
30     loadChildren: () => import(`./pages/administrator/administrator.module`).then(m => m.AdministratorModule),  
31     canActivate: [RoleGuard],  
32     data: {role: ["Administrador"]}  
33   },  
34 > { ...  
39   },  
40 > { ...  
45   },  
46 > { ...  
51   },  
52 > { ...  
57   },  
58   {path: '**', pathMatch: 'full', redirectTo: 'iniciar-sesion'},  
59 },  
60 ];
```

Código 2.9. EXPAPP-FRONTEND: Declaración de rutas en *AppRoutingModule*.

En el Código 2.10 se tiene la lógica del *guard* creado para la protección de rutas. Cuando un usuario navegue por la aplicación web se realizará el siguiente proceso para todas las rutas. Se extraen los datos del usuario que haya iniciado sesión del *localStorage* (línea 15) y se verifica que existan datos en la variable *currentUser* (línea 17), en el caso de que su valor sea *null* o *undefined*, quiere decir que no se ha iniciado sesión y se le redirige a la pantalla de inicio de sesión.

En el caso de que la variable *currentUser* tenga datos, significa que el usuario tiene una sesión activa y se procede a verificar que el rol del usuario sea igual al rol que protege la ruta que llega en la variable *data* (línea 18). Si es que son iguales la función tienen un valor de retorno igual a *true*, lo cual permite el acceso a la ruta. Caso contrario devuelve se redirige a una ruta predeterminada y se retorna *false*.

La ruta predeterminada en el caso de que el usuario no tenga acceso a una ruta es la del componente 404 que muestra al usuario un mensaje de que no se encontró la ruta.

```
1 import { Injectable } from '@angular/core';
2 import { CanActivate, ActivatedRouteSnapshot, UrlTree, Router } from '@angular/router';
3 import { Observable } from 'rxjs';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class RoleGuard implements CanActivate {
9
10  constructor(private router: Router){}
11
12  canActivate(next: ActivatedRouteSnapshot):
13  Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean | UrlTree {
14
15    const currentUser = JSON.parse(localStorage.getItem('user'));
16
17    if(currentUser){
18      if(next.data.role[0] == currentUser.role ){
19        return true;
20      }else{
21        this.router.navigate(['/404']);
22        return false;
23      }
24    }else{
25      this.router.navigate(['/login']);
26      return false;
27    }
28  }
29 }
```

Código 2.10. EXPAPP-FRONTEND: Configuración de *RoleGuard*.

En el Código 2.11 se tiene un ejemplo de cómo se ven los archivos *routing.module* para cada uno de los usuarios. Se va definiendo el *path* de cada una de las rutas y su respectivo componente. La ruta padre corresponde a *TutorComponent* desde el cual se redirige a otro componente según la ruta especificada.

```
1 import { Routes, RouterModule } from '@angular/router';
2 import { NgModule } from '@angular/core';
3 import { TutorComponent } from './tutor.component';
4 import { ProfileComponent } from 'src/app/Shared/profile/profile.component';
5 import { InternshipsActiveComponent } from './components/internships-active/internships-active.component';
6 import { InternshipsRecordComponent } from './components/internships-record/internships-record.component';
7 import { InternshipDetailsComponent } from './components/internship-details/internship-details.component';
8
9
10 const TUTOR_ROUTES: Routes = [
11   {
12     path: '', component: TutorComponent, children: [
13       {path: 'perfil', component: ProfileComponent},
14       {path: 'practicas-activas', component: InternshipsActiveComponent},
15       {path: 'historial-practicas', component: InternshipsRecordComponent},
16       {path: 'practica-detalles/:id:opcion', component: InternshipDetailsComponent},
17       {path: "", redirectTo:"perfil", pathMatch:"full"}
18     ]
19   },
20 ];
21
22 @NgModule ({
23   imports: [RouterModule.forChild(TUTOR_ROUTES)],
24   exports: [RouterModule]
25 })
26 export class TutorRoutingModule { }
```

Código 2.11. EXPAPP-FRONTEND: Declaración de rutas hijas para el usuario tutor.

2.3.3.3 Uso de librerías

Una librería es un archivo o conjunto de archivos que se utilizan durante el desarrollo de una aplicación web para añadir funcionalidad adicional en menor tiempo. En esta sección se describe la implementación y uso de todas las librerías utilizadas en EXPAPP.

2.3.3.3.1 Angular Material

Para usar los elementos de Angular Material se realiza la instalación, como se indicó en la creación del proyecto, y es necesario importar uno a uno al módulo que contiene el componente en donde se requiere usar. Esto puede presentar un problema debido a que vuelve un proceso repetitivo y los módulos se vuelven muy extensos al tener que importar en cada uno los elementos necesarios.

Para que el uso de Angular Material sea más sencillo y para tener un orden en los módulos del proyecto se creó uno que contiene todos los componentes de esta librería. Este módulo se importará en otros módulos en los cuales se requiera utilizar Angular Material.

El módulo creado se llama *angular-material.module.ts* y al inicio del archivo se importan los módulos de los componentes de Angular Material; en el Código 2.12 se muestra un ejemplo de cómo se importan.

```
25
26 // Buttons & Indicators
27 import { MatButtonModule } from '@angular/material/button';
28 import { MatButtonModule } from '@angular/material/button-toggle';
29 import { MatBadgeModule } from '@angular/material/badge';
30 import { MatChipsModule } from '@angular/material/chips';
31 import { MatIconModule } from '@angular/material/icon';
32 import { MatProgressSpinnerModule } from '@angular/material/progress-spinner';
33 import { MatProgressBarModule } from '@angular/material/progress-bar';
34
```

Código 2.12. EXPAPP-FRONTEND: Ejemplo de cómo importar componentes de Angular Material en un módulo de Angular.

Una vez que se importaron todos los componentes de Angular Material, se agrupan en un arreglo de tipo *any* como se muestra en el Código 2.13. Esta es una de las ventajas de Angular, TypeScript permite guardar datos en variables sin especificar el tipo de dato al que pertenecen.

```

51 const modules: any[] = [
52   MatAutocompleteModule,
53   MatCheckboxModule,
54   MatDatepickerModule,
55   MatFormFieldModule,
56   MatInputModule,
57   MatRadioModule,
58   MatSelectModule,
59   MatSliderModule,
60   MatSlideToggleModule,
61   MatNativeDateModule,
62   MatMenuModule,
63   MatSidenavModule,
64
65   MatToolbarModule,
66   MatCardModule,
67   MatDividerModule,
68   MatExpansionModule,
69   MatGridListModule,
70   MatListModule,
71   MatStepperModule,
72   MatTabsModule,
73   MatTreeModule,
74   MatButtonModule,
75   MatButtonModuleToggleModule,
76   MatBadgeModule,
77   MatChipsModule,
78
79   MatIconModule,
80   MatProgressSpinnerModule,
81   MatProgressBarModule,
82   MatBottomSheetModule,
83   MatDialogModule,
84   MatSnackBarModule,
85   MatTooltipModule,
86   MatPaginatorModule,
87   MatSortModule,
88   MatTableModule,
89   FormsModule,
90   ReactiveFormsModule
91 ];

```

Código 2.13. EXPAPP-FRONTEND: Arreglo que contiene los elementos que se importan desde Angular Material.

A continuación, se definen los *imports* (Código 2.14), que permiten traer componentes de otros módulos, en este caso de Angular Material, y los *exports* que permiten usarlos desde otros módulos de la aplicación.

```

97 @NgModule({
98   imports: [ ...modules ],
99   exports: [ ...modules ],
100  providers: [{provide: MatPaginatorIntl, useClass: MatPaginatorIntlSpanish}]
101 })
102 export class AngularMaterialModule { }
103

```

Código 2.14. EXPAPP-FRONTEND: Definición de *imports*, *exports* y *providers* de *AngularMaterialModule* (líneas 98-100).

En el Código 2.14 también se puede observar que existe la sección de *providers* (línea 100). Angular Material está en inglés por defecto y para cambiar los rótulos y colocarlos en español se creó una clase para personalizar el paginado de tablas de Angular Material.

En el Código 2.15 se tiene la clase creada tiene el nombre de *MatPaginatorIntlSpanish* que hereda de *MatPaginatorIntl*, que es una clase que Angular Material provee con ciertos atributos para personalizar la paginación de tablas, como *itemsPerLabel*, *nextPageLabel* y *previousPageLabel*, los cuales están definidos en las líneas 6-8.

En la línea 10 del Código 2.15 se tiene la función *getRangeLabel* la cual define la etiqueta para el rango de elementos dentro de la página actual de la tabla y la longitud de toda la lista.

```

1  import { Injectable } from '@angular/core';
2  import { MatPaginatorIntl } from '@angular/material/paginator';
3
4  @Injectable()
5  export class MatPaginatorIntlSpanish extends MatPaginatorIntl {
6      itemsPerPageLabel = 'Items por página';
7      nextPageLabel     = 'Siguiente página';
8      | previousPageLabel = 'Página anterior';
9
10     getRangeLabel = function (page, pageSize, length) {
11         if (length === 0 || pageSize === 0) {
12             return '0 de ' + length;
13         }
14         length = Math.max(length, 0);
15         const startIndex = page * pageSize;
16         const endIndex = startIndex < length ?
17             Math.min(startIndex + pageSize, length) :
18             startIndex + pageSize;
19         return startIndex + 1 + ' - ' + endIndex + ' de ' + length;
20     };
21 }

```

Código 2.15. EXPAPP-FRONTEND: Clase *MatPainatorIntlSpanish*.

Para utilizar los elementos de Angular Material se importa el módulo creado. En el Código 2.16 se muestra cómo se importa *AngularMaterialModule* en el *AppModule*, el cual es el módulo principal de cualquier aplicación de Angular. Una vez que se ha importado el módulo ya se puede hacer uso de los elementos de Angular en los componentes que se encuentren dentro del contexto de ese módulo.

```

21  import { AngularMaterialModule } from './angular-material/angular-material.module';
22  import { FlexLayoutModule } from '@angular/flex-layout';
23  import { PdfViewerModule } from 'ng2-pdf-viewer';
24
25  @NgModule({
26      declarations: [
27          AppComponent,
28          LoginComponent,
29          RegisterComponent,
30          PasswordResetComponent
31      ],
32      imports: [
33          BrowserModule,
34          AppRoutingModule,
35          AngularFireModule.initializeApp(environment),
36          AngularFireAuthModule,
37          AngularFireStoreModule,
38          AngularFireStorageModule,
39          BrowserAnimationsModule,
40          FormsModule,
41          AngularMaterialModule,
42          ReactiveFormsModule,
43          FlexLayoutModule,
44          PdfViewerModule
45      ],

```

Código 2.16. EXPAPP-FRONTEND: Importación de *AngularMaterialModule* en *AppModule*.

El uso de los elementos de Angular Material facilita la creación de componentes, en la página oficial se tienen todos los elementos y la manera de usarlos en archivos HTML y SCSS. En el Código 2.17 se tiene un ejemplo de uso para un botón para el inicio de sesión extraído del archivo *login.component.html*.

```
57 <mat-grid-tile>
58   <button mat-raised-button color="accent" (click)="login()">Iniciar sesión</button>
59 </mat-grid-tile>
```

Código 2.17. EXPAPP-FRONTEND: Implementación de botón de *Angular Material*.

2.3.3.3.2 *AngularFlexLayout*

Esta librería permite organizar los elementos de un archivo *HTML* en columnas y filas. Además, se puede definir la alineación del elemento (izquierda, centro o derecha) y cuánto espacio de la fila o columna va a utilizar. Para usar esta librería se requiere importar el módulo dentro del módulo que contiene el componente en dónde será consumida la librería. En las líneas 15 y 32 del Código 2.18 se muestra cómo se importa.

```
1  import { NgModule } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3  import { RouterModule } from '@angular/router';
4
5  import { SharedModule } from '../Shared/shared.module';
6  import { AngularMaterialModule } from '../angular-material/angular-material.module';
7
8  import { TutorComponent } from './tutor.component';
9  import { SidebarTutorComponent } from './components/sidebar-tutor/sidebar-tutor.component';
10 import { TutorRoutingModule } from './tutor.routing.module';
11 import { InternshipsActiveComponent } from './components/internships-active/internships-active.component';
12 import { InternshipsRecordComponent } from './components/internships-record/internships-record.component';
13 import { InternshipDetailsComponent } from './components/internship-details/internship-details.component';
14
15 import { FlexLayoutModule } from '@angular/flex-layout';
16 import { ChartsModule } from 'ng2-charts';
17
18 @NgModule({
19   declarations: [
20     TutorComponent,
21     SidebarTutorComponent,
22     InternshipsActiveComponent,
23     InternshipsRecordComponent,
24     InternshipDetailsComponent
25   ],
26   imports: [
27     CommonModule,
28     AngularMaterialModule,
29     SharedModule,
30     RouterModule,
31     TutorRoutingModule,
32     FlexLayoutModule,
33     ChartsModule
34   ]
35 })
36 export class TutorModule { }
37
```

Código 2.18. EXPAPP-FRONTEND: Ejemplo de importación de *FlexLayoutModule* en *TutorModule* (línea 15 y 32).

En el Código 2.19 se muestra un ejemplo de cómo utilizar *FlexLayout*. Se requiere usar un `<div></div>` que es un elemento de HTML, en el cual se especifica si se va a organizar los elementos en columna o fila, la alineación y el espaciado [49]. todos los elementos que se coloquen dentro del *div* configurado seguirán la alineación y orden especificados.

```
3  <div class="headerContainer">
4    <div fxLayout="row" fxLayoutAlign="space-between center" fxLayoutGap="5px">
5      <h1>Práctica Preprofesional</h1>
6      <button mat-raised-button color="primary" type="button" (click)="backPage()">
7        <mat-icon>reply</mat-icon> Atrás
8      </button>
9    </div>
10 </div>
```

Código 2.19. EXPAPP-FRONTEND: Ejemplo de uso de *FlexLayout* (línea 4).

2.3.3.3 Ng2-charts

Esta librería permite integrar a la aplicación web gráficos para análisis de datos o estadísticas, pueden ser: lineal, barras, pastel, entre otros [50]. *Ng2-charts* se la utilizó para cumplir con el Rq12 y para poder usarla debe ser importada en el módulo que contiene el componente que tendrá las gráficas.

Para la implementación de las gráficas se requiere declarar variables que determinan características de la gráfica como el tipo, colores, leyendas, entre otras. También se debe guardar en una variable los datos que se van a mostrar. Estas variables deben estar en el archivo *.ts* que contiene la lógica del componente en donde se va a implementar las gráficas. En el Código 2.20 se tiene la configuración necesaria.

