

**Conectando oportunidades: Plataforma de empleo por horas para jóvenes en Colombia.**

**Gabriel Leal Herrera**

**Ricardo Prieto**

**María José Díaz**

**Universidad EAN**

**Proyecto de Grado**

**Bogotá D.C**

**2025**

## Tabla de Contenido.

1. Introducción.....	5
2. Justificación.....	6
3. Objetivos. ....	7
3.1 Objetivo General.....	7
3.2 Objetivos Específicos.....	7
4. Planteamiento del Problema.....	8
5. Marco Teórico. ....	9
5.1 Modelos internacionales de contratación por horas.....	10
5.2 Regulación del trabajo por horas en Colombia.....	10
5.3 Plataformas digitales de empleo y su impacto en la inserción laboral juvenil .....	11
6. Análisis de requerimientos. ....	12
6.1 Fase establecimiento de requerimientos funcionales y no funcionales. ....	13
6.1.1 <i>Historias de usuario</i> .....	13
6.1.1.1 <i>Historias de usuario para el estudiante</i> .....	13
6.1.1.2 <i>Historias de usuario para el Docente</i> . ....	14
6.1.1.3 <i>Historias de usuario para el empleador</i> . ....	15
6.1.1.4 <i>Historias de usuario para el administrador</i> . ....	16
7. Análisis de Restricciones. ....	21
8. Método de Selección. ....	23
9. Método de Desarrollo. ....	25
9.1 Metodología.....	25
10. Análisis de Costos del MVP.....	27
2. Costos de Desarrollo.....	28
3. Costos Básicos (Indirectos).....	29
4. Resumen General del Análisis de Costos .....	29
11. Aplicación del método de desarrollo.....	29
11.1 Product Backlog.....	30
11.2 Análisis Arquitectura y Diseño .....	32
11.3 Arquitectura de la solución.....	32
11.4 Modelo de Datos.....	36
11.5 Prototipo no Funcional.....	39

11.6 Implementación.....	52
11.7 Pruebas y QA de Software.....	53
12. Conclusiones.....	56
13. Bibliografía.....	58

### **Lista de Figuras.**

Figura 1 Diagrama de contexto C4.....	36
Figura 2 Diagrama de contenedores.....	37
Figura 3 Diagrama de componentes.....	37
Figura 4 Modelo Relacional de la Base de Datos.....	38
Figura 5 Inicio de Sesión.....	41
Figura 6 Rol Estudiante.....	42
Figura 7 Rol Estudiante Postulaciones.....	43
Figura 8 Rol Empleador.....	45
Figura 9 Rol empleador pantallas.....	46
Figura 10 Soporte.....	46
Figura 11 Rol Mentor.....	47
Figura 12 Contenidos Mentoría.....	48
Figura 13 Soporte Mentorías.....	49
Figura 14 Rol Administrador.....	50
Figura 15 Gestión Mentorías.....	51
Figura 16 Evidencia Prueba.....	52
Figura 17 Evidencia Prueba.....	53

### **Lista de Tablas.**

Tabla 1 Comparación entre Kotlin y Flutter.....	24
Tabla 2 Ventajas y desventajas del modelo de prototipos.....	26
Tabla 3 Costos de Investigación.....	28
Tabla 4 Costos de Desarrollo.....	28
Tabla 5 Costos Básicos (Indirectos).....	29
Tabla 6 Resumen General del Análisis de Costos.....	29
Tabla 7 Product Backlog.....	30
Tabla 8 Diagrama de Casos.....	32
Tabla 9 Usuarios.....	37
Tabla 10 De Ofertas.....	37
Tabla 11 De Mensajes de Soporte.....	38
Tabla 12 de Mentorías.....	38
Tabla 13 Implementación.....	52

## Resumen.

La tecnología ha revolucionado la forma en que las personas se relacionan, creando nuevas oportunidades de interacción y crecimiento. En este contexto, el desarrollo de un Producto Mínimo Viable (MVP) permitirá aprovechar el potencial de las plataformas digitales para diseñar una solución móvil innovadora que brinde a los jóvenes acceso a trabajos por horas, mentorías con expertos y una red de apoyo estudiantil.

Esta aplicación ofrecerá a los usuarios la posibilidad de encontrar oportunidades laborales flexibles alineadas con sus habilidades e intereses, acceder a asesorías especializadas con profesionales y fortalecer su desarrollo personal y profesional dentro de una comunidad colaborativa. Con un enfoque centrado en la experiencia del usuario y el aprovechamiento de la tecnología, esta solución no solo impulsará la empleabilidad juvenil, sino que también fomentará la generación de conexiones estratégicas en un entorno digital dinámico y en constante evolución.

**Palabras clave:** MVP, empleabilidad juvenil, estudiantes universitarios, mentorías, Firebase, Android Studio, Kotlin, , arquitectura C4, trabajo por horas, aplicación móvil, desarrollo ágil, tecnología.

## 1. Introducción.

En un mundo cada vez más dinámico, los estudiantes universitarios enfrentan el desafío de generar ingresos sin comprometer su desempeño académico. La falta de oportunidades laborales flexibles, adaptadas a sus horarios y habilidades, limita su crecimiento y desarrollo profesional. En varios países, el trabajo por horas ha demostrado ser una alternativa efectiva para fomentar la inclusión laboral y la sostenibilidad económica. Economías desarrolladas como las de Estados Unidos, Australia, Alemania y el Reino Unido han evidenciado que esta modalidad no solo mejora la productividad, sino que también facilita la adaptación del empleo a las necesidades tanto de empresas como de trabajadores (The Guardian, 2023).

Sin embargo, en Colombia, la realidad es distinta. El país enfrenta desafíos económicos, sociales y tecnológicos que han generado descontento social, especialmente por la falta de empleo digno y accesible (Universidad del Rosario, 2022). Uno de los sectores más afectados es la juventud, particularmente los estudiantes de educación superior. Según el Departamento Administrativo Nacional de Estadística (DANE, 2023), la tasa de desempleo juvenil alcanza el 17.7 %, lo que evidencia la dificultad de esta población para acceder a trabajos que se ajusten a sus horarios académicos y necesidades económicas. Esta situación contribuye a la deserción estudiantil y a la precarización laboral, especialmente para aquellos en contextos socioeconómicos más vulnerables (Carrasco Ruano, 2018).

Para abordar esta problemática, se propone el desarrollo de una plataforma innovadora diseñada exclusivamente para estudiantes universitarios, conectándolos con oportunidades de trabajo por horas en sus áreas de estudio y otras actividades demandadas en el mercado. Más que una simple bolsa de empleo, esta plataforma será un ecosistema de crecimiento y apoyo, donde los estudiantes podrán acceder a empleos flexibles, interactuar

con compañeros y recibir mentorías de expertos en diversas disciplinas. A través de un sistema de mentorías dinámico, los usuarios podrán reservar sesiones personalizadas con profesionales, obteniendo orientación valiosa para potenciar su desarrollo personal y profesional.

Esta iniciativa no solo responde a una necesidad real, sino que también redefine la forma en que los estudiantes se preparan para el mundo laboral. Al combinar flexibilidad, aprendizaje colaborativo y acceso equitativo a oportunidades, esta plataforma tiene el potencial de generar un impacto transformador en la comunidad universitaria, fortaleciendo la empleabilidad y promoviendo un futuro más accesible para todos.

## **2. Justificación.**

Inicialmente, se considera pertinente mencionar que una de las principales motivaciones para desarrollar este proyecto es la necesidad de brindar a los jóvenes universitarios una alternativa de empleo flexible y accesible. La propuesta de una aplicación que les permita acceder a trabajos remunerados por horas parte de la premisa de que el contexto colombiano enfrenta diversos retos sociales, económicos y políticos, los cuales han impactado significativamente la empleabilidad juvenil.