```
17 public pieChartType: ChartType = 'pie';
18 public chartLegend = true;
19
20 public pieChartOptions: ChartOptions = {
21   responsive: true,
22   legend: {
23     position: 'top',
24     labels: {
25       fontColor: 'black',
26       fontSize: 20
27     }
28   },
29   plugins: {
30     datalabels: {
31       formatter: (value, ctx) => {
32         const label = ctx.chart.data.labels[ctx.dataIndex];
33         return label;
34       },
35     },
36   }
37 };
38
39 public pieChartLabelsStudent1: Label[] = [];
40 public pieChartDataStudent1: number[] = [];
```

Código 2.20. EXPAPP-FRONTEND: Declaración de variables para la configuración de *Ng2-charts*.

La variable *pieChartDataStudent1* tendrá los datos que se van a observar en la gráfica, y son traídos desde la base de datos mediante un servicio, esto se describe en la sección 2.3.3.4.

Cuando ya se haya personalizado las variables, se modifica el archivo *.html* del componente, en el Código 2.21 se tiene la configuración de la etiqueta *<canvas>*.

```
10 <div class="parent">
11   <div class="title">
12     <h1 style="color: ■ black;">Estudiantes en Prácticas por Carrera</h1>
13   </div>
14   <div style="display: block" class="child">
15     <canvas baseChart [data]="pieChartDataStudent1"
16       [labels]="pieChartLabelsStudent1"
17       [chartType]="pieChartType"
18       [options]="pieChartOptions"
19       [colors]="pieChartColors"
20       [legend]="ChartLegend">
21     </canvas>
22   </div>
23   <br>
24 </div>
```

Código 2.21. EXPAPP-FRONTEND: Código HTML requerido para la implementación de *Ng2-charts*.

2.3.3.3.4 Pdfmake

Esta librería permite generar, descargar o imprimir archivos en formato PDF. Para su uso se requiere importarla en el módulo y en el componente en el cual se va a trabajar, como se muestra en el Código 2.22.

```
13 import * as pdfMake from 'pdfmake/build/pdfmake';
14 import * as pdfFonts from 'pdfmake/build/vfs_fonts';
15 (<any>pdfMake).vfs = pdfFonts.pdfMake.vfs;
```

Código 2.22. EXPAPP-FRONTEND: Importación de Pdfmake en el componente que se va a utilizar.

En la aplicación se requiere crear documentos en formato PDF para visualizarlos y/o exportarlos para los informes de prácticas preprofesionales y para el CV del estudiante. En el Código 2.23 se tiene dos funciones.

La primera función abre un archivo en PDF en otra ventana del navegador utilizando la sentencia *pdfmake.createPdf(documentDefinition).open()* y la segunda función descarga el

archivo `pdfmake.createPdf(documentDefinition).download()`. El método `getDocumentDefinition` es llamado en ambas funciones (líneas 204 y 209) para generar la estructura del archivo que se abrirá o descargará.

```
203     openPdf(){
204         const documentDefinition = this.getDocumentDefinition();
205         pdfMake.createPdf(documentDefinition).open();
206     }
207
208     exportar(){
209         const documentDefinition = this.getDocumentDefinition();
210         pdfMake.createPdf(documentDefinition).download();
211     }
```

Código 2.23. EXPAPP-FRONTEND: Método para abrir documento PDF en otra pestaña del navegador (línea 203-206) y método para descargar documento PDF (líneas 209-211).

La función `getDocumentDefinition` devuelve el contenido del archivo. La librería brinda el formato en que se debe definir los elementos que deben colocarse dentro de un arreglo que tiene el nombre de `content: []`. Los elementos son objetos de tipo *JSON* y están definidos dentro de llaves `{ }`. Cada elemento corresponde a una fila del archivo y sus atributos pueden personalizar tipo de letra, estilo, ancho que ocupa, entre otros.

Dentro de cada columna se pueden definir columnas para organizar el contenido, también se puede insertar tablas e imágenes. En el Código 2.24 se tiene un ejemplo de cómo se definen elementos del encabezado del archivo correspondiente al Informe Inf3.

En el Código 2.25 se muestran configuraciones adicionales del archivo. Después del contenido también se puede tener un objeto llamado *info* en donde se define información como nombre, autor y asunto del documento que se creará (líneas 405-409). Otro objeto que puede definirse es una biblioteca de estilos (líneas 410-433), en donde se tendrá estilos personalizados los cuales son utilizados en el *content* como se observa en las líneas 230, 235 y 240 del Código 2.24.


```

213  getDocumentDefinition() {
214      return {
215          content: [
216              //Encabezado
217              {
218                  columns: [
219                      {
220                          image: 'epn',
221                          width: 60,
222                          height: 60,
223                          alignment : 'left',
224                          with: '4%'
225                      },
226                  ],
227                  [
228                      {
229                          text: 'ESCUELA POLITÉCNICA NACIONAL',
230                          style: 'header'
231                      },
232                      {
233                          text: 'Facultad de Ingeniería Eléctrica y Electrónica',
234                          fontSize: 11,
235                          style: 'header'
236                      },
237                      {
238                          text: 'INFORME DE MITAD DE PERÍODO DE PRÁCTICAS PREPROFESIONALES',
239                          margin: [0, 0, 0, 10],
240                          style: 'header'
241                      }
242                  ]
243              ]
244          },
245

```

Código 2.24. EXPAPP-FRONTEND: Función *getDocumentDefinition*.

```

405      info: {
406          title: 'INFORME MITAD PERIODO TUTOR',
407          author: this.student.name,
408          subject: 'INFORME MITAD PERIODO TUTOR'
409      },
410      styles: {
411          header: {
412              fontSize: 11,
413              bold: true,
414              alignment : 'center',
415              margin: [0, 0, 0, 5]
416          },
417          header2: {
418              fontSize: 11,
419              bold: true,
420              margin: [0, 20, 0, 20]
421          },
422
423          subheader: {
424              alignment: 'left',
425              fontSize: 15,
426              margin: [5, 5, 5, 5],
427          },
428          tableHeader: {
429              bold: true,
430              fontSize: 11,
431              alignment: 'left',
432          }
433      },
434      images: {
435          epn: "https://firebasestorage.googleapis.com/v0/b/prototipoexp.appspot.com/
436      }
437  }

```

Código 2.25. EXPAPP-FRONTEND: Continuación de la función *getDocumentDefinition*.

2.3.3.3.5 Sweetalert2

Esta es una librería que permite integrar de manera rápida cuadros emergentes de JavaScript [51]. Después de la instalación se requiere importar la función *Swal* en el componente en el cual se requiere utilizar como se muestra en la línea 6 del Código 2.26.

```
1 import { Component } from '@angular/core';
2 import { User } from '../../models/users.interface';
3 import { Validators, FormGroup, FormBuilder } from '@angular/forms';
4 import { AuthService } from '../../services/auth.service';
5 import { Router } from '@angular/router';
6 import Swal from 'sweetalert2';
7
8 @Component({
9   selector: 'app-login',
10  templateUrl: './login.component.html',
11  styleUrls: ['./login.component.scss']
12 })
13 > export class LoginComponent { ...
80 }
```

Código 2.26. EXPAPP-FRONTEND: Importación de *Swal* en el componente LoginComponent.

Para desplegar un cuadro emergente únicamente se requiere llamar a la función *Swal.fire()* y dentro de ella enviar un objeto que contiene características como: icono, texto, título, dimensiones, entre otras. En el Código 2.27 se tiene la definición de las características del cuadro emergente en las líneas 62 a la 67 y en la línea 68 se utiliza la función *showLoading()*, la cual despliega el cuadro emergente definido más un ícono de carga como se observa en la Figura 2.30.

```
62 ✓ Swal.fire({
63   allowOutsideClick: false,
64   icon: 'info',
65   text: 'Espere por favor...',
66   heightAuto: false
67 });
68 Swal.showLoading();
```

Código 2.27. EXPAPP-FRONTEND: Ejemplo de uso de *Swal* para mostrar alertas.

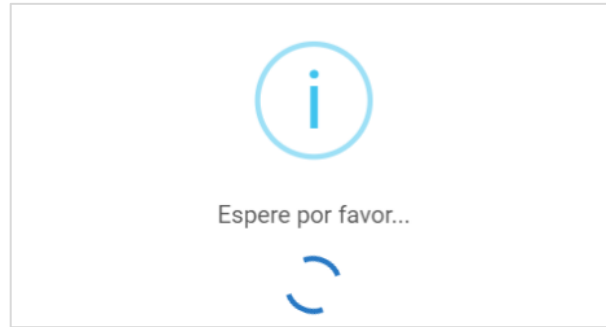


Figura 2.30. EXPAPP-FRONTEND: Resultado de la alerta creada en el Código 2.27.

También se puede tener cuadros emergentes que requieren la acción del usuario, como por ejemplo el ingreso de texto o tener una respuesta mediante botones de confirmación o negación (Figura 2.31). En el Código 2.28 se tiene la configuración de un cuadro emergente más complejo, en el cual se requiere la confirmación del usuario. Si el usuario presiona el botón de confirmación se realizan ciertas acciones (línea 109) y si presiona el botón para cancelar se ejecutan otras.

```
99     Swal.fire({
100       title: 'Atención',
101       text: `Este anuncio será desactivado, desea continuar?`,
102       icon: 'warning',
103       showCloseButton: true,
104       showCancelButton: true,
105       cancelButtonText: 'Cancelar',
106       confirmButtonText: `Sí, desactivar anuncio`,
107       heightAuto: false
108     }).then((result)=> {
109 >     if (result.isConfirmed) { ...
121 >     } else if (result.dismiss == Swal.DismissReason.cancel) { ...
128     }
129 });
```

Código 2.28. EXPAPP-FRONTEND: Uso de alerta de *Swal* con botón de confirmación.

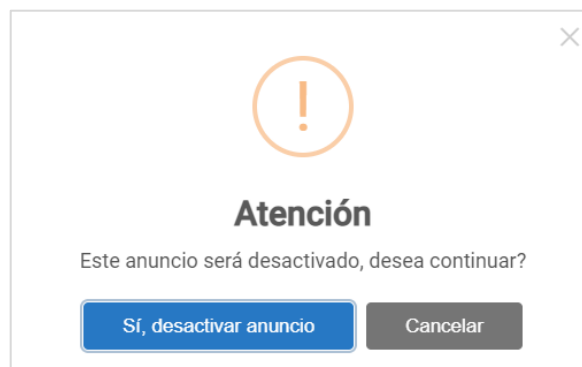


Figura 2.31. EXPAPP-FRONTEND: Resultado de la alerta creada en el Código 2.28.

2.3.3.3.6 AngularFire

Al crear el proyecto mediante Angular CLI se crea la estructura del proyecto que incluye los archivos de configuración, entre ellos está el archivo *environment.prod.ts*. En este archivo se debe crear una constante de tipo JSON que tiene atributos para acceder a la aplicación de Firebase. En este paso se requiere la información de la aplicación de Firebase creada en el ANEXO A y en el Código 2.29 se tiene cómo debe quedar la constante.

```
1  export const environment = {
2    production: true,
3    apiKey: "AIzaSyBPG_gvEbhPndMoDcOIY0i4SV9L2bsRrdE",
4    authDomain: "prototipoexp.firebaseio.com",
5    databaseURL: "https://prototipoexp.firebaseio.com",
6    projectId: "prototipoexp",
7    storageBucket: "prototipoexp.appspot.com",
8    messagingSenderId: "37394215082",
9    appId: "1:37394215082:web:02ec67cc977a2eca3020d3"
10 };
```

Código 2.29. EXPAPP-FRONTEND: Configuración del archivo *environment.prod.ts* para uso de Firebase en el proyecto de Angular.

```
20 import {AngularFireModule} from '@angular/fire';
21 import {AngularFireAuthModule} from '@angular/fire/auth';
22 import {AngularFirestoreModule} from '@angular/fire/firestore';
23 import {AngularFireStorageModule} from '@angular/fire/storage';
24
25 @NgModule({
26   declarations: [
27     AppComponent,
28     LoginComponent,
29     RegisterComponent,
30     PasswordResetComponent
31   ],
32   imports: [
33     BrowserModule,
34     AppRoutingModule,
35     AngularFireModule.initializeApp(environment),
36     AngularFireAuthModule,
37     AngularFirestoreModule,
38     AngularFireStorageModule,
39     BrowserModule,
40     FormsModule,
41     AngularMaterialModule,
42     ReactiveFormsModule,
43     FlexLayoutModule,
44     PdfViewerModule
45   ],
46   providers: [RoleGuard],
47   bootstrap: [AppComponent]
48 })
49 export class AppModule { }
```

Código 2.30. EXPAPP-FRONTEND: Módulos importados en AppModule para uso de *AngularFire*.

En el AppModule, se importa los Módulos que se requieren de *AngularFire* como se muestra en las líneas 20 a la 23 del Código 2.30. *AngularFireAuth* para hacer uso de Authentication de Firebase, *AngularFirestore* para el manejo de los datos con Firestore y *AngularFireStorage* para la carga y descarga de documentos. En el Código 2.30 línea 35 se tiene la inicialización del proyecto y la vinculación del proyecto de Angular con *Firebase*, para esto se utiliza la variable *environment* del Código 2.29.

Con los pasos realizados previamente ya es posible el uso de *AngularFire* en el proyecto de Angular, en la siguiente sección se explica cómo se implementa esta librería para el uso de Firebase.

2.3.3.4 SERVICIOS

En esta sección se presentan las funciones más importantes de los servicios creados para EXPAPP. Se tiene cuatro servicios: *auth.service* para la gestión de los usuarios y sus sesiones, *document.service* en donde se realiza el manejo de archivos, *firestore.service* que tiene funciones para realizar el CRUD de documentos y colecciones en la base de datos, y *storage.service* que gestiona el almacenamiento de archivos en la nube.

En los servicios principalmente se realiza la comunicación con Firebase, para esto se utilizó la librería *AngularFire*, que requiere inyectar en el constructor del servicio la dependencia como se muestra en el Código 2.31 para el uso de los productos de Firebase. Por otra parte, Firebase ya se encuentra instalado y también es posible importar todos los módulos directamente como se muestra en el Código 2.32.

```
37     constructor(private afs: AngularFirestore,  
38                private fns: AngularFireFunctions,  
39                private storage: AngularFireStorage,  
40                public afAuth: AngularFireAuth) {  
41     }
```

Código 2.31. EXPAPP-FRONTEND: Inyección de dependencias de *AngularFire* en los servicios.