Muchos estudiantes encuentran dificultades para conseguir empleo debido a la incompatibilidad de los horarios laborales con sus responsabilidades académicas. Según cifras del Departamento Administrativo Nacional de Estadística (DANE, 2023), la tasa de desempleo juvenil en Colombia es del 17.7 %, lo que evidencia la magnitud del problema y la necesidad de soluciones innovadoras.

Para abordar esta problemática, este proyecto de grado se enfocará en el desarrollo de un Producto Mínimo Viable (MVP) de una plataforma digital que no solo conecte a

estudiantes con empleos por horas, sino que también fomente un ecosistema de apoyo y aprendizaje. A través de esta plataforma, los estudiantes no solo podrán encontrar oportunidades laborales que se ajusten a sus horarios, sino también acceder a mentorías con expertos y generar redes de contacto con otros universitarios, promoviendo su crecimiento académico y profesional.

Este enfoque integral busca no solo mejorar la empleabilidad juvenil, sino también crear un espacio en el que los estudiantes puedan fortalecer sus habilidades, recibir orientación y prepararse para su futuro laboral. La implementación del MVP servirá como una fase inicial para validar la efectividad de la solución, permitiendo iteraciones y mejoras basadas en la experiencia de los usuarios antes de su escalabilidad.

### **3. Objetivos.**

#### **3.1 Objetivo General.**

Desarrollar un Producto Mínimo Viable (MVP) de una aplicación móvil orientada a estudiantes universitarios, que facilite el acceso a empleos por horas y a una red de mentoría, promoviendo su desarrollo profesional de forma flexible, segura y confiable.

#### **3.2 Objetivos Específicos**

1. Definir la arquitectura del MVP, asegurando el cumplimiento de los requerimientos funcionales y no funcionales para su correcto funcionamiento.
2. Seleccionar las herramientas tecnológicas adecuadas para construir un sistema escalable, seguro y eficiente.
3. Implementar medidas de seguridad, incluyendo autenticación, verificación de usuarios y protección de datos, para garantizar la confiabilidad de la plataforma.
4. Realizar pruebas de funcionalidad y usabilidad, documentando los hallazgos y aplicando mejoras para optimizar el rendimiento del MVP.

5. Elaborar la documentación técnica y académica del desarrollo, detallando el diseño, tecnologías utilizadas, pruebas realizadas y mejoras implementadas.

#### **4. Planteamiento del Problema.**

Los estudiantes universitarios en Colombia enfrentan dificultades para acceder a empleos que se adapten a sus horarios académicos y necesidades económicas. Aunque existen plataformas de empleo, muchas exigen disponibilidad de tiempo completo o experiencia previa, lo que limita sus oportunidades. Además, la falta de regulación y de incentivos para el trabajo por horas dificulta su implementación en el país, afectando tanto a los estudiantes como a las empresas (Rodríguez, 2023).

Esta situación se agrava con las brechas de desigualdad, ya que muchos jóvenes no solo tienen problemas para encontrar empleo, sino también para acceder a redes de mentoría y formación complementaria que fortalezcan sus habilidades y mejoren su perfil profesional. La escasez de opciones laborales flexibles y las altas exigencias del mercado reducen sus posibilidades de inserción laboral (Carrasco Ruano, 2018).

Ante esta problemática, se propone el desarrollo de una plataforma digital que conecte a los estudiantes con empleos por horas y mentorías con expertos, garantizando accesibilidad, seguridad y confianza. Esta solución no solo facilitará la generación de ingresos sin afectar su formación académica, sino que también contribuirá a su desarrollo profesional a través de un ecosistema de apoyo.

Con esta premisa en mente, surge la siguiente pregunta que guiara el desarrollo de este proyecto:

***¿Cómo puede una plataforma digital conectar a los estudiantes universitarios con***

*oportunidades de trabajo por horas, mentorías con expertos y una red de apoyo, garantizando seguridad, confianza y una experiencia óptima para todos los usuarios?*

## 5. Marco Teórico.

El presente estudio se fundamenta en antecedentes y enfoques tecnológicos que permiten contextualizar la problemática del desempleo juvenil en Colombia y la pertinencia de una aplicación que facilite el acceso a empleos por horas. Para ello, se analizan modelos de empleo flexible implementados en otros países, normativas laborales relevantes y herramientas tecnológicas adecuadas para el desarrollo del Producto Mínimo Viable (MVP).

El desempleo juvenil representa un desafío significativo a nivel global. Según el Banco Mundial (2023), la tasa de desempleo entre jóvenes de 15 a 24 años es del 13.8 %, con algunas regiones alcanzando índices superiores al 20 %. En Colombia, esta problemática es aún más pronunciada, con una tasa de desempleo juvenil del 17.7 % (DANE, 2023), lo que evidencia las dificultades que enfrentan los estudiantes universitarios para acceder a empleos formales y flexibles.

Uno de los principales factores que agravan la inserción laboral juvenil es la falta de experiencia y la desconexión entre el sistema educativo y las demandas del mercado laboral. La Comisión Económica para América Latina y el Caribe (CEPAL, 2003). señala que las instituciones educativas no siempre alinean sus programas con las necesidades del mercado, lo que dificulta la empleabilidad de los jóvenes egresados. Este desajuste se ve reflejado en la ausencia de oportunidades laborales acordes con las competencias adquiridas durante la formación académica.

En este contexto, las plataformas digitales de empleo han surgido como una alternativa viable para facilitar la vinculación laboral de los jóvenes, permitiéndoles acceder a

trabajos por horas que se ajusten a sus horarios académicos. Estudios previos indican que este modelo ha sido exitoso en otros países y podría representar una solución innovadora para mejorar la empleabilidad juvenil en Colombia.

### **5.1 Modelos internacionales de contratación por horas**

El trabajo por horas ha sido implementado con éxito en países como Australia, Reino Unido y Estados Unidos, donde se han desarrollado regulaciones específicas que garantizan condiciones laborales dignas para los trabajadores a tiempo parcial. Según el análisis comparativo de normativas laborales realizado en este estudio, se observa que:

- En **Australia**, el salario mínimo por hora es de \$23.23 AUD, con beneficios adicionales como cotización para jubilación y vacaciones pagadas.
- En **Reino Unido**, el salario mínimo asciende a £10.42 por hora, con una legislación que protege los derechos de los trabajadores temporales.
- En **Estados Unidos**, el salario mínimo federal es de \$7.25 por hora, aunque cada estado puede establecer montos superiores.

Estos modelos destacan la importancia de contar con regulaciones claras que protejan a los trabajadores de la precarización laboral. Sin embargo, en América Latina y sobre todo en Colombia, la implementación de este tipo de contratos enfrenta desafíos adicionales, como la alta informalidad laboral y la falta de normativas específicas que regulen esta modalidad.

### **5.2 Regulación del trabajo por horas en Colombia**

En el contexto colombiano, la legislación laboral no contempla un marco normativo claro para la contratación por horas. Aunque existen contratos de prestación de servicios y modalidades flexibles en algunos sectores, la informalidad sigue siendo un problema estructural. Según Rodríguez (2023), el 30 % de los jóvenes universitarios abandona sus

estudios por dificultades económicas, lo que evidencia la necesidad de alternativas laborales accesibles y compatibles con la educación superior.