```
9  
10    import * as firebase from "firebase/app";  
11
```

Código 2.32. EXPAPP-FRONTEND: Importación de todos los módulos de Firebase.

En las siguientes secciones se presentan funciones implementadas en los servicios para el uso de los productos de Firebase.

2.3.3.4.1 Authentication

La autenticación de usuarios en EXPAPP se implementó por medio de Firebase Authentication, que permite gestionar las sesiones con diferentes métodos como SMS, autenticación por medio de redes sociales, entre otros. Para EXPAPP se ha escogido el método que requiere correo y contraseña.

Para el inicio de sesión se tiene una función llamada *signIn*, la cual se encuentra en el Código 2.33. Esta función es de tipo asincrónica, es decir, que no se puede realizar ninguna otra acción hasta que se termine de ejecutar. Recibe dos parámetros de entrada de tipo *string* que son el correo y la contraseña.

En la línea 304 del Código 2.33 se tiene el uso de *AngularFireAuth*, que como se observa, es muy sencillo porque únicamente se llama a la función *signInWithEmailAndPassword* y se envía las credenciales. Esta acción al tener “*await*” debe terminarse de ejecutar para continuar. Esta sentencia se encuentra dentro de un *try-catch*, y en el caso de producirse algún error en la autenticación se configura una alerta que le indica al usuario un mensaje de error con la librería *SweetAlert* (líneas 330-339).

Si es que se realizó la autenticación de manera correcta en la línea 305 se trae los datos del usuario mediante una función del servicio *firestore.service* que devuelve un objeto de tipo observable (Código 2.36). En la línea 306 del Código 2.33 se realiza la subscripción al objeto y en la línea 308 se realiza la verificación de la existencia del usuario. Si es que el usuario no existe despliega un mensaje de error. Si es que el usuario existe en la base de datos se guardan los datos en el *localStorage* (línea 312) con el fin de tener constancia de la sesión activa y los datos del usuario que inició sesión.

Finalmente, en la línea 314 del Código 2.33 se utiliza una función para navegar hacia la ruta principal del usuario según el rol del usuario. En el ANEXO C se puede encontrar las funciones llamadas en la función *signIn*.

```

302 async signIn(user: User, password: string ) {
303   try {
304     const result = await this.afAuth.signInWithEmailAndPassword(user.email, password);
305     const userObs = this.firestoreService.findUserByEmail(user.role, user.email);
306     this.subscripton = userObs.subscribe(resp => {
307       {
308         if (resp[0]) {
309           Swal.close();
310           this.subscripton.unsubscribe();
311           this.userData = resp[0];
312           this.firestoreService.updateUserLocalStorage(this.userData);
313           this.userAuth = JSON.parse(localStorage.getItem('user'));
314           this.routeUserByRole(resp[0].role);
315         } else {
316           Swal.close();
317           Swal.fire({
318             title: 'Usuario incorrecto',
319             text: 'Este usuario no pertenece a ' + user.role,
320             icon: 'error',
321             confirmButtonText: 'Ok',
322             heightAuto: false
323           });
324           this.firestoreService.updateUserLocalStorage(null);
325           this.userAuth = JSON.parse(localStorage.getItem('user'));
326         }
327       }
328     });
329     return user[0];
330   } catch (error) {
331     Swal.close();
332     Swal.fire({
333       title: 'Usuario incorrecto',
334       text: 'Este usuario no tiene una cuenta creada o los datos ingresados son incorrectos',
335       icon: 'error',
336       confirmButtonText: 'Ok',
337       heightAuto: false
338     });
339   }
340 }

```

Código 2.33. EXPAPP-FRONTEND: Uso de *AngularFireAuth*.

Para la recuperación de contraseña se utiliza directamente las funciones de Firebase que se importaron en el Código 2.32. En la línea 389 del Código 2.34 se utiliza *auth().currentUser* para obtener el objeto usuario que tiene la sesión activa y en la línea 390 se emplea la función *updatePassword()* que recibe como parámetro la nueva contraseña; esta función devuelve una promesa que al resolverse de manera exitosa despliega en pantalla el mensaje configurado en la línea 392 o si existe algún error en la actualización de la contraseña se atrapa la excepción y se despliega en pantalla el mensaje configurado en la línea 395.

```

388     updateUserPassword(newPassword: string){
389         let user = firebase.auth().currentUser;
390         user.updatePassword(newPassword)
391         .then(function() {
392             Swal.fire('Contraseña actualizada!', '', 'success');
393         }).catch(function(error) {
394             console.log(error);
395             Swal.fire('Ha ocurrido un error!', '', 'error');
396         });
397     }

```

Código 2.34. EXPAPP-FRONTEND: Función para cambio de contraseña.

2.3.3.4.2 Cloud Storage

Cloud Storage de Firebase permite el almacenamiento de archivos en la nube. En EXPAPP se utiliza este producto para el almacenamiento de las fotos de perfil de los usuarios. Para la implementación de la carga de imágenes se utilizó Firebase directamente, para esto se requiere importar los productos de Firebase como se muestra en el Código 2.32.

En el Código 2.35 se tiene la función *uploadToFirebase* que recibe el archivo que se va a cargar, en este caso la foto de perfil, y el usuario. En la línea 20 se crea una constante que contiene la referencia hacia Firebase Storage y en la línea 21 se coloca el valor `true` en el atributo del archivo *file.isUploading* para verificar el estado de la carga.

En las líneas 22 a la 28 del Código 2.35 se configura una alerta con *SweetAlert* que informa al usuario que la foto de perfil se está cargando. En la línea 30 se define una constante que de tipo *uploadTask* que representa el proceso de carga de un archivo a Firebase y con la referencia a storage, definida anteriormente, se crea una ruta hija con el método *child* el cual recibe la ruta en dónde se ubicará el archivo (línea 31) y con el método *put* se envía el archivo (línea 32).

En la línea 34 del Código 2.35 se utiliza el método *on* para escuchar eventos relacionados a la carga del archivo. Este método tiene cuatro parámetros de entrada los cuales se encuentran encerrados en recuadros y se explican a continuación:

- Primer parámetro (línea 35): representa el evento que se dispara, en este caso el evento es el cambio de estado del proceso de carga.
- Segundo parámetro (líneas 37): es una función de tipo *callback* que se ejecuta mientras se carga el archivo, en este caso se tiene un valor *null* porque no se requiere su uso.

- Tercer parámetro (línea 37): función de tipo *callback* que se ejecuta en el caso de que se genere un error al cargar el archivo.
- Cuarto parámetro (líneas 38 - 50): función de tipo *callback* que se ejecuta una vez que la carga del archivo ha finalizado. En esta función se imprime un mensaje que indica que la imagen se ha cargado de forma correcta (línea 39) y se obtiene la URL de la imagen (línea 41). La URL se guarda en el campo *photo* del usuario (línea 43) y se actualiza en Firestore los datos del usuario. Finalmente, se detiene la alerta de carga que se configuró con *SweetAlert*.

```

18 private RESOURCES_FOLDER = 'profilePhotos';
19 uploadToFirebase( file: any, user: any) {
20   const storageRef = firebase.storage().ref();
21   file.isUploading = true;
22   Swal.fire({
23     allowOutsideClick: false,
24     icon: 'info',
25     text: 'Cargando foto de perfil...',
26     heightAuto: false
27   });
28   Swal.showLoading();
29
30   const uploadTask: firebase.storage.UploadTask = storageRef
31     .child(`${this.RESOURCES_FOLDER}/${user.uid}`)
32     .put(file);
33
34   uploadTask.on(
35     firebase.storage.TaskEvent.STATE_CHANGED,
36     null,
37     (error) => console.error('Error al subir', error),
38     () => {
39       console.log('Imagen cargada correctamente');
40       uploadTask.snapshot.ref
41         .getDownloadURL()
42         .then((downloadURL) => {
43           user.photo = downloadURL;
44           this.firestoreService.updateData(user);
45         })
46         .then(() => {
47           Swal.close();
48           file.isUploading = false;
49         });
50     });
51 }

```

Código 2.35. EXPAPP-FRONTEND: Uso de *Firebase storage*.

2.3.3.4.3 Firestore

Firestore es la base de datos que se utiliza en EXPAPP, se definió los campos que tendrán los documentos de cada colección con el fin de mantener un estándar y orden. El esquema de la base de datos se lo puede encontrar en la sección 2.2.1.

Para encontrar un registro en la base de datos se apunta a la colección en que se quiere buscar y se coloca la condición. En el Código 2.36 se tiene la función *findUserByEmail*, que recibe el rol y el *email* del usuario que se quiere buscar y devuelve un observable que contiene el resultado de la búsqueda.

En la línea se apunta a la referencia de la colección que tiene el nombre del rol del usuario y se coloca la condición con la sentencia *where*. En las líneas 95 a la 101 se mapea el resultado obtenido con la interfaz *User*.

```
93 findUserByEmail(role: string, email: string){
94   let observable$ = this.afs.collection(role, ref => ref.where('email', '==', email)).snapshotChanges().pipe(
95     map( changes => {
96       return changes.map(action => {
97         const data = action.payload.doc.data() as User;
98         data.uid = action.payload.doc.id;
99         return data;
100       });
101     });
102   );
103   return observable$;
104 }
```

Código 2.36. EXPAPP-FRONTEND: Uso de *AngularFirestore* para traer registros de la base de datos.

Para actualizar, guardar o borrar algún registro de la base de datos únicamente se debe apuntar a la referencia de la colección y el documento y utilizar las funciones *update*, *set* o *delete* respectivamente. En el Código 2.37 se muestra una función que actualiza registros de la base de datos.

```
461 updateData(user: any){
462   this.afs.collection(user.role).doc(user.uid).update(user)
463   .then(function() {
464     console.log("Document successfully updated!");
465   }).catch(function(error) {
466     console.error("Error updating document: ", error);
467   });
468 }
469 }
```

Código 2.37. EXPAPP-FRONTEND: Uso de *AngularFirestore* para actualizar registros de la base de datos.

2.3.3.4.4 Functions

Para utilizar las funciones del *backend* se deben realizar llamadas HTTP. En el Código 2.38 se tiene el ejemplo para el envío de correo electrónico cuando se crea una cuenta de usuario. En la línea 383 se define la llamada a la función *httpsCallable* de *AngularFireFunctions* y se envía como parámetro el nombre de la función que se requiere activar. En la línea 384 se define la variable *data\$* que es de tipo observable en la cual se envían los parámetros necesarios y en la línea 385 se realiza la subscripción la cual imprime el resultado que se envía desde *Functions* al activar la función.

```
382  sendInvitationEmail(email: string, password: string, name: string, role: string) {  
383      const callable = this.fns.httpsCallable('sendEmailInvitation');  
384      let data$ = callable({ email, password, name, role });  
385      data$.subscribe(data => console.log(data));  
386  }
```

Código 2.38. EXPAPP-FRONTEND: Uso de *AngularFireFunctions*.

3 RESULTADOS Y DISCUSIÓN

3.1 PRUEBAS DE FUNCIONAMIENTO

En esta sección se presentan los resultados de las pruebas que se realizaron para validar el correcto funcionamiento de las funcionalidades más importantes de cada uno de los roles. Las pruebas se realizaron en el ambiente local o de desarrollo cuya ruta es: <http://localhost:4200>.

En esta fase se simuló el proceso que se sigue para completar una práctica preprofesional; esto implica desde la creación de usuarios, creación de anuncios, el inicio de una práctica preprofesional y su finalización mediante el seguimiento de la práctica por medio de los informes.

A lo largo del flujo del proceso de prácticas preprofesionales todos los roles de usuario participan; por esta razón, se especifica con letras mayúsculas en cada figura de esta sección el nombre del rol que realiza la acción.

3.1.1 CREACIÓN Y CONFIGURACIÓN DE USUARIOS

Para que el proceso de prácticas preprofesionales pueda iniciar es necesaria la creación de al menos un usuario Administrador, una Empresa con un Supervisor y un Tutor. El Administrador Técnico es quien crea estos usuarios, por esta razón, se creó previamente un usuario con este rol.

La creación de usuarios es similar a pesar de que sean diferentes roles, por lo cual, se presenta únicamente los resultados de la prueba de la creación del usuario Administrador. En la Figura 3.1 se visualiza la página para la creación de un nuevo usuario cuya ruta es <http://localhost:4200/admin-tecnico/nuevo-usuario/Administrador>. Esta página presenta un formulario en donde el usuario Administrador Técnico tiene que ingresar la información del nuevo usuario (Nombre completo, correo electrónico de la EPN y contraseña) y presionar en el botón “**Crear usuario**”.

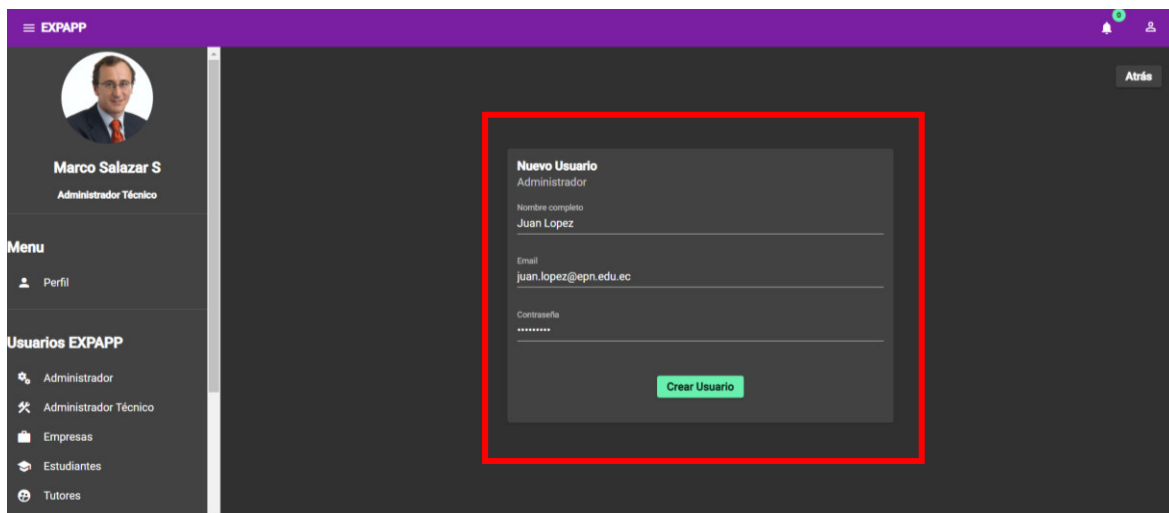


Figura 3.1. ADMINISTRADOR TÉCNICO: Creación de usuario Administrador.

La Figura 3.2 contiene el mensaje que se muestra cuando la creación del usuario es exitosa. El único error que se puede generar en la creación de un usuario es que ya exista una cuenta creada con el correo electrónico ingresado.

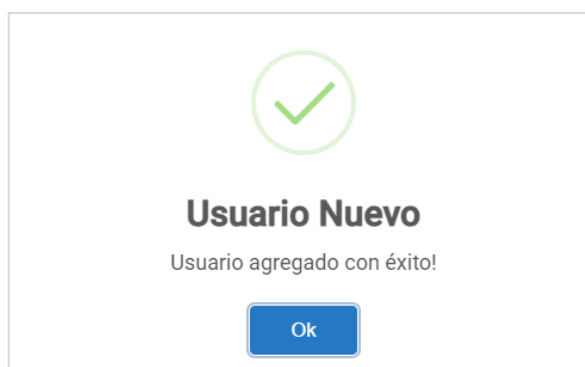


Figura 3.2. ADMINISTRADOR TÉCNICO: Mensaje obtenido tras crear el usuario Administrador.

Cuando se crea una cuenta en EXPAPP, el nuevo usuario recibe un correo electrónico tal como se muestra en la Figura 3.3, en donde recibe las credenciales con las que puede iniciar sesión en la página web.

Saludos Juan Lopez,

Te damos la bienvenida a EXPAPP!

Tu usuario tiene el rol de: Administrador y tus credenciales son:

Email: juan_lopez@epn.edu.ec
Contraseña: 123456789

Ingresa mediante el siguiente link: <https://prototipoexp.web.app/iniciar-sesion>.

Por seguridad cambia tu contraseña cuando inicies sesión!

Saludos,
EXPAPP Team

Figura 3.3. ADMINISTRADOR: Correo electrónico recibido al crearse la cuenta de usuario.

Para la creación de una cuenta de Estudiante se agregaron los datos en la sección “**Estudiantes FIEE**” que se encuentra en la ruta <http://localhost:4200/admin-tecnico/estudiantes-fiee> del rol Administrador Técnico (Figura 3.4). Para cargar los datos se presiona el botón “**Importar**” y se selecciona un archivo de tipo CSV con datos del estudiante: número único, nombre, cédula, email, carrera, créditos aprobados y teléfono.

Número Único	Nombre	Cédula	Email	Carrera	Créditos aprobados	Teléfono
201420192	Alejandra Nohella Cruz Aldean	1104744998	alejandra.cruz@epn.edu.ec	Ingeniería Eléctrica	180	0986644374
201520518	Mayte Soledad Baez Barragan	1726076464	maybaez_97@hotmail.com	Ingeniería Electrónica y Control	200	098396586
201520136	Omar Sebastian Carrillo Teran	1004339790	omar.carrillo@epn.edu.ec	Ingeniería en electrónica y redes de información	226	0998916523

Figura 3.4. ADMINISTRADOR TÉCNICO: Listado de estudiantes.

Cuando los datos del estudiante se hayan cargado en la base de datos ya es posible para el estudiante la creación de su cuenta, para esto se ingresa en la pestaña de registro, en la ruta <http://localhost:4200/registro>, que contiene el formulario que se encuentra en la Figura 3.5, se ingresa los campos solicitados y se presiona “**Crear cuenta**”.

Figura 3.5. ESTUDIANTE: Formulario de registro de Estudiante en EXPAPP.

Para verificar las validaciones implementadas se probó el ingreso de un correo electrónico de un estudiante que no haya sido registrado en la base de datos y también el correo electrónico que ya se encuentra en uso, el sistema no permite la creación de la cuenta y se muestra los mensajes de error que se observan en la Figura 3.6 a y b respectivamente.



Figura 3.6. ESTUDIANTE: a) Mensaje de error en la creación de la cuenta Estudiante cuando no se ha cargado los datos a la base. b) Mensaje de error cuando el correo electrónico ya se encuentra en uso.

Una vez que se tiene creada la cuenta del estudiante es necesaria la creación de su CV. Se probó esta funcionalidad accediendo a la pantalla, en la ruta <http://localhost:4200/estudiante/perfil> en la pestaña “CV-Maker” que se visualiza en la Figura 3.7. Se ingresa todos los datos requeridos en el formulario y finalmente se presiona

el botón “**Guardar**”. Al presionar los botones “**Exportar como PDF**” o “**Ver PDF**” se genera el CV con un formato predeterminado que se observa en la Figura 3.8.

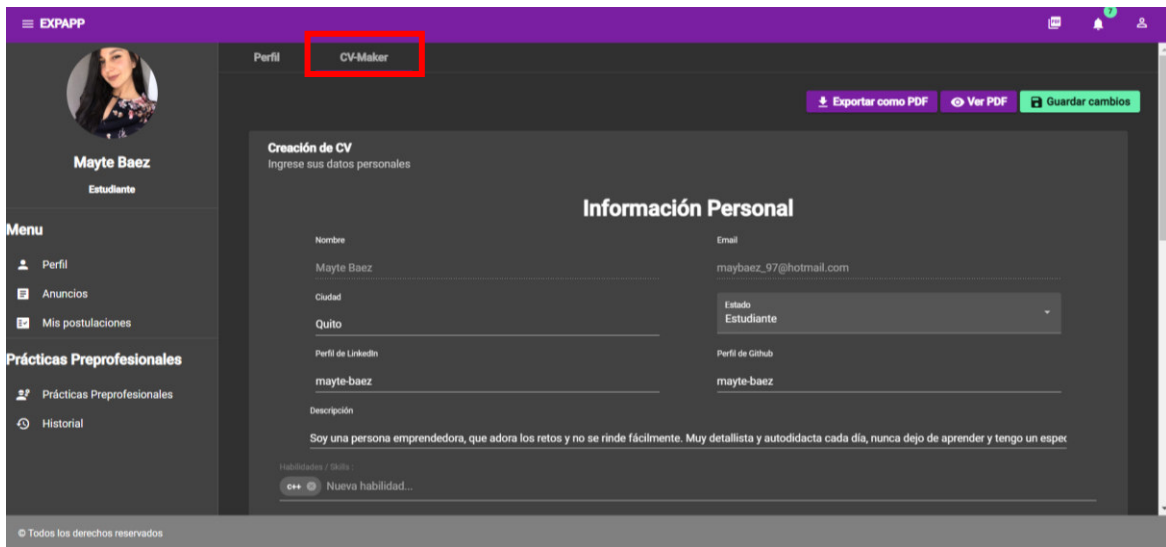


Figura 3.7. ESTUDIANTE: Creación del CV del estudiante.

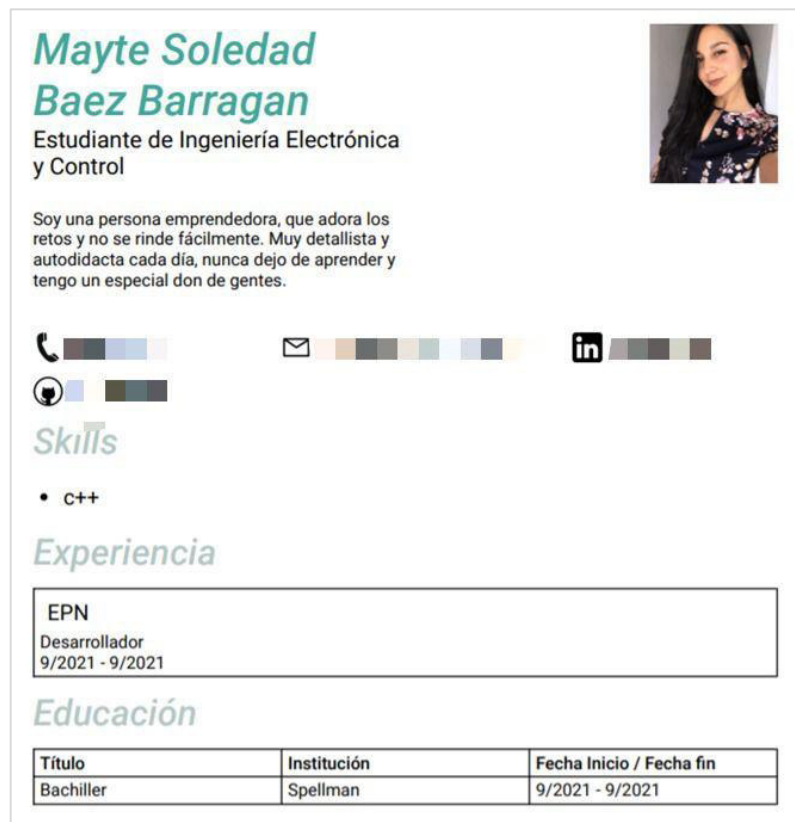


Figura 3.8. ESTUDIANTE: Resultado del CV generado en formato PDF.

3.1.2 GESTIÓN DE ANUNCIOS

Después de la creación de los usuarios de cada rol, se probó la creación de anuncios desde la cuenta del Supervisor Empresa en la ruta <http://localhost:4200/empresa/nuevo-anuncio>. En esta ruta se tiene un formulario que se observa en la **Figura 3.9** con los datos necesarios para la creación del anuncio; después de ingresar la información necesaria se crea el anuncio dando clic en “**Crear anuncio**”.

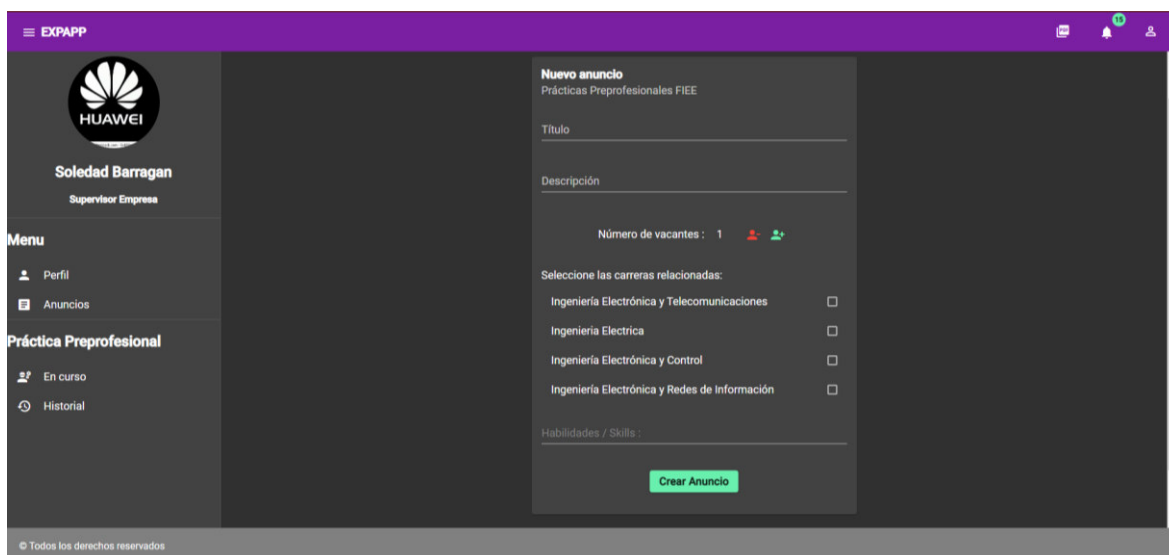


Figura 3.9. SUPERVISOR EMPRESA: Formulario para la creación de un anuncio.

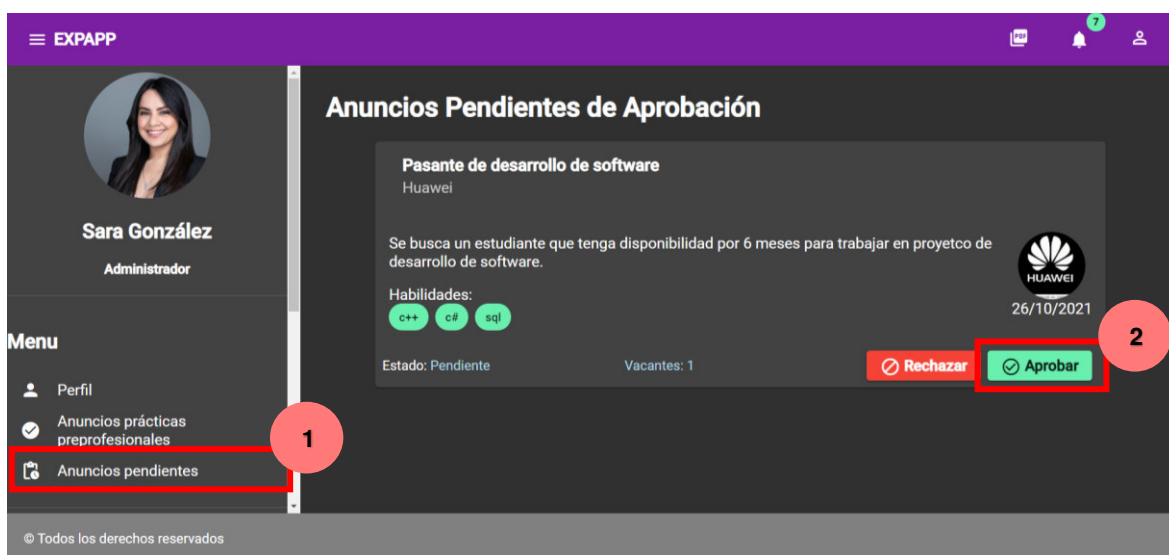


Figura 3.10. ADMINISTRADOR: Pestaña de anuncios pendientes de aprobar.

Cuando se crea el anuncio por el Supervisor Empresa se requiere la aprobación por parte del Administrador. En pestaña “**Anuncios pendientes**” que se encuentra en la Figura 3.10,

en la ruta <http://localhost:4200/admin/anuncios-no-publicados>, el Administrador tiene la opción de aprobar o rechazar el anuncio; en este caso se probó la aprobación del anuncio dando clic en el botón “**Aprobar**”.

Una vez que el anuncio ha sido aprobado ya será visible para los estudiantes de las carreras que hayan sido configuradas en el anuncio. En la ruta <http://localhost:4200/estudiante/anuncios>, el estudiante puede ver los anuncios que se encuentran activos.

Se probó la postulación de un Estudiante dando clic sobre el botón “**Postular**” que se muestra en la descripción de los anuncios. Cuando se presiona este botón aparece en pantalla la advertencia que visualiza en la Figura 3.11. Finalmente, se presiona en el botón “**Sí, enviar.**” para que el estudiante sea agregado a la lista de candidatos al puesto.

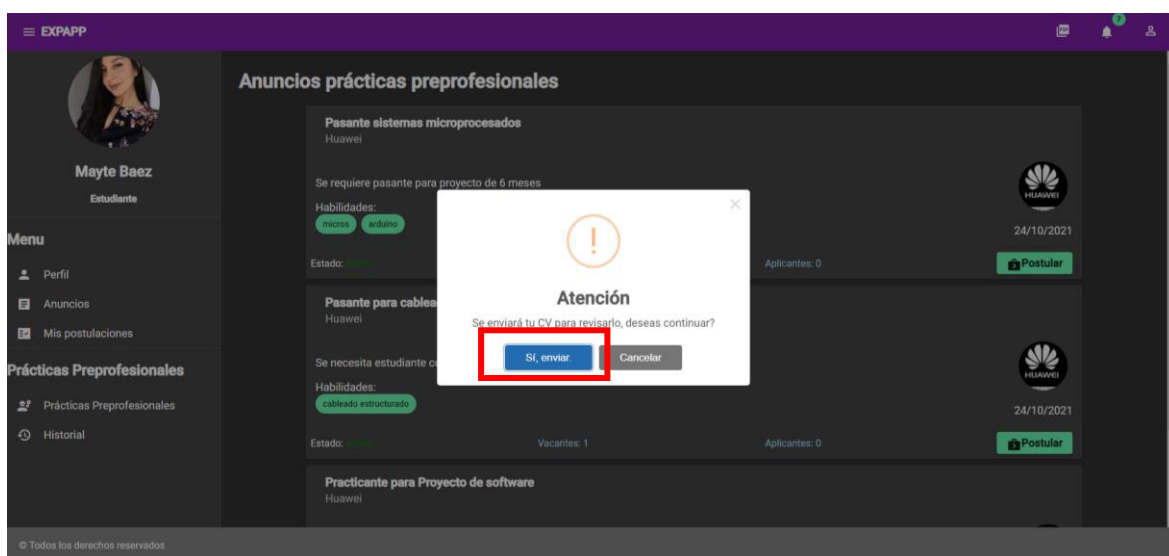


Figura 3.11. ESTUDIANTE: Pantalla de anuncios.

Se verificó que los anuncios en los que ha aplicado el estudiante se muestran en la sección “**Mis postulaciones**”, en la ruta <http://localhost:4200/estudiante/mis-postulaciones>, en donde se puede retirar la postulación dando clic en el botón “**Quitar Postulación**” que se observa en la Figura 3.12.

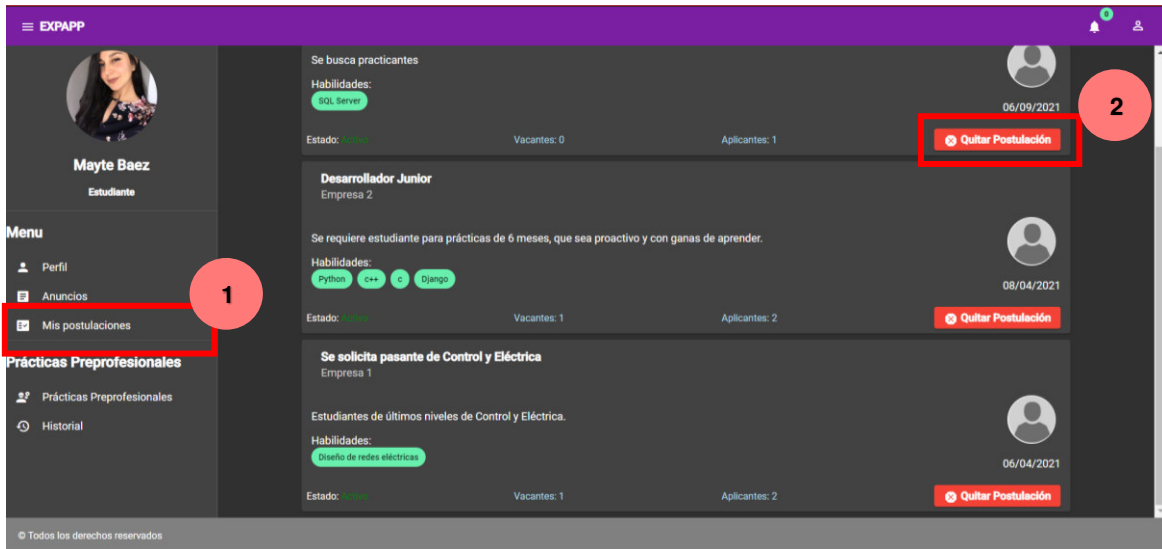


Figura 3.12. ESTUDIANTE: Anuncios en los que ha postulado el estudiante.

En la Figura 3.13 se visualiza el detalle de los anuncios para el Administrador en la ruta <http://localhost:4200/admin/detalles-anuncio/id-anuncio>, en donde puede ver los estudiantes que han aplicado al puesto y se tiene acceso al CV de cada estudiante mediante el botón “Ver Estudiante”.

Se probó el envío de un estudiante a la empresa para lo cual se preselecciona uno o varios estudiantes en la pestaña “Estudiantes aplicantes”, luego en la sección “Estudiantes Preseleccionados” se selecciona el o los estudiantes que se enviarán y finalmente se envían los datos del estudiante a la empresa dando clic en el botón “Enviar Practicantes”.

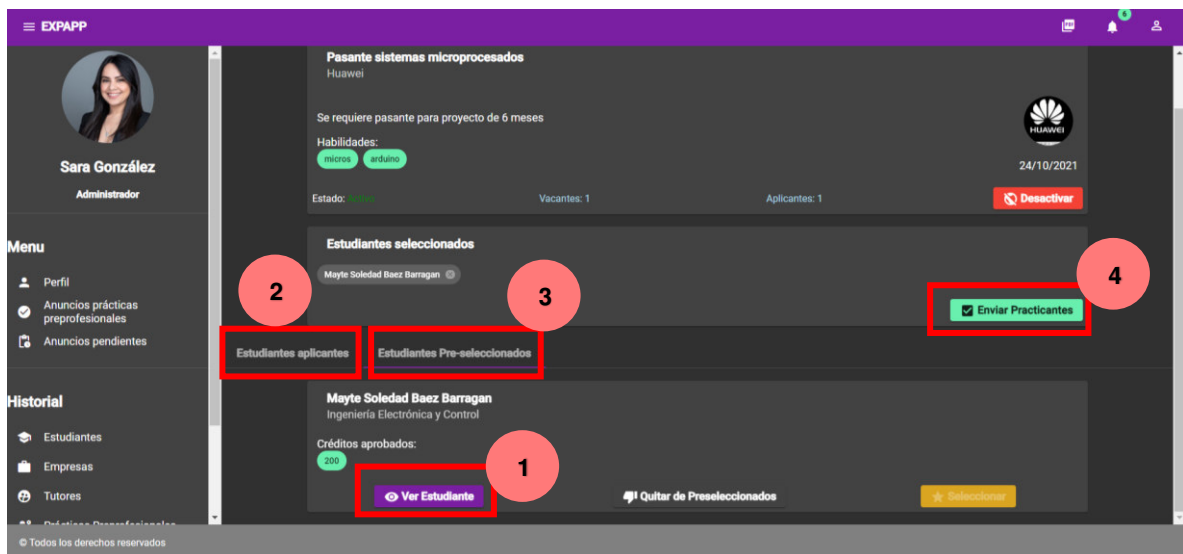


Figura 3.13. ADMINISTRADOR: Pantalla de detalles del anuncio.