Para que un modelo de empleo por horas sea viable en Colombia, sería necesario establecer regulaciones que garanticen derechos básicos, como acceso a seguridad social y condiciones laborales justas. En este sentido, una plataforma digital que gestione este tipo de empleos debería incorporar mecanismos de verificación y formalización que minimicen los riesgos de precarización.

### **5.3 Plataformas digitales de empleo y su impacto en la inserción laboral juvenil**

El uso de plataformas digitales para la búsqueda de empleo ha crecido significativamente en los últimos años. Ejemplos de éxito incluyen:

- **Instawork** que conecta trabajadores con empresas que requieren personal temporal, cobrando una comisión a los empleadores.
- **Freelancer** una plataforma que permite a profesionales ofrecer servicios independientes a nivel global.
- **Job Today** que facilita la contratación rápida mediante aplicaciones móviles.

Estas plataformas han demostrado que la digitalización del empleo no solo amplía el acceso a oportunidades laborales, sino que también optimiza los procesos de contratación. En el caso de Colombia, el desarrollo de una aplicación específica para estudiantes universitarios podría representar una alternativa viable para reducir la tasa de desempleo juvenil, siempre que se implemente con un enfoque que garantice la seguridad y confiabilidad de los procesos.

Finalmente, es importante mencionar que el desarrollo de un MVP debe incorporar buenas prácticas tecnológicas, promover la intermediación laboral y favorecer la vinculación de los diferentes roles en este mismo. Por lo tanto, este proyecto se centra en el desarrollo de

un Producto Mínimo Viable (MVP) que satisfaga estas demandas, basado en la utilización de tecnologías asequibles, con un diseño enfocado en el usuario y características fundamentales que faciliten la evaluación de su viabilidad y posible impacto.

## **6. Análisis de requerimientos.**

La plataforma está destinada a vincular a estudiantes universitarios con oportunidades de trabajo adecuadas para horarios flexibles, de modo que puedan asegurar empleo a tiempo parcial sin comprometer sus estudios. Su objetivo principal es disminuir el fenómeno del desempleo juvenil proporcionando un sistema intuitivo y seguro que mejore las relaciones entre los empleadores y los jóvenes en busca de empleo.

- Para asegurar que la plataforma funcione de manera efectiva, es necesario contar con parámetros clave de diseño.
- Hacer la interfaz fácil de usar.
- Sistemas de seguridad de pago, integración de pasarelas de pago para transacciones entre empleadores y empleados. Autenticación y registro de usuarios.
- Los estudiantes y empresas deberían poder registrarse utilizando correos electrónicos, números de teléfono y cuentas de redes sociales (Google, Apple, Facebook).
- Agregar recuperación de contraseña y validación en tiempo real de la información proporcionada.
- Los estudiantes pueden completar su perfil añadiendo su nombre, fotos, habilidades y disponibilidad.
- Los estudiantes pueden proporcionar detalles sobre la empresa y agregar nuevos puestos de trabajo en el perfil.
- Crear un motor de búsqueda con filtros para clasificar empleos por industria, región, nivel salarial y experiencia requerida.

- Permitir que los estudiantes apliquen a puestos de trabajo abiertos directamente desde el sitio web utilizando su CV.

## 6.1 Fase establecimiento de requerimientos funcionales y no funcionales.

### 6.1.1 Historias de usuario.

#### 6.1.1.1 Historias de usuario para el estudiante.

HISTORIA DE USUARIO		
Identificador	Nombre	
HU001	Aplicar a oferta de empleo	
Puntos de historia estimados		
Prioridad	Alta	
Responsable		
Descripción		
Como estudiante, quiero aplicar a ofertas de empleo, para conseguir una oportunidad laboral acorde a mi perfil.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El estudiante puede visualizar las ofertas de empleo	Funcionalidad
2	Se muestra un botón de "Aplicar" en cada oferta	Usabilidad
3	Al aplicar, el sistema debe confirmar la postulación	Funcionalidad
4	El estudiante recibe una notificación del estado	Disponibilidad

HISTORIA DE USUARIO		
Identificador	Nombre	
HU002	Buscar mentorías	
Puntos de historia estimados		
Prioridad	Media	
Responsable		
Descripción		
Como estudiante, quiero buscar mentorías, para obtener apoyo y guía en mi desarrollo profesional.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El estudiante puede ver las mentorías disponibles	Funcionalidad
2	Se muestran detalles del mentor y la descripción	Usabilidad
3	El estudiante puede solicitar una mentoría	Funcionalidad

HISTORIA DE USUARIO		
Identificador		
Nombre		
HU003	Actualizar perfil	
Puntos de historia estimados		
Prioridad	Media	
Responsable		
Descripción		
Como estudiante, quiero actualizar mi perfil, para que los empleadores y mentores tengan información relevante sobre mí.		
Numero de criterio		
Criterio de aceptación		
Atributo de calidad		
1	El estudiante puede editar sus datos	Funcionalidad
2	Los cambios se guardan correctamente	Usabilidad
3	Se envía una confirmación de actualización	Disponibilidad

#### 6.1.1.2 Historias de usuario para el Docente.

HISTORIA DE USUARIO		
Identificador		
Nombre		
HU004	Ofrecer mentorías	
Puntos de historia estimados		
Prioridad	Alta	
Responsable		
Descripción		
Como mentor, quiero ofrecer mentorías para ayudar a los estudiantes en su desarrollo profesional.		
Numero de criterio		
Criterio de aceptación		
Atributo de calidad		
1	El mentor puede publicar su disponibilidad	Funcionalidad
2	Se visualizan las mentorías en la plataforma	Usabilidad
3	El mentor puede aceptar solicitudes	Funcionalidad

HISTORIA DE USUARIO		
<b>Identificador</b>	<b>Nombre</b>	
HU005	Calificar estudiantes	
<b>Puntos de historia estimados</b>		
<b>Prioridad</b>	Media	
<b>Responsable</b>		
<b>Descripción</b>		
Como mentor, quiero calificar a los estudiantes con los que trabajé, para proporcionar retroalimentación y ayudar en su desarrollo.		
<b>Numero de criterio</b>	<b>Criterio de aceptación</b>	<b>Atributo de calidad</b>
1	El mentor puede acceder a un formulario de calificación	Funcionalidad
2	El feedback o nota queda registrada correctamente	Usabilidad

### 6.1.1.3 Historias de usuario para el empleador.

HISTORIA DE USUARIO		
<b>Identificador</b>	<b>Nombre</b>	
HU006	Publicar ofertas de empleo	
<b>Puntos de historia estimados</b>		
<b>Prioridad</b>	Alta	
<b>Responsable</b>		
<b>Descripción</b>		
Como empleador, quiero publicar ofertas de empleo, para atraer candidatos que cumplan con el perfil que busco.		
<b>Numero de criterio</b>	<b>Criterio de aceptación</b>	<b>Atributo de calidad</b>
1	El empleador puede llenar un formulario de empleo	Funcionalidad
2	La oferta es visible para los estudiantes	Usabilidad

HISTORIA DE USUARIO		
Identificador		
Nombre		
HU007	Gestionar postulaciones	
Puntos de historia estimados		
Prioridad		
Responsable		
Descripción		
Como empleador, quiero gestionar las postulaciones a mis ofertas, para seleccionar a los candidatos más adecuados.		
Numero de criterio		
Criterio de aceptación		
Atributo de calidad		
1	El empleador puede ver una lista de postulantes	Funcionalidad
2	El empleador puede marcar postulantes como "seleccionado" o "rechazado"	Funcionalidad