Cuando se presiona “**Enviar Practicantes**” se despliega en pantalla un mensaje de confirmación del envío como se muestra en la Figura 3.14.

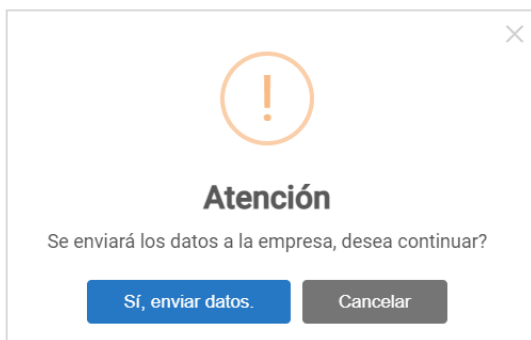


Figura 3.14. ADMINISTRADOR: Confirmación de envío de estudiante a empresa.

Al confirmar el envío aparece otro recuadro que solicita el ingreso del periodo académico en el que se realizará la práctica preprofesional como se muestra en la Figura 3.15.



Figura 3.15. ADMINISTRADOR: Ingreso de periodo académico para la práctica preprofesional.

Finalmente, se presenta un mensaje indicando que los datos del estudiante se han enviado con éxito como se observa en la Figura 3.16 y queda pendiente la confirmación del Supervisor Empresa.

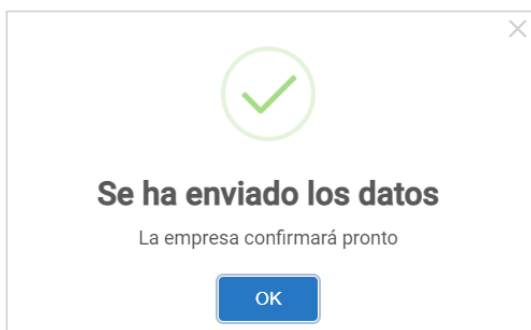


Figura 3.16. ADMINISTRADOR: Mensaje de envío de datos de estudiante exitoso.

3.1.3 GESTIÓN DE PRÁCTICAS PREPROFESIONALES

Para iniciar una práctica preprofesional el usuario Supervisor Empresa debe aceptar el estudiante enviado por el Administrador. En la pantalla de detalles del anuncio en la ruta <http://localhost:4200/empresa/detalles-anuncio/id-anuncio> que se observa en la Figura 3.17. Con el usuario Supervisor Empresa se probó la recepción de datos del estudiante y su confirmación dando clic sobre “**Aceptar Estudiante**”. Aparece el mensaje de confirmación que se visualiza en la Figura 3.18.

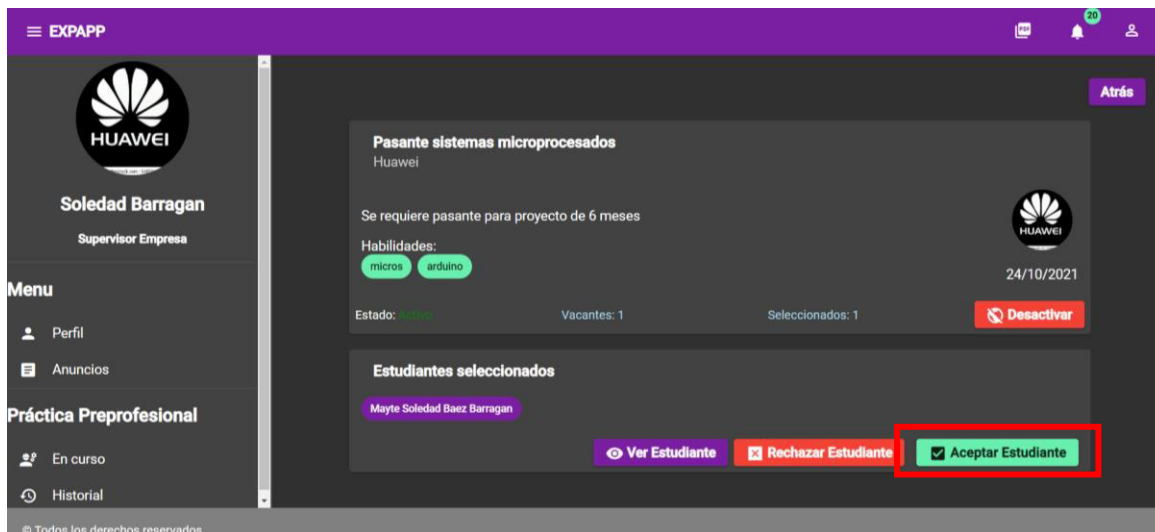


Figura 3.17. SUPERVISOR EMPRESA: Detalles del anuncio de prueba y el estudiante seleccionado.

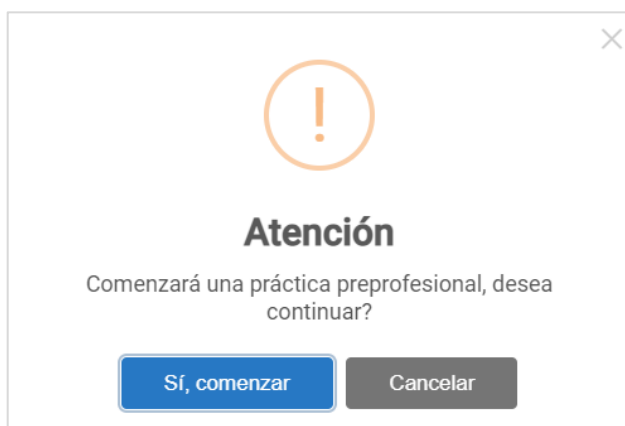


Figura 3.18. SUPERVISOR EMPRESA: Mensaje de confirmación para el inicio de práctica preprofesional.

Al confirmar el inicio de la práctica preprofesional se presenta la pantalla que se visualiza en la Figura 3.19, en donde se ingresó el Supervisor Empresa que estará a cargo de la

práctica preprofesional, y las fechas de inicio y fin de esta, finalmente, se presionó el botón “Iniciar Práctica”.

Una vez iniciada la práctica preprofesional aparece el detalle en la sección “En curso” que se encuentra en la ruta <http://localhost:4200/empresa/practicas-activas> para el Supervisor Empresa y para el Administrador en la sección “Prácticas Preprofesionales”, la cual se observa en la Figura 3.20, en la ruta <http://localhost:4200/admin/practicas-preprofesionales>.

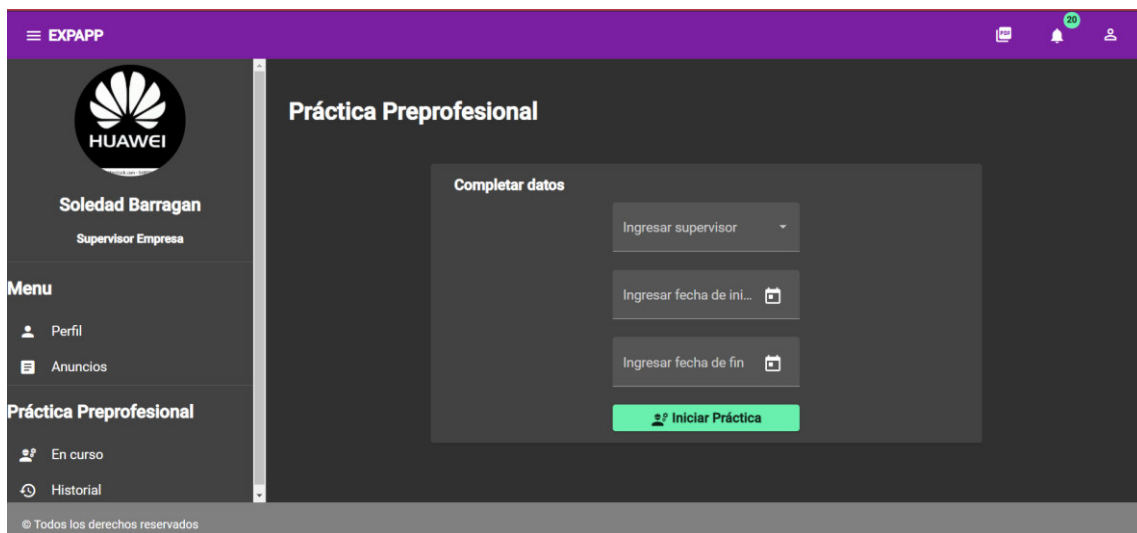


Figura 3.19. SUPERVISOR EMPRESA: Pantalla de datos de práctica preprofesional.

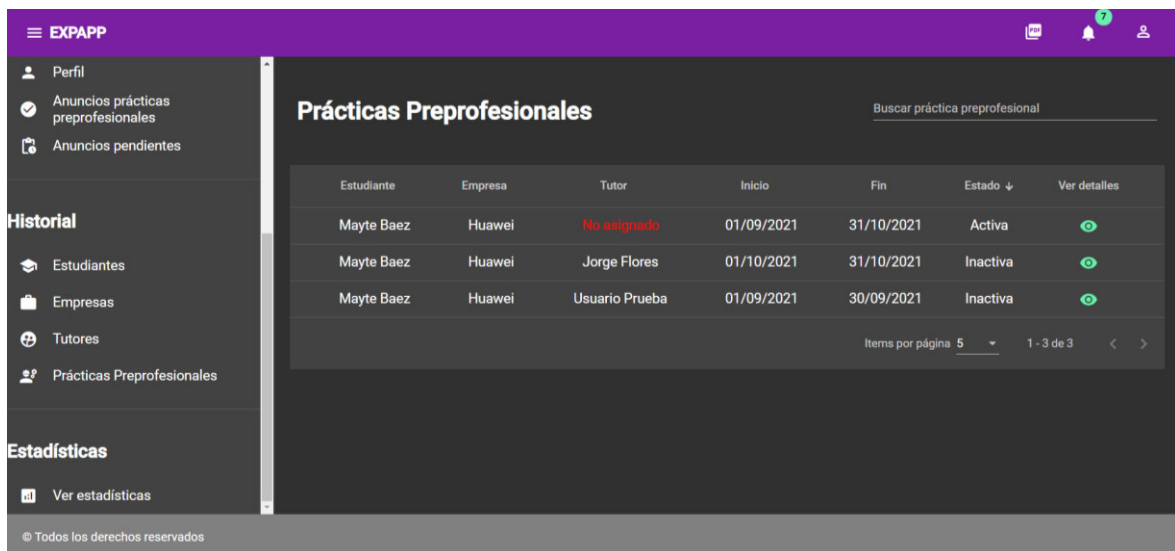


Figura 3.20. ADMINISTRADOR: Pantalla de resumen de prácticas preprofesionales.

Con el rol Administrador se probó la asignación de un Tutor en la práctica preprofesional, esto se realizó ingresando a los detalles de la práctica preprofesional de prueba presionando el botón encerrado en el recuadro de la Figura 3.20 en la sección “**Prácticas Preprofesionales**”.

Para la asignación de Tutor, desde la pantalla de detalles de la práctica preprofesional, en la ruta <http://localhost:4200/admin/practica-detalles/id-practica-preprofesional>, se seleccionó el usuario con rol Tutor creado previamente en menú de selección que se observa en la Figura 3.21, y finalmente, se presionó el botón “**Guardar**”. El Tutor recibe un correo electrónico notificando que ha sido seleccionado, el mensaje del correo se visualiza en la Figura 3.22.

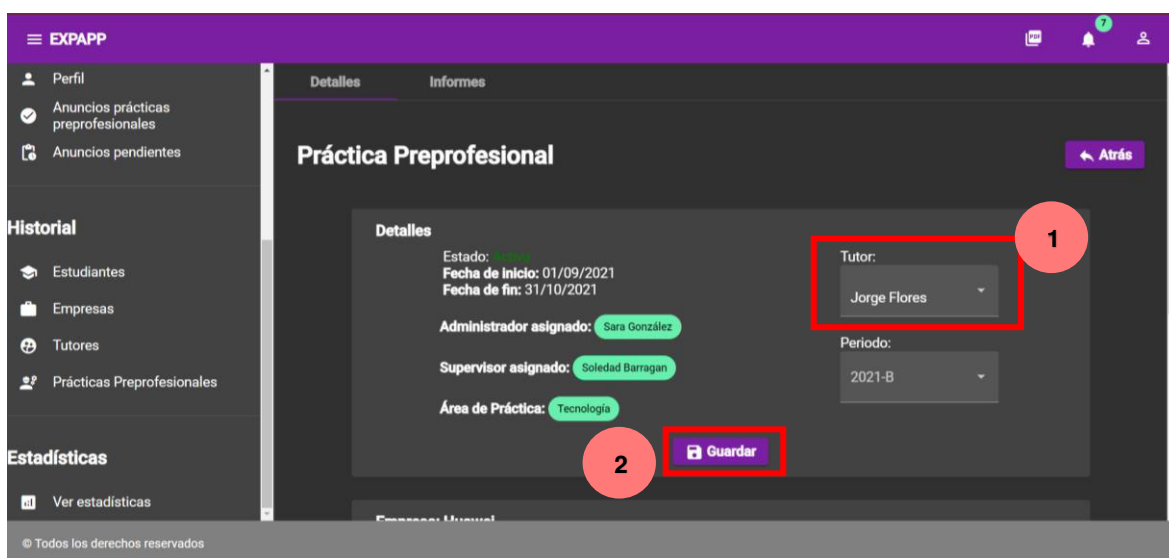


Figura 3.21. ADMINISTRADOR: Asignación de Tutor para la práctica preprofesional de prueba.

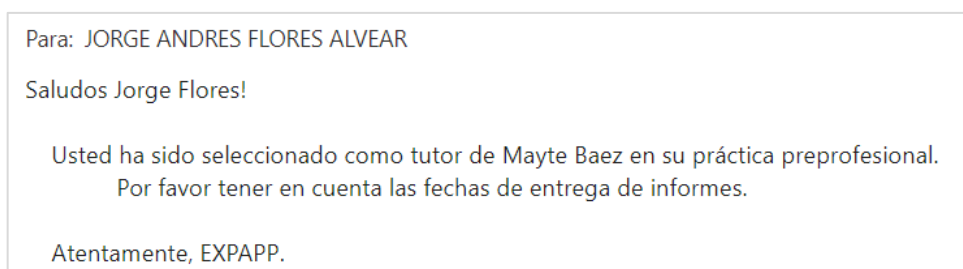


Figura 3.22. TUTOR: Correo recibido al ser asignado el Tutor.

Para el usuario Tutor se probó el ingreso de datos y generación de los informes correspondientes a la práctica preprofesional asignada. En la Figura 3.23 se observa la

pantalla en donde se ingresó datos de prueba para el Informe y en la Figura 3.24 se tiene el informe generado tras exportar el formulario como PDF.

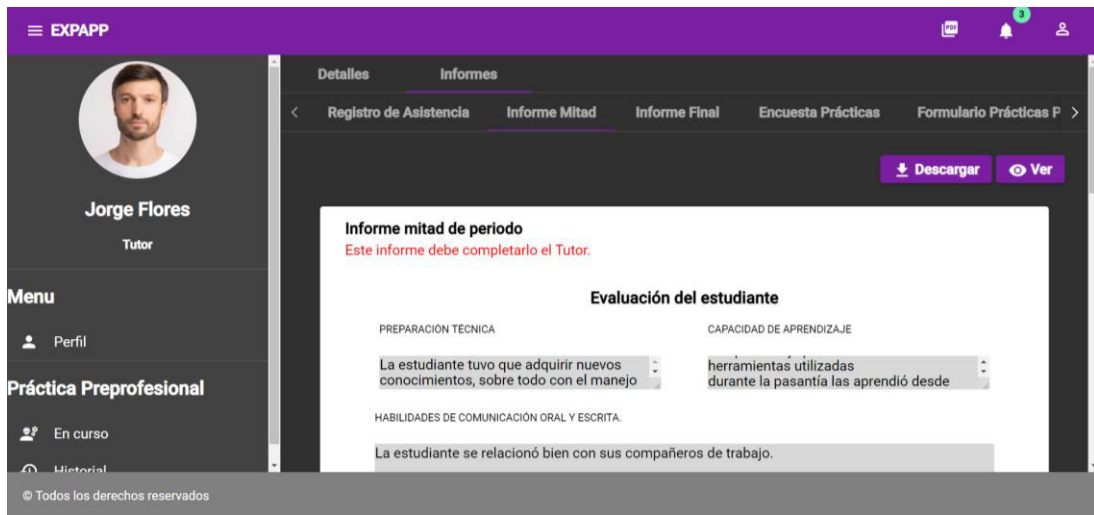


Figura 3.23. TUTOR: Formulario para completar el Informe.

ESCUELA POLITÉCNICA NACIONAL Facultad de Ingeniería Eléctrica y Electrónica INFORME DE MITAD DE PERIODO DE PRÁCTICAS PREPROFESIONALES		
1. DATOS DE LA EMPRESA O INSTITUCIÓN		
RAZÓN SOCIAL: Huawei		
DIRECCIÓN: Tumbaco n50-78	CIUDAD: Quito	
TELÉFONO: 022056564	FAX:	CORREO ELECTRÓNICO: maysol.baez@gmail.com
2. DATOS DEL PRACTICANTE		
NOMBRE: Mayte Soledad Baez Barragan		
CARRERA: Ingeniería Electrónica y Control		
PERIODO DE PRÁCTICAS: 2021-B		
3. FECHAS DE VISITA O COMUNICACIÓN CON EL ESTUDIANTE Y CON EL JEFE INMEDIATO (El seguimiento por parte del Tutor puede ser mediante visita in situ, correo electrónico o comunicación telefónica)		
FECHA	MEDIO	TEMA/OBSERVACIONES
14/9/2021	Email	Actividades que realizó la estudiante durante las prácticas, fechas y horario.
4. EVALUACIÓN DEL ESTUDIANTE		
PREPARACIÓN TÉCNICA		
La estudiante tuvo que adquirir nuevos conocimientos, sobre todo con el manejo del Framework Angular.		
CAPACIDAD DE APRENDIZAJE		
La estudiante presenta buena capacidad de aprendizaje pues todas las herramientas utilizadas durante la pasantía las aprendió desde cero en un periodo corto de tiempo.		
HABILIDADES DE COMUNICACIÓN ORAL Y ESCRITA. EN EL CASO DE ESTUDIANTES CON DISCAPACIDAD, DEBERÁ INDICARSE EL GRADO DE AUTONOMÍA PARA ESTA HABILIDAD Y SI REQUIERE DE ALGÚN TIPO DE RECURSO TÉCNICO Y/O HUMANO PARA LA MISMA:		
La estudiante se relacionó bien con sus compañeros de trabajo.		
CAPACIDAD DE TRABAJO EN EQUIPO:	SI	
CREATIVIDAD E INICIATIVA:	SI	
FACILIDAD DE ADAPTACIÓN:	SI	
SENTIDO DE RESPONSABILIDAD:	SI	
PUNTUALIDAD:	SI	
5. CERTIFICACIONES		
PROFESOR TUTOR	JEFE DE DEPARTAMENTO	
FECHA DE ELABORACIÓN: ___/___/___	FECHA DE RECEPCIÓN: ___/___/___	
	FECHA DE APROBACIÓN: ___/___/___	
	APROBADO POR:	
f. _____	f. _____	
Profesor Tutor	(SELLO OFICIAL)	
Nombre:	Nombre:	
Cargo:	Cargo:	

Figura 3.24. TUTOR: Informe generado (Inf3) tras presionar “Descargar” o “Ver” de la Figura 3.23.

Finalmente, para el Administrador se probó la sección Estadísticas que se encuentran en la ruta: <http://localhost:4200/admin/estadisticas> que se muestra en la Figura 3.25. Se verificó en la base de datos que los datos mostrados en las gráficas sean los correctos.



Figura 3.25. ADMINISTRADOR: Estadísticas.

3.2 PRUEBAS DE USUARIO

Las pruebas se realizaron desplegando la página web en Firebase Hosting para que los usuarios que probaran su funcionamiento puedan acceder desde cualquier navegador con acceso a Internet. Se seleccionaron dos usuarios para cada rol que tiene la aplicación: Administrador técnico, Administrador, Estudiante, Supervisor Empresa y Tutor.

Para Administradores y Tutores se seleccionaron docentes de la FIEE; en el caso del rol Supervisor Empresa se seleccionaron personas externas a la universidad que han trabajado con practicantes en su ámbito laboral. Para el rol Administrador técnico se seleccionó un trabajador de la FIEE y un estudiante; y, finalmente, dos egresados de la facultad para el rol Estudiante.

A cada usuario se le entregó el Manual de Usuario de EXPAPP (ANEXO D) para que se familiarice con el uso de la aplicación web y un listado de las actividades que le correspondía realizar. Se dividió en dos grupos a los usuarios, en cada grupo había un usuario de cada rol, y a cada uno le correspondía realizar ciertas actividades necesarias en el proceso de práctica preprofesional.

A continuación, se describen brevemente las actividades mínimas que cada usuario realizó durante el proceso de pruebas.

Tabla 3.1. Descripción del proceso de pruebas de usuario de EXPAPP.

Rol	Actividades por realizar
Administrador técnico	<ul style="list-style-type: none"> • Iniciar sesión con las credenciales enviadas a su correo electrónico. • Modificar datos personales de su perfil. • Actualizar su contraseña. • Crear una cuenta de tipo Administrador. • Crear una cuenta de tipo Tutor. • Crear una cuenta de tipo Empresa. • Crear dos carreras y materias que corresponden a dichas carreras. • Subir información de estudiantes importando un documento de tipo CSV.
Administrador	<ul style="list-style-type: none"> • Iniciar sesión con las credenciales enviadas a su correo electrónico. • Modificar datos personales de su perfil. • Actualizar su contraseña. • Revisar anuncio creado por el usuario de tipo Supervisor Empresa y rechazar (poner la razón del rechazo). • Navegar por todas las pestañas de la sección Historial. • Navegar por la sección Estadísticas. • Crear un Anuncio para la empresa asignada, con una vacante para las carreras Control y Eléctrica. • Preseleccionar estudiantes una vez que hayan aplicado al puesto. • Seleccionar un estudiante para el proceso de práctica preprofesional. • Enviar otro estudiante de la lista de preseleccionados en el caso de que la empresa rechace el estudiante seleccionado. • Asignar un tutor y período académico cuando el estudiante seleccionado sea aceptado por la empresa. • Revisar los informes de prácticas preprofesionales.
Empresa	<ul style="list-style-type: none"> • Iniciar sesión con las credenciales enviadas a su correo electrónico. • Modificar datos personales de su perfil. • Actualizar su contraseña. • Actualizar la dirección de la empresa. • Agregar un supervisor con datos ficticios. • Crear un anuncio solicitando estudiante de Ingeniería Civil (El propósito es que el anuncio sea rechazado). • Recibir notificación de creación de anuncio por parte del usuario Administrador. • Recibir notificación asignación de estudiante para la práctica preprofesional. • Rechazar el primer estudiante enviado y seleccionar la opción de solicitar otro. • Aceptar el nuevo estudiante enviado por el usuario Administrador Técnico.

	<ul style="list-style-type: none"> • Colocar la fecha de inicio de práctica preprofesional y la fecha de fin. Seleccionarse como supervisor. • Recibir recordatorios para llenar los datos en informes de la práctica preprofesional. • Exportar a PDF los informes de prácticas preprofesionales. • Finalizar la práctica preprofesional.
Estudiante	<ul style="list-style-type: none"> • Crear cuenta en la página web. • Actualizar su contraseña. • Llenar los datos requeridos para crear el CV y exportarlo como PDF. • Postular a las prácticas preprofesionales de los anuncios que se encuentren activos. • Revisar que la postulación se encuentre en la sección Mis Postulaciones. • Recibir notificación indicando que ha sido seleccionado para el puesto. • Recibir recordatorios al correo electrónico para llenar los informes de la práctica preprofesional. • Exportar a PDF los informes de prácticas preprofesionales.
Tutor	<ul style="list-style-type: none"> • Iniciar sesión con las credenciales enviadas a su correo electrónico. • Modificar datos personales de su perfil. • Actualizar su contraseña. • Tratar de ingresar al siguiente enlace: https://prototipoexp.web.app/admin (esto con el fin de probar la seguridad en las rutas de otros usuarios). • Recibir alerta al correo electrónico cuando haya sido seleccionado como tutor en una práctica preprofesional. • Llenar los informes cuando reciba los recordatorios por correo electrónico; suponer que se comunicó con el supervisor de la empresa. • Exportar a PDF los informes de prácticas preprofesionales.

3.2.1 RESULTADOS

Para analizar el resultado de las pruebas y la satisfacción de los usuarios se crearon dos cuestionarios, uno para los usuarios involucrados en el proceso de prácticas preprofesionales, y otro para el usuario Administrador Técnico debido a que este rol no participa directamente en el proceso de prácticas preprofesionales.

En la Tabla 3.2 tienen las respuestas obtenidas para el rol Administrador Técnico, en la cual se puede observar que todas las respuestas son positivas y no se reportaron errores en EXPAPP.

Tabla 3.2. Resultado del cuestionario aplicado al rol Administrador Técnico.

#	Pregunta	Respuesta		
		SÍ	NO	OTRAS
1	¿Le pareció el Manual de Usuario útil para entender el funcionamiento de la página?	100%	0%	No Aplica
2	¿En qué podría mejorar el Manual de Usuario?	No Aplica	No Aplica	Los usuarios opinaron que es de fácil utilización, y que se podría incluir más imágenes.
3	¿Pudo realizar la creación de usuarios fácilmente?	100%	0%	No Aplica
4	¿Encontró algún error realizando las actividades requeridas?	100%	0%	No Aplica
5	Si la pregunta anterior fue afirmativa, ¿Qué problemas encontró?	No Aplica	No Aplica	No Aplica
6	Comentarios y Sugerencias	No Aplica	No Aplica	Los usuarios opinaron que el aplicativo es muy fácil y accesible de usar

En la Tabla 3.3 se tienen las respuestas de los roles de usuario que participan en el proceso de prácticas preprofesionales: Administrador, Supervisor Empresa, Estudiante y Tutor.

Tabla 3.3. Resultado del cuestionario aplicado a los roles participantes en el proceso de prácticas preprofesionales.

#	Pregunta	Respuesta		
		SÍ	NO	OTRAS
1	¿Le pareció el Manual de Usuario útil para entender el funcionamiento de la página?	100%	0%	No Aplica
2	¿En qué podría mejorar el Manual de Usuario?	No Aplica	No Aplica	En general los usuarios opinaron que el manual de usuario es claro y que debería presentarse en alguna sección de EXPAPP.
3	¿Recibió notificaciones y correos del proceso de prácticas preprofesionales?	85.7%	14.3%	No Aplica
4	¿Cree que las notificaciones fueron claras?	71.4%	0%	El 14.3% de los usuarios no recibieron notificaciones y el otro 14.3% de usuarios opina que hay notificaciones que pueden omitirse al usuario Administrador porque generan confusión.
5	Al generar los archivos relacionados con la práctica preprofesional ¿Se generaron los PDF exitosamente?	85.7%	0%	El 14.3% de usuarios solo pudo ver los PDF generados al ver el CV del estudiante, pero no de los informes de la práctica preprofesional.
6	¿Pudo completar las actividades solicitadas de manera correcta y sin inconvenientes?	85.7%	14.3%	No Aplica
7	¿Encontró algún error realizando las actividades requeridas?	57.1%	42.9%	No Aplica

Para el proceso de prácticas preprofesionales se generaron ciertos errores y en la Tabla 3.4 se presenta la lista de errores y sus soluciones.

Tabla 3.4. Errores reportados en las pruebas de usuario de EXPAPP.

#	Errores reportados	Descripción	Solución
1	El rol estudiante no recibió la notificación de que un anuncio de su carrera fue creado.	En el envío de la notificación se estaba enviando el email del estudiante en lugar del identificador, por esta razón, no se encontraba el estudiante en la base de datos pues se lo busca por <i>uid</i> para guardar las notificaciones.	En la función que envía el correo de notificación se cambió el parámetro student.email por student.uid
2	El campo de capacidad de aprendizaje del formulario del informe mitad no carga el dato almacenado cuando se reingresa a la aplicación. Se muestra vacío a pesar de haber sido ingresado en una ocasión anterior. Al generar el informe se muestra que si se cargó en la base de datos.	Cuando se traen los datos de los informes se asignan en los campos del formulario, y este campo "Capacidad de aprendizaje" no se estaba asignando.	Se agregó la asignación de este valor en el campo del formulario correspondiente.
3	Las fechas de los anuncios y de las prácticas preprofesionales eran inconsistentes.	En Firebase se guardan las fechas con formato <i>TimeStamp</i> , pero al momento de traer estos campos de <i>Firestore</i> para mostrarlos deben estar en formato <i>Date</i> .	Para cambiar de formato se realizó la siguiente transformación al momento de traer fechas de <i>Firestore</i> : <i>date.seconds*1000</i>
4	La hora de almuerzo en el informe de horas no se guarda.	En el informe de horas se tiene la variable <i>informe.lunchHour</i> e <i>informe.lunchHourCheck</i> . la primera indica la hora de almuerzo y la segunda es un valor de tipo <i>boolean</i> que define si se tiene o no hora de almuerzo. Estos dos valores se encontraban cambiados al momento de guardar en la base de datos y solo se guardaba <i>informe.lunchHourCheck</i>	En la función se asignó el valor correspondiente a la variable <i>informe.lunchHour</i> .

En el cuestionario también se colocó un apartado llamado "Comentarios y sugerencias"; en la Tabla 3.5 se listan las sugerencias que se aplicaron a EXPAPP.

Tabla 3.5. Cambios sugeridos por los usuarios.

#	Descripción
1	Agrandar el tamaño de los campos de los informes que requieren gran cantidad de texto.
2	Los informes que se descarguen que tengan un nombre predefinido con datos del estudiante.
3	Listar los periodos académicos desde el más reciente al más antiguo.
4	En el menú principal en lugar de mostrar la opción de "Anuncios Publicados" cambiar a "Anuncios prácticas preprofesionales".
5	En el perfil colocar el botón "Guardar" en la sección que contiene los datos que se permite actualizar.
6	Uso de mayúsculas y minúsculas en los títulos de los formularios. Solo la primera letra de la primera palabra de las frases debería ir en mayúscula.
7	Colocar datos de contacto del Supervisor de empresa en el detalle de la práctica preprofesional.
8	En la lista de prácticas preprofesionales cambiar "Ver detalles" por "Realizar seguimiento".
9	En los informes mejorar los colores del texto y avisos.
10	En el informe de mitad de periodo cambiar "Agregar Campo en Visitas" por "Registrar seguimiento", "Medio" por "Medio de comunicación", y al botón le pondría "Guardar seguimiento".
11	En el Formulario prácticas preprofesionales dividir en dos secciones claras para que se sepa lo que llena el tutor y lo que llena el supervisor.
12	En el Formulario prácticas preprofesionales al escoger las materias, debe quedar más claro que se puede escoger varias.
13	Colocar descripción en botones para que quede claro la acción que realizan.
12	Tener disponible el manual de usuario en la aplicación web.