#### 6.1.1.4 Historias de usuario para el administrador.

HISTORIA DE USUARIO		
Identificador		
Nombre		
HU008	Gestionar usuarios	
Puntos de historia estimados		
Prioridad		
Responsable		
Descripción		
Como administrador, quiero gestionar los usuarios registrados, para asegurar que la plataforma sea utilizada correctamente.		
Numero de criterio		
Criterio de aceptación		
Atributo de calidad		
1	El administrador puede ver la lista de usuarios	Funcionalidad
2	Puede crear, bloquear, eliminar, actualizar y aprobar usuarios	Usabilidad

HISTORIA DE USUARIO		
Identificador	Nombre	
HU009	Generar reportes de actividad	
Puntos de historia estimados		
Prioridad		
Responsable		
Descripción		
Como administrador, quiero generar reportes de actividad en la plataforma, para monitorear el uso y rendimiento de la app.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El administrador puede seleccionar fechas y tipos de reportes	Funcionalidad
2	Los reportes se generan en vista previa	Usabilidad

### ***Requerimientos No Funcionales.***

Requerimiento NO funcional		
Identificador	Nombre	
HU001	Seguridad de la plataforma	
Puntos de historia estimados		
Prioridad	Alta	
Responsable		
Descripción		
Como usuario de la plataforma (estudiante, mentor, empleador o administrador), quiero que mis datos estén protegidos y no sean accesibles sin autorización, para sentirme seguro al usar la app.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	Los datos personales y credenciales están cifrados en la base de datos.	Funcionalidad
2	El sistema cuenta con autenticación y validación de sesión activa.	Usabilidad
3	Se aplican reglas de seguridad en Firebase que evitan lecturas/escrituras indebidas.	Disponibilidad

Requerimiento NO funcional		
<b>Identificador</b>	<b>Nombre</b>	
HU002	Rendimiento y tiempos de respuesta	
<b>Puntos de historia estimados</b>		
<b>Prioridad</b>	Media	
<b>Responsable</b>		
Descripción		
Como usuario de la aplicación, quiero que las pantallas y funciones carguen rápidamente para tener una experiencia fluida y no frustrarme con demoras.		
<b>Numero de criterio</b>	<b>Criterio de aceptación</b>	<b>Atributo de calidad</b>
1	El tiempo de carga de cada vista no debe superar los 3 segundos bajo condiciones de red estables.	Funcionalidad
3	Las animaciones o transiciones no deben generar retrasos en la interacción.	Disponibilidad

Requerimiento NO funcional		
<b>Identificador</b>	<b>Nombre</b>	
HU003	Disponibilidad del sistema	
<b>Puntos de historia estimados</b>		
<b>Prioridad</b>	Alta	
<b>Responsable</b>		
Descripción		
Como usuario registrado, quiero que la aplicación esté disponible la mayor parte del tiempo para acceder cuando lo necesite.		
<b>Numero de criterio</b>	<b>Criterio de aceptación</b>	<b>Atributo de calidad</b>
1	El sistema debe tener una disponibilidad mínima del 95% mensual.	Funcionalidad
2	En caso de caídas o mantenimiento, debe mostrarse un mensaje informativo al usuario.	Usabilidad
3	El backend en Firebase no debe superar los límites del plan gratuito sin una alternativa definida.	Disponibilidad

Requerimiento NO funcional		
<b>Identificador</b>	<b>Nombre</b>	
HU004	Compatibilidad y accesibilidad	
<b>Puntos de historia estimados</b>		
<b>Prioridad</b>	Media	
<b>Responsable</b>		
Descripción		
Como usuario, quiero poder acceder a la aplicación desde distintos dispositivos Android con diferentes resoluciones, para asegurar una buena experiencia independientemente del celular que use.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	La app debe ser compatible con al menos Android 8.0 (Oreo) en adelante.	Funcionalidad
2	El diseño debe adaptarse correctamente a pantallas desde 5 hasta 7 pulgadas.	Usabilidad
3	Los botones y textos deben tener un tamaño mínimo para garantizar su accesibilidad (WCAG nivel AA).	Disponibilidad

Requerimiento NO funcional		
<b>Identificador</b>	<b>Nombre</b>	
HU005	Mantenibilidad del código	
<b>Puntos de historia estimados</b>		
<b>Prioridad</b>	Media	
<b>Responsable</b>		
Descripción		
Como desarrollador del equipo, quiero que el código esté bien documentado y estructurado, para que futuras actualizaciones y correcciones se puedan realizar de forma eficiente.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	Todo componente debe tener comentarios que expliquen su propósito.	Funcionalidad
2	Se debe usar control de versiones (Git) para cada funcionalidad nueva.	Usabilidad
3	El código debe seguir convenciones de estilo definidas por el equipo (por ejemplo, Kotlin coding standards).	Disponibilidad

Requerimiento NO funcional		
<b>Identificador</b>	<b>Nombre</b>	
HU006	Escalabilidad del sistema	
<b>Puntos de historia estimados</b>		
<b>Prioridad</b>	Media	
<b>Responsable</b>		
Descripción		
Como administrador del sistema, quiero que la plataforma esté diseñada para crecer en número de usuarios y funciones sin que el rendimiento se degrade.		
<b>Numero de criterio</b>	<b>Criterio de aceptación</b>	<b>Atributo de calidad</b>
1	Firestore debe permitir migración a planes pagos sin reescribir el código base.	Funcionalidad
2	Las consultas a Firestore deben estar indexadas para mejorar tiempos de respuesta.	Usabilidad
3	Las colecciones deben permitir paginación para evitar sobrecargas de lectura.	Disponibilidad

Requerimiento NO funcional		
<b>Identificador</b>	<b>Nombre</b>	
HU007	Análítica y trazabilidad	
<b>Puntos de historia estimados</b>		
<b>Prioridad</b>	Media	
<b>Responsable</b>		
Descripción		
Como administrador, quiero poder monitorear el uso de la app, para detectar problemas, medir la adopción y tomar decisiones basadas en datos.		
<b>Numero de criterio</b>	<b>Criterio de aceptación</b>	<b>Atributo de calidad</b>
1	Deben registrarse eventos importantes (registro, inicio de sesión, postulaciones).	Funcionalidad
2	Los logs deben almacenarse de forma segura y organizada.	Usabilidad
3	Debe integrarse con una herramienta de analítica como Firebase Analytics.	Disponibilidad

Requerimiento NO funcional		
<b>Identificador</b>	<b>Nombre</b>	
HU008	Facilidad de despliegue	
<b>Puntos de historia estimados</b>		
<b>Prioridad</b>	Media	
<b>Responsable</b>		
Descripción		
Como miembro del equipo de desarrollo, quiero que el proceso de despliegue de la aplicación sea automatizado y simple, para asegurar entregas constantes sin errores manuales.		
<b>Numero de criterio</b>	<b>Criterio de aceptación</b>	<b>Atributo de calidad</b>
1	La app se debe poder compilar y ejecutar con un solo comando desde Android Studio.	Funcionalidad
2	Las credenciales y configuraciones deben estar separadas del código fuente.	Usabilidad
3	Se debe documentar el proceso de despliegue paso a paso.	Disponibilidad

Requerimiento NO funcional		
<b>Identificador</b>	<b>Nombre</b>	
HU009	Recuperación ante errores	
<b>Puntos de historia estimados</b>		
<b>Prioridad</b>	Media	
<b>Responsable</b>		
Descripción		
Como usuario, quiero que la aplicación maneje errores de forma clara, para saber qué hacer cuando algo sale mal y evitar perder información.		
<b>Numero de criterio</b>	<b>Criterio de aceptación</b>	<b>Atributo de calidad</b>
1	Deben mostrarse mensajes de error comprensibles en lugar de fallas técnicas.	Funcionalidad
2	Las transacciones deben tener rollback si ocurre un error crítico.	Usabilidad
3	El sistema debe registrar internamente los errores para revisión del equipo técnico.	Disponibilidad

## 7. Análisis de Restricciones.

### Restricción Técnica – Dependencia de Firebase

El MVP depende de servicios de Firebase como Auth y Firestore en modo gratuito (Spark plan). Esto limita el número de operaciones por día y almacenamiento, y sugiere que el sistema no es aún escalable sin migración a un plan de pago.