Gracias a los comentarios y sugerencias realizados por los usuarios que participaron en el proceso de pruebas se mejoró visualmente la interfaz de usuario. Para que el aspecto de la aplicación esté relacionado a la FIEE se decidió cambiar la paleta de colores a tonos azules como se muestra en la Figura 3.26.

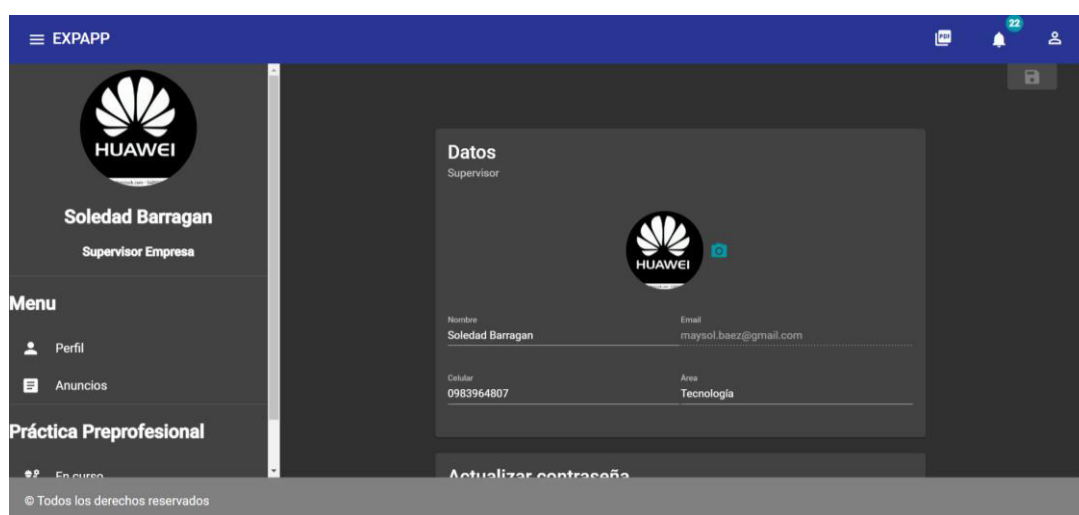


Figura 3.26. Interfaz con nueva paleta de colores.

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

Como resultado final se tiene el prototipo de una aplicación web, EXPAPP, para la obtención y seguimiento de prácticas preprofesionales en la Facultad de Ingeniería Eléctrica y Electrónica (FIEE). Para la implementación de este prototipo se adquirieron conocimientos de Angular y Typescript para el desarrollo del *frontend*, y Firebase y JavaScript para el *backend*. A continuación, se presentan las conclusiones tras la culminación de este trabajo de titulación.

- Para conocer cuáles son los pasos del proceso de prácticas preprofesionales y para evaluar cómo se ejecutan actualmente en la FIEE, se realizaron entrevistas a los involucrados. Con las entrevistas se obtuvo el flujo de trabajo definido para la asignación y aprobación de una práctica preprofesional. Con los testimonios de los entrevistados se llegó a la conclusión de que existen demoras en el proceso de prácticas preprofesionales. Las demoras se presentan principalmente cuando: el estudiante busca prácticas preprofesionales, y, una vez empezada la práctica, en la asignación del tutor y en la entrega de informes de seguimiento.
- Una vez evaluado el proceso de prácticas preprofesionales se realizó el diseño de EXPAPP. En el diseño se definió que la aplicación web requiere de cinco roles: Administrador Técnico, Administrador, Supervisor Empresa, Estudiante y Tutor.
- El diagrama de contexto facilitó el diseño de EXPAPP, pues en este diagrama se define los límites entre el sistema y su entorno, mostrando las entidades externas involucradas y las acciones que realizará cada entidad (rol de usuario) en el prototipo; a partir de este se elaboraron los diagramas de casos de uso y los requerimientos funcionales.
- El rol Administrador Técnico no tiene participación directa en el proceso de prácticas preprofesionales; sin embargo, es fundamental su existencia en EXPAPP. Este usuario realiza el CRUD de usuarios, carreras, materias y periodos académicos. Todos estos datos son necesarios para una práctica preprofesional. Este rol puede ser algún miembro del personal administrativo de la FIEE o un estudiante, debido a que se maneja información relacionada con la facultad y se debe tener conocimiento previo sobre el proceso de prácticas preprofesionales y los involucrados.

- En este trabajo se desarrolló el prototipo de aplicación web considerando los roles identificados y cada ruta de los diferentes roles se protege utilizando *guards* de Angular. Cada usuario tiene acceso a pantallas exclusivas de acuerdo con su rol, debido a que todas las rutas de navegación están protegidas por el *RoleGuard*. También existen pantallas en común, como la visualización de informes de práctica preprofesionales, y pantallas públicas como la pantalla con mensaje de error 404 o el inicio de sesión.
- Gracias a EXPAPP la FIEE puede brindar a los estudiantes la posibilidad de encontrar prácticas preprofesionales y realizar su seguimiento en menor tiempo del que toma actualmente. En EXPAPP se realiza desde la creación de anuncios por parte de las empresas, la postulación y selección de estudiantes para la práctica preprofesional, y finalmente, existe el envío de recordatorios para completar el seguimiento de la práctica mediante la edición de formularios en la aplicación web. Facilitar el proceso y reducir el tiempo que toma es un gran beneficio para los estudiantes pues cumplir sus horas de prácticas preprofesionales es requisito para la titulación.
- EXPAPP también brinda al personal administrativo la posibilidad de involucrarse más en el proceso al tener un control más directo de las prácticas preprofesionales. El usuario Administrador, que corresponde al personal administrativo y miembros de la Comisión de Prácticas Preprofesionales de la FIEE, tiene accesible la información de las prácticas preprofesionales que realizan los estudiantes. Además, tiene la posibilidad de observar estadísticas relacionadas a estudiantes, tutores y empresas, con el fin de tener datos de cómo se ejecutan las prácticas y con estos ver la manera de mejorar el proceso.
- Antes de probar el prototipo EXPAPP con los usuarios se realizaron pruebas de funcionamiento, para lo cual se siguió el proceso de inicio a fin, desde la creación de usuarios hasta la finalización de la práctica preprofesional. En esta fase fue importante la participación de todos los roles de manera secuencial basándose en el flujo de prácticas preprofesionales. Esto debido a que ciertas acciones que realiza un rol de usuario dependen de otro rol, por ejemplo, un Estudiante no puede aplicar a un anuncio si es que el Supervisor Empresa no ha creado uno.
- En la fase de pruebas los usuarios aprendieron el uso de EXPAPP de manera rápida, pues se desarrollaron apropiadamente y realizaron correctamente las actividades

solicitadas. La fase de pruebas culminó de manera exitosa, a pesar de que existieron errores, no fueron bloqueantes en el proceso y se comprobó que la aplicación web cumple con los requerimientos necesarios para completar el proceso de prácticas preprofesionales de inicio a fin.

- La Comisión de Prácticas Preprofesionales de la FIEE se involucró en la fase de pruebas realizando una demostración del prototipo EXPAPP. Los miembros de la comisión opinaron que esta aplicación es necesaria en la FIEE para agilizar el proceso de prácticas preprofesionales.

En conclusión, la implementación del prototipo EXPAPP ayudaría a tener el control del proceso de prácticas preprofesionales de manera centralizada. Se les brinda a los estudiantes de la FIEE la oportunidad de encontrar prácticas preprofesionales y junto con personal administrativo, docentes y empresas llevar su seguimiento de manera rápida y ordenada.

4.2 RECOMENDACIONES

EXPAPP es un prototipo de aplicación que permite reducción de tiempos en el proceso de prácticas preprofesionales, que en el caso de ser implementada en la FIEE se recomienda al personal administrativo crear convenios con empresas e incorporarlas a la aplicación web, para brindar a los estudiantes la posibilidad de encontrar alguna plaza para realizar sus prácticas de manera rápida.

EXPAPP cumple exitosamente con el proceso de prácticas preprofesionales de inicio a fin de manera exitosa, sin embargo, para este prototipo existen posibles implementaciones futuras, las cuales se presentan a continuación:

- Actualmente, al finalizar la práctica preprofesional se requiere la impresión de los informes para que sea firmado por la empresa, tutor, estudiante y personal administrativo. Para implementaciones futuras se recomienda integrar un sistema de firma electrónica a EXPAPP. De esta manera los informes pueden ser firmados directamente desde la aplicación web, con lo que se evita la impresión de documentos y se disminuiría considerablemente el tiempo de entrega de informes.
- En el proceso de prácticas preprofesionales la comunicación del tutor y el supervisor de la empresa se realiza por medio de correo electrónico o llamada telefónica. Se puede recomendar la integración un sistema de videoconferencia en EXPAPP, para

que la comunicación que se da en el seguimiento de prácticas preprofesionales se realice directamente desde la aplicación web.

- Con el uso de inteligencia artificial (IA) se puede incorporar motores de recomendación al sistema EXPAPP. Esto ayudaría a partes del proceso como: la asignación de Tutor, selección de estudiantes para una práctica, envío de recordatorios inteligentes. Con el uso de IA se puede reducir aún más el tiempo que transcurre desde que una empresa crea el anuncio, hasta que se termina la práctica preprofesional.
- Se recomienda implementar la generación periódica de copias de seguridad en otra base de datos, debido a que hoy en día los datos se guardan únicamente en Firebase y si es que se genera alguna eliminación indeseada, los datos de prácticas preprofesionales: informes, estudiantes, tutores y empresas, entre otros, se perderían, sin tener la posibilidad de ser recuperados.
- Se recomienda la implementación de un sistema de validación de datos como: RUC y números de cédula para empresas, supervisores de empresa y estudiante. La DGIP de la EPN posee un servicio web para la validación de estos datos que puede ser integrado a EXPAPP.

Para mejorar el desarrollo de EXPAPP o de aplicaciones que utilicen la misma tecnología se representan las siguientes recomendaciones técnicas:

- Cuando se usen librerías se recomienda tener cuidado de que no afecten al funcionamiento de la aplicación web. Ciertas librerías pueden tener elementos con el mismo nombre de elementos de la aplicación o elementos *HTML* y puede alterar su funcionamiento. Tal es el caso de *SweetAlert*, que, como parte de la configuración de las alertas tiene un elemento llamado *footer*, y al desplegar la alerta se movía el elemento de la aplicación llamado igual. Como solución a esto se configuró cada alerta con el valor en *heighAuto* en *false*.
- Para la fase de diseño se recomienda la elaboración del diagrama de contexto pues ayuda a entender los límites del sistema, las entradas y salidas; y cómo se relaciona con otros sistemas. Además, en base a este diagrama se puede obtener los diagramas de casos de uso y requerimientos funcionales con mayor facilidad.
- Se recomienda mantener el estándar de programación en idioma inglés, debido a que se encuentra dentro de las buenas prácticas de programación.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Inc, «What is Front-End Development: Key Technologies and Concepts», *Medium*, may 19, 2020. <https://altexsoft.medium.com/what-is-front-end-development-key-technologies-and-concepts-5c08dd46027b> (accedido ene. 04, 2021).
- [2] «2.11 DOM avanzado». https://books.adalab.es/materiales-front-end-e/sprint-2.-programando-la-web/2_11_dom_avanzado (accedido ene. 06, 2021).
- [3] «JavaScript HTML DOM». https://www.w3schools.com/js/js_htmlDOM.asp (accedido ene. 04, 2021).
- [4] J. Gutiérrez, «¿Qué es un framework web?», p. 4.
- [5] J. R. M. Ríos, N. M. L. Mora, M. P. Z. Ordóñez, y E. L. L. Sojos, «Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python», *Rev. Latinoam. Ing. Softw.*, vol. 4, n.º 4, Art. n.º 4, sep. 2016, doi: 10.18294/relais.2016.201-207.
- [6] O. Cea, «RxJS: de cero a experto en 15 minutos», *Medium*, sep. 24, 2020. <https://osmancea.medium.com/programaci%C3%B3n-reactiva-con-rxjs-bebc9432485f> (accedido ago. 11, 2021).
- [7] «Introducción teórica a los observables en Angular». <https://desarrolloweb.com/articulos/introduccion-teorica-observables-angular.html> (accedido ago. 11, 2021).
- [8] M. Rodriguez, «¡Conozcamos los Observables!», *Medium*, ago. 22, 2019. <https://medium.com/@mayrarodriguez/conozcamos-los-observables-15ee9e7c5aa9> (accedido ago. 11, 2021).
- [9] L. Puciarelli, *Angular*. Six Ediciones, 2020.
- [10] «Angular - CLI Overview and Command Reference». <https://angular.io/cli> (accedido nov. 23, 2020).
- [11] *angular/angular-cli*. Angular, 2020. Accedido: nov. 23, 2020. [En línea]. Disponible en: <https://github.com/angular/angular-cli>
- [12] T. Molina, «Angular: Arquitectura del Framework», *Medium*, sep. 26, 2018. <https://medium.com/angular-chile/angular-arquitectura-del-framework-e46204f38fef> (accedido ene. 07, 2021).
- [13] C. Tuzi, «Understanding Angular modules (NgModule) and their scopes», *Medium*, feb. 19, 2020. <https://medium.com/@cyrilletuzi/understanding-angular-modules-ngmodule-and-their-scopes-81e4ed6f7407> (accedido ago. 25, 2021).

- [14] C. Flores, «Providers en Angular», *Medium*, jun. 01, 2020. <https://medium.com/@cristianflores.ee/providers-en-angular-1832e8093e2a> (accedido ene. 11, 2021).
- [15] «Angular - Introducción a Angular Docs». <https://angular.io/docs> (accedido ago. 25, 2021).
- [16] «The Good and the Bad of Angular Development», *AltexSoft*. https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/?utm_source=MediumCom&utm_medium=referral&utm_campaign=shared/ (accedido ene. 04, 2021).
- [17] «El concepto de los componentes en Angular y su arquitectura». <https://desarrolloweb.com/articulos/concepto-teorico-componente-angular2.html> (accedido ene. 07, 2021).
- [18] «¿Qué patron usa Angular? MVC o MVVM», *OpenWebinars.net*, oct. 16, 2018. <https://openwebinars.net/blog/que-patron-usa-angular-mvc-o-mvvm/> (accedido ene. 12, 2021).
- [19] Okta, «MVC in an Angular World», *Scotch*. <http://scotch.io/tutorials/mvc-in-an-angular-world> (accedido ene. 12, 2021).
- [20] C. Caballero, «Understanding MVC Services for Front End: Angular», *Medium*, nov. 13, 2019. <https://medium.com/better-programming/https-medium-com-ccaballero-understanding-mvc-services-for-front-end-angular-a6196492ee74> (accedido ene. 12, 2021).
- [21] R. Chenkie, «Angular Authentication: Using Route Guards», *Medium*, jul. 09, 2019. https://medium.com/@ryanchenkie_40935/angular-authentication-using-route-guards-bf7a4ca13ae3 (accedido jul. 06, 2021).
- [22] «Angular - CanActivate». <https://angular.io/api/router/CanActivate> (accedido jul. 20, 2021).
- [23] «Angular: agregue navegación con enrutamiento». <https://angular.io/tutorial/toh-pt5> (accedido ago. 25, 2021).
- [24] «Footer Only (for guides iframe)». <https://www.pluralsight.com/utilities/footer-only> (accedido ago. 25, 2021).
- [25] «Material Design», *Material Design*. <https://material.io/components> (accedido jul. 20, 2021).
- [26] D. L. García, «Tutorial Angular Material - Aprende a usarlo rápido», *Coding Potions*. <https://codingpotions.com/angular-material> (accedido jul. 20, 2021).

- [27] Nestrategia, «¿Qué es el Back End y Front End? ⚡ 20% Dto en servicios de Desarrollo», *Agencia Inbound Marketing Madrid*, feb. 16, 2017. <https://nestrategia.com/desarrollo-web-back-end-front-end/> (accedido may 27, 2021).
- [28] «Bases de datos no relacionales | Bases de datos de gráficos | AWS», *Amazon Web Services, Inc.* <https://aws.amazon.com/es/nosql/> (accedido may 30, 2021).
- [29] «Qué es Firebase: funcionalidades, ventajas y conclusiones», *DIGITAL55*, may 17, 2020. <https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/> (accedido jun. 07, 2021).
- [30] «Firebase Authentication | Simple, free multi-platform sign-in», *Firebase*. <https://firebase.google.com/products/auth?hl=es> (accedido ago. 25, 2021).
- [31] «Cloud Firestore», *Firebase*. <https://firebase.google.com/docs/firestore?hl=es> (accedido jun. 21, 2021).
- [32] «Cloud Storage for Firebase | Store and serve content with ease», *Firebase*. <https://firebase.google.com/products/storage?hl=es> (accedido ago. 25, 2021).
- [33] «Cloud Functions for Firebase | Run your mobile backend code without managing servers», *Firebase*. <https://firebase.google.com/products/functions?hl=es> (accedido ago. 25, 2021).
- [34] «Firebase Hosting | Fast and secure web hosting», *Firebase*. <https://firebase.google.com/products/hosting?hl=es> (accedido ago. 25, 2021).
- [35] *angular/angularfire*. Angular, 2021. Accedido: jul. 20, 2021. [En línea]. Disponible en: <https://github.com/angular/angularfire>
- [36] «AngularFire», *Firebase Open Source*. <https://firebaseopensource.com/projects/angular/angularfire2/> (accedido ago. 16, 2021).
- [37] «PROCEDIMIENTO_PARA_SEGUIMIENTO_DE_PRACTICAS_PRE_PROFESIONALES.pdf». Accedido: sep. 01, 2021. [En línea]. Disponible en: https://fieee.epn.edu.ec/images/fieee/estudiantes/pregrado/PROCEDIMIENTO_PARA_SEGUIMIENTO_DE_PRACTICAS_PRE_PROFESIONALES.pdf
- [38] «PROCEDIMIENTO_PARA_REALIZACION_DE_PRACTICAS_PRE_PROFESIONALES.pdf». Accedido: sep. 01, 2021. [En línea]. Disponible en: https://fieee.epn.edu.ec/images/fieee/estudiantes/pregrado/PROCEDIMIENTO_PARA_REALIZACION_DE_PRACTICAS_PRE_PROFESIONALES.pdf

- [39] C. Jiménez de Parga, *UML Aplicaciones en Java y C++*. Madrid, España: RA-MA, 2015.
- [40] R. L. Granados La Paz, *Desarrollo de aplicaciones web en el entorno servidor*. IC Editorial, 2015.
- [41] «Requerimientos Funcionales y No Funcionales, ejemplos y tips | by Requeridos Blog | Medium». <https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a> (accedido nov. 04, 2021).
- [42] «Documentación», *Firebase*. <https://firebase.google.com/docs?hl=es> (accedido sep. 07, 2021).
- [43] «Primeros pasos: Escribe, prueba e implementa tus primeras funciones», *Firebase*. <https://firebase.google.com/docs/functions/get-started?hl=es> (accedido ago. 29, 2021).
- [44] «Llama a funciones desde tu app», *Firebase*. <https://firebase.google.com/docs/functions/callable?hl=es> (accedido ago. 29, 2021).
- [45] «Nodemailer :: Nodemailer». <https://nodemailer.com/about/> (accedido ago. 29, 2021).
- [46] «Uso :: Nodemailer». <https://nodemailer.com/usage/> (accedido ago. 29, 2021).
- [47] «Dynamic Scheduled Background Jobs in Firebase». <https://fireship.io/lessons/cloud-functions-scheduled-time-trigger/> (accedido ago. 30, 2021).
- [48] «Usar promesas - JavaScript | MDN». https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Using_promises (accedido ago. 30, 2021).
- [49] «Angular Flex-Layout Demos». <https://tburleson-layouts-demos.firebaseio.com/#/docs> (accedido ago. 17, 2021).
- [50] «Demostración de Angular 2 Charts». <https://valor-software.com/ng2-charts/#/GeneralInfo> (accedido ago. 17, 2021).
- [51] «SweetAlert2». <https://sweetalert2.github.io/> (accedido ago. 23, 2021).

ANEXOS

ANEXO A. CREACIÓN DEL PROYECTO EN FIREBASE

ANEXO B. CÓDIGO DEL BACKEND

ANEXO C. CÓDIGO DEL FRONTEND

ANEXO D. MANUAL DE USUARIO DE EXPAPP

ORDEN DE EMPASTADO