### **Restricción de Seguridad – Configuración de Firestore**

Actualmente la base de datos está en modo test, lo que implica que cualquier usuario fuera de los servidores puede leer/escribir datos mediante alguna vulneración del sistema. Por lo que es obligatorio implementar reglas de seguridad para evitar bloqueo del sistema.

### **Restricción Temporal – Plazo Corto para el Desarrollo**

El equipo cuenta con un tiempo limitado aproximadamente 4 meses para desarrollar y probar el MVP. Esto limita la cantidad y complejidad de funcionalidades que se pueden implementar y obliga a priorizar tan solo lo esencial (registro, login, vista por rol, mentorías y ofertas).

### **Restricción Financiera – Presupuesto Limitado**

El proyecto tiene costos directos estimados y por lo tanto no contempla gastos en marketing, servidores adicionales o soporte extendido. Lo cual restringe su crecimiento y alcance inicial.

### **Restricción Humana – Equipo de Desarrollo Pequeño**

El equipo está compuesto por 3 personas que asumen múltiples roles (desarrollo, diseño, pruebas). Esto puede generar sobrecarga de trabajo y riesgos de retrasos o errores si no se gestiona adecuadamente el tiempo.

## **8. Método de Selección.**

### **Metodología para la selección de herramientas de software en el desarrollo del MVP.**

Para el desarrollo de un Producto Mínimo Viable (MVP), la selección de herramientas de software juega un papel fundamental en la eficiencia, escalabilidad y facilidad de integración. Para esta decisión se tomaron 3 criterios esenciales que permitieron llevar a cabo la evaluación de las distintas opciones tecnológicas disponibles.

#### **Facilidad de Integración con Bases de Datos**

Un MVP requiere una conexión rápida y estable con bases de datos, ya que estas gestionan la información de los usuarios y los servicios del sistema. En este caso, Firebase ha sido la opción seleccionada debido a su flexibilidad en el manejo de datos y su compatibilidad con Android Studio y Kotlin.

#### **Compatibilidad con APIs y Servicios Externos**

Dado que el MVP necesita interactuar servicios terceros, Kotlin junto a Android Studio proporciona SDK nativos que facilitan la conexión con estos servicios. En contraste, aunque Flutter permite integraciones similares, requiere configuraciones adicionales y el uso de paquetes externos, lo que puede representar un esfuerzo adicional en la implementación del MVP.

#### **Eficiencia en el Desarrollo y Despliegue**

Para garantizar que el MVP se desarrolle en el menor tiempo posible, es esencial contar con herramientas que faciliten la depuración, la gestión de dependencias y la integración con entornos de desarrollo. En este aspecto, Android Studio se destaca por proporcionar herramientas avanzadas que optimizan la eficiencia del código. Por otro lado,

Flutter, aunque potente en el desarrollo multiplataforma, requiere la adopción de su arquitectura basada en widgets, lo que implica un proceso de aprendizaje y adaptación más prolongado. Esto podría afectar la velocidad de desarrollo en la etapa inicial del MVP.

*Tabla 1 Comparación entre Kotlin y Flutter*

<b>Criterio</b>	<b>Kotlin (Android Studio)</b>	<b>Flutter (Dart)</b>
Integración con bases de datos	Soporte nativo para Firebase, con SDKs optimizados para Android.	Compatible con Firebase, pero requiere configuración extra en algunas funcionalidades.
Curva de aprendizaje	El equipo ya tiene <b>experiencia en Kotlin</b> , lo que reduce tiempos de desarrollo.	Requiere aprender Dart y la arquitectura de Flutter, aumentando la curva de aprendizaje.
Eficiencia en desarrollo	Android Studio ofrece herramientas avanzadas de testing y optimización para apps nativas.	Flutter tiene herramientas propias, pero su depuración y optimización pueden requerir más tiempo.
Compatibilidad con APIs	Kotlin tiene SDKs oficiales para Android, lo que facilita la integración con APIs y servicios de terceros.	Flutter usa paquetes de terceros para integraciones, lo que puede generar dependencias externas adicionales.
Rendimiento	Rendimiento nativo optimizado para Android.	Aunque Flutter tiene buen rendimiento, su capa intermedia puede afectar la eficiencia en algunas animaciones o procesos pesados.

**Nota.** Elaboración propia con base en los criterios.

### **¿Por qué Kotlin y Android Studio para este MVP?**

Kotlin con Android Studio es la mejor opción para el desarrollo del MVP debido a su facilidad de integración con Firebase, su compatibilidad con APIs y servicios externos, y la experiencia previa del equipo en este lenguaje, lo que reduce la curva de aprendizaje.

Además, Android Studio proporciona herramientas avanzadas de depuración, testing y optimización, permitiendo un desarrollo más eficiente y rápido. Aunque Flutter es una alternativa viable para aplicaciones multiplataforma, en esta fase del proyecto se prioriza la rapidez y estabilidad que ofrece Kotlin dentro del ecosistema de Android.

## **9. Método de Desarrollo.**

### **9.1 Metodología.**

Para el desarrollo de este proyecto, se emplearán metodologías ágiles, incluyendo enfoques relacionados con Scrum, que fomentan la flexibilidad, colaboración y entrega incremental. Estas metodologías permitirán la adaptabilidad a los cambios y la entrega continua de valor al cliente a través de iteraciones cortas.

Gracias a su estructura iterativa, este enfoque facilitará:

- Iteraciones rápidas y continuas, alineadas con el modelo de prototipos.
- Mejor organización del equipo de desarrollo.
- Entrega incremental del producto.
- Mayor transparencia y adaptación a los cambios en los requerimientos.

El uso de prácticas relacionadas con Scrum permitirá estructurar el trabajo en ciclos cortos, favoreciendo la retroalimentación constante y asegurando que el MVP evolucione de manera eficiente y alineada con las necesidades del proyecto.

### **Ciclo de vida del software sobre la App**

El ciclo de vida trata un proceso estructurado que permite gestionar el desarrollo, implementación y mantenimiento un producto de manera eficiente. Comúnmente, este ciclo

incluye fases como el análisis de requerimientos, diseño, implementación, pruebas, despliegue y mantenimiento. Garantizando calidad, funcionalidad y sostenibilidad.

El modelo que se empleara es conocido por modelo de prototipos. Este modelo se caracteriza por la creación de una versión preliminar del sistema que permite validar requerimientos y mejorar el producto final a través de iteraciones sucesivas. Esta metodología facilita la comunicación entre desarrolladores y usuario final. En este enfoque, el prototipo inicial es evaluado por los usuarios, quienes proporcionan retroalimentación que permite refinar y mejorar el diseño antes de su implementación definitiva. El modelo de prototipos se utiliza este proyecto debido a la necesidad de validar antes de su desarrollo completo.

La creación de prototipos permitirá realizar pruebas tempranas con usuarios reales, detectar posibles problemas de usabilidad y optimizar el diseño antes de su implementación. Además, este enfoque reducirá riesgos asociados a la falta de alineación con las necesidades del público objetivo. A continuación, se presenta una tabla con las ventajas y desventajas del modelo de prototipos:

*Tabla 2 Ventajas y desventajas del modelo de prototipos.*

<b>Ventajas</b>	<b>Desventajas</b>
Permite una mejor comprensión de los requisitos del usuario.	Puede aumentar los costos si se realizan demasiadas iteraciones.
Facilita la detección temprana de errores y mejoras.	Puede generar expectativas irreales en los usuarios si el prototipo no refleja la funcionalidad final.
Mejora la comunicación entre los desarrolladores y los usuarios.	Puede llevar a una planificación deficiente si no se gestiona correctamente.
Reduce el riesgo de desarrollar un sistema que no satisfaga las necesidades del cliente.	Puede retrasar el desarrollo final si no se controla el número de revisiones.
Fomenta la iteración y mejora continua del diseño.	Requiere mayor participación de los usuarios, lo que puede prolongar el proceso de desarrollo.

**Nota.** Elaboración propia.

## 10. Análisis de Costos del MVP

Para el desarrollo del MVP (Producto Mínimo Viable) de Conectando App, se realizó un análisis de costos que contempla tanto los costos directos como los costos indirectos, con base en el trabajo de tres integrantes del equipo: María José Díaz, Gabriel Leal y Ricardo Prieto. Las variables clave se agrupan en tres categorías: investigación, desarrollo y costos básicos.

### Costos de Investigación

Corresponden a las actividades previas al desarrollo, tales como: levantamiento de requerimientos, análisis de usuarios, diseño conceptual y planeación inicial del proyecto.

- Tiempo estimado: 20 horas por persona.
- Valor hora: \$70.000 COP.
- Duración: Trabajo concentrado en la primera fase del proyecto.

Costo total investigación:

$$20 \text{ horas} \times \$70.000 \times 3 \text{ personas} = \$4.200.000 \text{ COP}$$

### Costos de Desarrollo

Incluyen el trabajo técnico en el diseño, programación, integración y pruebas de las funcionalidades esenciales del MVP, como el login, registro, conexión a Firebase y pantallas por rol.

- Tiempo estimado: 30 horas por persona.
- Valor hora: \$130.000 COP (considerando el perfil técnico del equipo).
- Duración: Distribuido en un período de 4 a 6 semanas.

Costo total desarrollo:

$$30 \text{ horas} \times \$130.000 \times 3 \text{ personas} = \$11.700.000 \text{ COP}$$

### **Costos Básicos (Costos Indirectos)**

Reflejan los apoyos logísticos, servicios o recursos mínimos sostenidos durante el desarrollo del MVP (internet, herramientas, equipos, hosting, entre otros).

Duración: 4 meses de trabajo.

Valor mensual estimado por persona: \$100.000 COP.

Costo total básicos:

$$4 \text{ meses} \times \$100.000 \times 3 \text{ personas} = \$1.200.000 \text{ COP.}$$

### **1. Costos de Investigación**

*Tabla 3 Costos de Investigación*

Nombre	Horas trabajadas	Valor/hora	Total
María José Díaz	20	\$70.000	\$1.400.000
Gabriel Leal	20	\$70.000	\$1.400.000
Ricardo Prieto	20	\$70.000	\$1.400.000
Total			\$4.200.000

**Nota.** Elaboración propia.

### **2. Costos de Desarrollo**

*Tabla 4 Costos de Desarrollo*

Nombre	Horas trabajadas	Valor/hora	Total
María José Díaz	30	\$130.000	\$3.900.000
Gabriel Leal	30	\$130.000	\$3.900.000
Ricardo Prieto	30	\$130.000	\$3.900.000
Total			\$11.700.000

**Nota.** Elaboración propia.

### 3. Costos Básicos (Indirectos)

*Tabla 5 Costos Básicos (Indirectos)*

Nombre	Meses trabajados	Valor/mes	Total
María José Díaz	4	\$100.000	\$400.000
Gabriel Leal	4	\$100.000	\$400.000
Ricardo Prieto	4	\$100.000	\$400.000
Total			\$1.200.000

**Nota.** Elaboración propia.

### 4. Resumen General del Análisis de Costos

*Tabla 6 Resumen General del Análisis de Costos*

Tipo de costo	Total
Costos Directos	\$15.900.000
Costos Indirectos	\$1.200.000
<b>TOTAL, GENERAL</b>	<b>\$17.100.000</b>

**Nota.** Elaboración propia.

## 11. Aplicación del método de desarrollo.

Para el desarrollo del MVP de “Conectando Oportunidades” se aplicó una metodología ágil basada en Scrum y el modelo de prototipos, lo que permitió avanzar de forma iterativa y con entregas funcionales frecuentes. Cada ciclo de desarrollo (sprint) tuvo una duración de dos semanas, y al finalizar cada uno se revisaron avances y se recolectó retroalimentación de usuarios reales. Desde las primeras fases se diseñaron prototipos en Miro, ayudando a ajustar los flujos de navegación, priorizar funciones clave como la postulación y la creación de mentorías.

El desarrollo se realizó en Android Studio con Kotlin, utilizando Firebase como backend para autenticación, base de datos y almacenamiento. Esta combinación de enfoque ágil y validación con prototipos permitió adaptar el proyecto a las necesidades reales del público objetivo, reducir riesgos y lograr un MVP funcional, usable y con proyección de mejora continua.

## 11.1 Product Backlog

Tabla 7 Product Backlog.

Código HU	Rol	Requerimiento	¿Para qué?	Criterios de aceptación
HU001	Estudiantes	Aplicar a ofertas de empleo, para conseguir una oportunidad laboral acorde al perfil o estudio.	Conseguir ese primere empleo que pueda extender su conocimiento en su carrera o conseguir dinero extra.	<ol style="list-style-type: none"> <li>1. El estudiante ingresa a la app.</li> <li>2. Visualiza las ofertas disponibles.</li> <li>3. Hace clic en "Aplicar".</li> </ol>
HU002	Estudiantes	Como estudiante, quiero buscar mentorías, para obtener apoyo y guía en mi desarrollo profesional.	Con el fin de que alguien pueda traspasar ese conocimiento extra que necesite para su profesión.	<ol style="list-style-type: none"> <li>1. El estudiante puede ver las mentorías disponibles</li> <li>2. Se muestran detalles del mentor y la descripción</li> <li>3. El estudiante puede solicitar una mentoría</li> </ol>
HU003	Estudiantes	Como estudiante, quiero actualizar mi perfil, para que los empleadores y mentores tengan información relevante sobre mí.	Mantener el perfil actualizado, corregir algún dato o agregar información personal.	<ol style="list-style-type: none"> <li>1. El estudiante puede editar sus datos.</li> <li>2. Los cambios se guardan correctamente</li> <li>3. Se envía una confirmación de actualización.</li> </ol>
HU004	Mentores	Como mentor, quiero ofrecer mentorías para ayudar a los estudiantes en su	Dar ese conocimiento extra que puede darle a un estudiante que necesite centrarse en su carrera.	<ol style="list-style-type: none"> <li>1. El mentor puede publicar su disponibilidad.</li> <li>2. Se visualizan las mentorías</li> </ol>

		desarrollo profesional.		en la plataforma. 3. El mentor puede aceptar solicitudes.
HU005	Mentores	Como mentor, quiero calificar a los estudiantes con los que trabajé, para proporcionar retroalimentación y ayudar en su desarrollo.	Poder tener un criterio de que el estudiante está aprendiendo.	1. El mentor puede acceder a un formulario de calificación. 2. El feedback o nota queda registrada correctamente.
HU006	Empleador	Como empleador, quiero publicar ofertas de empleo, para atraer candidatos que cumplan con el perfil que busco.	Obtener nuevos talentos y de ser posible, formar un nuevo empleado.	1. El empleador puede llenar un formulario de empleo. 2. La oferta es visible para los estudiantes.
HU007	Empleador	Como empleador, quiero gestionar las postulaciones a mis ofertas, para seleccionar a los candidatos más adecuados.		1. El empleador puede ver una lista de postulantes 2. El empleador puede marcar postulantes como "seleccionado" o "rechazado".
HU008	Administrador	Como administrador, quiero gestionar los usuarios registrados, para asegurar que la plataforma sea utilizada correctamente.	Dado caso poder bloquear usuarios que no usen correctamente la aplicación.	1. El administrador puede ver la lista de usuarios. 2. Puede crear, bloquear, eliminar, actualizar y aprobar usuarios.
HU009	Administrador	Como administrador, quiero generar reportes de actividad en la plataforma, para monitorear el uso y rendimiento de la app.	Con estos datos podremos realizar toma de decisiones para el mejoramiento de la app.	1. El administrador puede seleccionar fechas y tipos de reportes. 2. Los reportes se generan en vista previa.

**Nota.** Elaboración propia.

## 11.2 Análisis Arquitectura y Diseño

Diagrama de Casos de uso con su respectiva descripción.

*Tabla 8 Diagrama de Casos.*

<b>Nombre del caso</b>	<b>Descripción</b>	<b>Actor principal</b>
<b>UC1 - Registrarse / Iniciar sesión</b>	Permite a cualquier usuario acceder a la plataforma mediante autenticación segura.	Todos
<b>UC2 - Aplicar a ofertas de empleo</b>	El estudiante se postula a un empleo disponible dentro de la plataforma.	Estudiante
<b>UC3 - Buscar y registrarse a mentorías</b>	Permite al estudiante ver mentorías disponibles y solicitar inscripción a una.	Estudiante
<b>UC4 - Ver historial de postulaciones</b>	El estudiante consulta un registro de sus aplicaciones pasadas, con estados y detalles.	Estudiante
<b>UC5 - Ver/Editar perfil</b>	Permite a cualquier usuario actualizar sus datos personales y visualizar su calificación.	Todos
<b>UC6 - Publicar ofertas de empleo</b>	El empleador crea una nueva oferta laboral incluyendo toda la información del puesto.	Empleador
<b>UC7 - Gestionar postulaciones</b>	Permite al empleador revisar postulantes y cambiar su estado (aceptado / rechazado).	Empleador
<b>UC8 - Ofrecer mentorías</b>	El mentor crea una nueva sesión de mentoría ingresando tema, horario y duración.	Mentor
<b>UC9 - Gestionar usuarios</b>	El administrador puede crear, editar, aprobar, bloquear o eliminar cuentas dentro de la plataforma.	Administrador
<b>UC10 - Generar reportes de actividad</b>	El administrador puede generar reportes de uso (por fechas, por tipo de usuario, etc.)	Administrador

*Nota. Elaboración propia.*

## 11.3 Arquitectura de la solución.

La arquitectura del MVP *Conectando* se estructura bajo el modelo C4, que permite representar el sistema en distintos niveles de detalle, facilitando su comprensión y desarrollo.

En el nivel de contexto, la aplicación móvil conecta a estudiantes universitarios con oportunidades laborales por horas y mentorías. Los principales actores son el estudiante (usuario central), el empleador, el mentor y el administrador. Todos interactúan a través de la app móvil, que funciona como punto de acceso único.

En el nivel de contenedores, la solución se compone de:

Una aplicación móvil desarrollada en Kotlin con Android Studio, que contiene toda la lógica de interfaz :

Un backend basado en Firebase, que incluye:

- **Authentication** para gestionar roles y accesos.
- **Firestore** base de datos en tiempo real.
- **Analytics** para rastrear el comportamiento de los usuarios (Herramienta que proporciona Firebase).
- **Firestore** para lógica de backend automatizada junto con almacenamiento en cloud.

En el nivel de componentes, la app se divide en módulos según el rol del usuario:

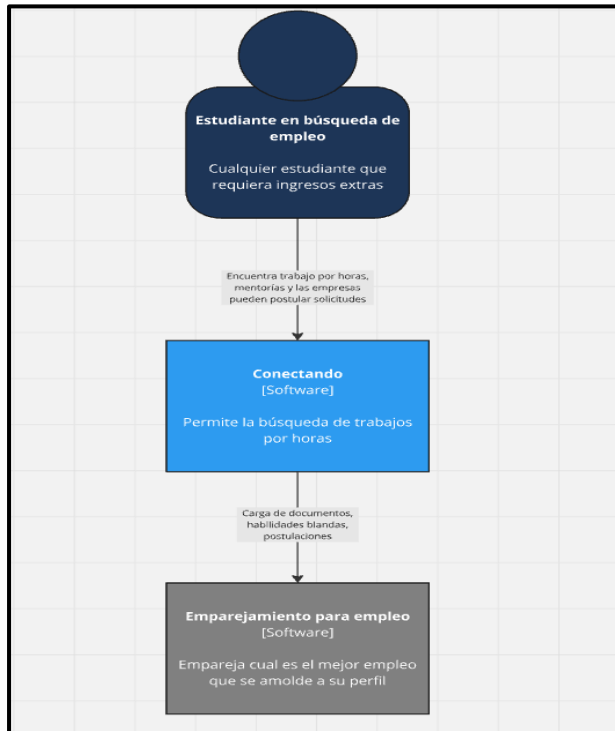
- Estudiantes: acceso a módulos de ofertas, mentorías y perfil.
- Empleadores: publicación y gestión de vacantes.
- Mentores: creación y administración de sesiones.
- Administradores: gestión global de usuarios, mentorías y empleos.

Todos los roles comparten acceso al módulo de soporte técnico, y la información se sincroniza en tiempo real gracias a la integración con Firebase.

## Diagrama de contexto C4

Figura 1

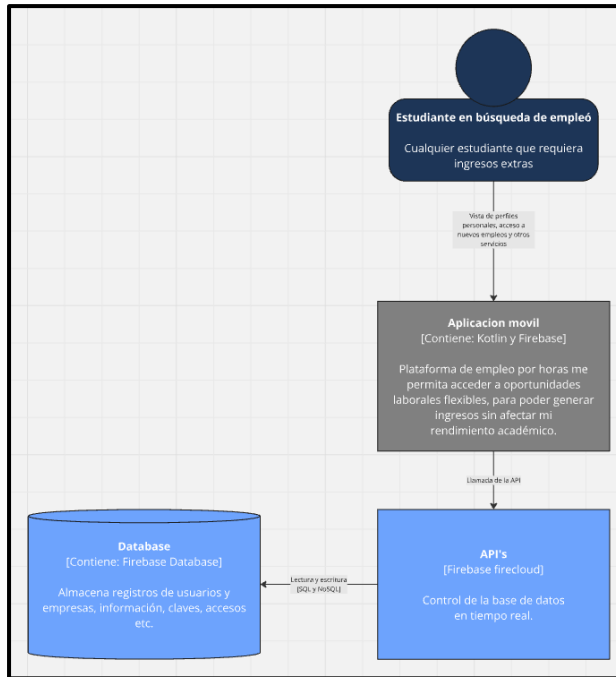
Diagrama de contexto C4



**Nota.** Elaboración propia

Figura 2

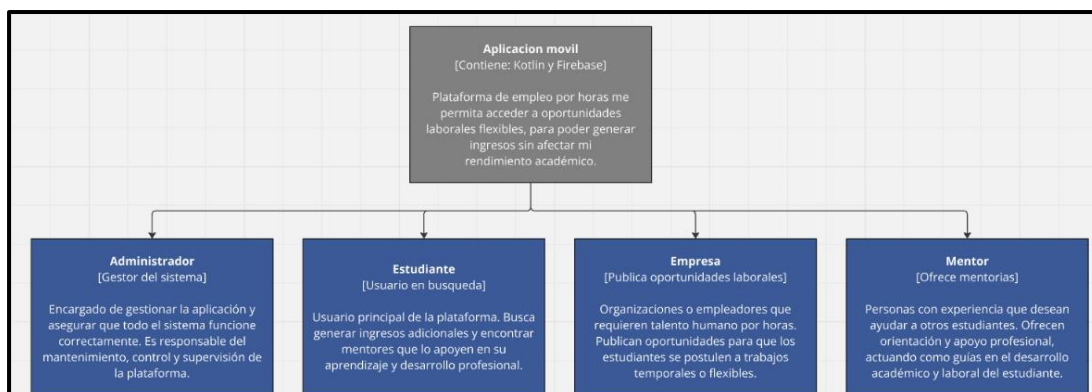
Ilustración 1 Diagrama de contenedores.



**Nota.** Elaboración propia

Figura 3

Ilustración 2 Diagrama de componentes



**Nota.** Elaboración propia

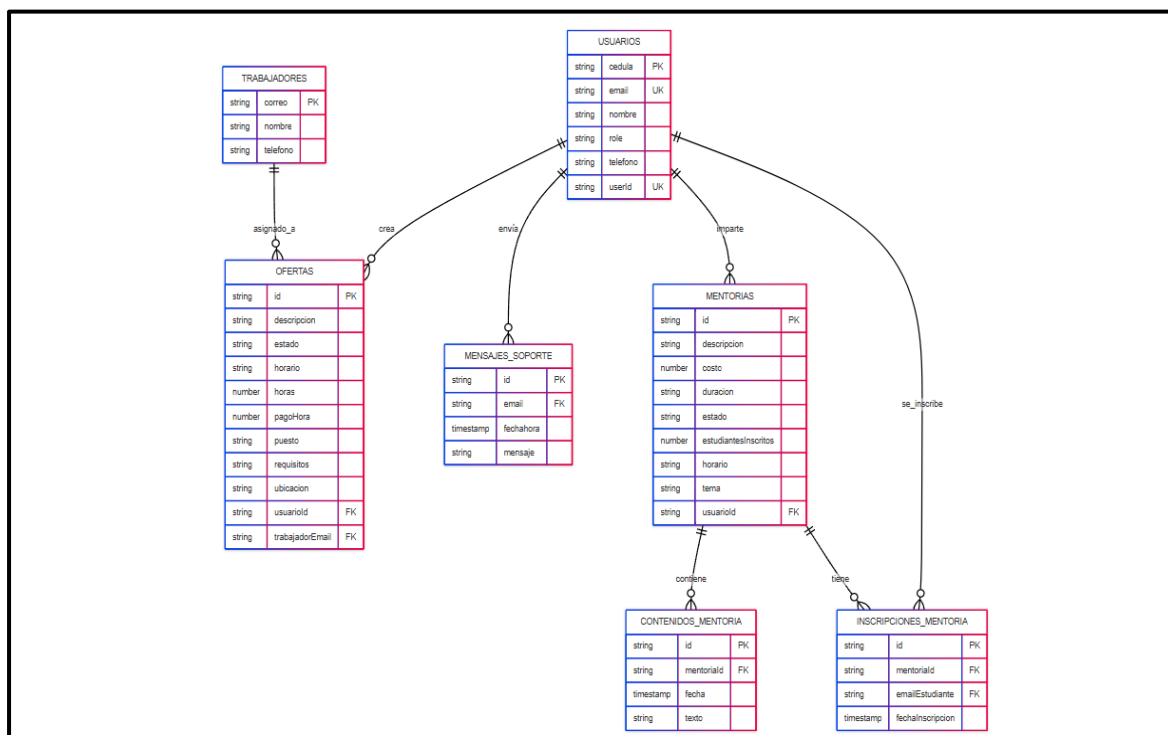
## 11.4 Modelo de Datos.

### 11.4.1 Modelo Relacional de la Base de Datos.

Este modelo de datos está creado para evidenciar las bases de datos de la app Conectando donde tiene el propósito de unir a estudiantes, mentores y empleadores en un entorno educativo y laboral. Esta herramienta tecnológica ayuda a manejar trabajos temporales, programas de tutoría y formas de comunicación con asistencia técnica.

Figura 4

Ilustración 3 Modelo Relacional de la Base de Datos.



**Nota.** Elaboración propia

### 11.4.2 Diccionario de datos.

Tabla 9 Usuarios.

<b>Campo</b>	<b>Tipo de dato</b>	<b>Descripción</b>
cedula	String	Número de identificación del usuario (cédula).
email	String	Correo electrónico del usuario.
nombre	String	Nombre completo del usuario.
role	String	Rol asignado dentro del sistema (ej. "Estudiante", "Administrador").
teléfono	String	Número de contacto telefónico del usuario.
userId	String	Identificador único del usuario, generalmente relacionado con la autenticación (Firebase Auth).

**Nota.** Elaboración propia

Tabla 10 De Ofertas.

<b>Campo</b>	<b>Tipo de dato</b>	<b>Descripción</b>
descripción	String	Texto que describe el trabajo o actividad a realizar.
estado	String	Estado actual de la postulación o tarea (ej. "Pagada", "Pendiente").
horario	String	Franja u horario asignado (se representa como texto).
horas	Number	Número de horas trabajadas o programadas.
id	String	Identificador único del documento o tarea.
pagoHora	Number	Valor del pago por hora.
Puesto	String	Cargo o función para desarrollar.
requisitos	String	Requisitos solicitados para el puesto.
ubicación	String	Dirección o lugar de ubicación de la oferta.
usuarioId	String	ID del usuario que creo la oferta
trabajador	Map	Información anidada del trabajador asignado.
correo.trabajador	String	Correo electrónico del trabajador.
nombre.trabajador	String	Nombre completo del trabajador.
telefono.trabajador	String	Teléfono de contacto del trabajador.

**Nota.** Elaboración propia

*Tabla 11 De Mensajes de Soporte*

<b>Campo</b>	<b>Tipo de dato</b>	<b>Descripción</b>
email	String	Correo electrónico del usuario que envió el mensaje a soporte.
fechahora	Marca de tiempo (timestamp)	Fecha y hora en que se envió el mensaje (formato de Firebase Timestamp).
mensaje	String	Contenido del mensaje enviado por el usuario.

**Nota.** Elaboración propia

*Tabla 12 de Mentorías*

<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>
contenidos	Map	Lista de contenidos asociados a la mentoría.
contenidos.fecha	Marca de tiempo	Fecha del contenido.
contenidos.texto	String	Texto del contenido.
costo	Número	Costo de la mentoría.
descripcion	String	Descripción general de la mentoría.
duracion	String	Duración de la mentoría.
estado	String	Estado actual de la mentoría.
estudiantesInscritos	Número	Cantidad de estudiantes inscritos.
horario	String	Horario de la mentoría.
id	String	ID único de la mentoría.
inscritos	Map	Lista de correos de inscritos.
tema	String	Tema principal de la mentoría.
usuarioId	String	ID del usuario que creó la mentoría.

**Nota.** Elaboración propia.

## 11.5 Prototipo no Funcional.

### Estrategias de Evaluación de requerimiento (Mockups)

Figura 4

#### Inicio de Sesión



**Nota.** Elaboración propia

Este conjunto de pantallas representa los puntos de entrada fundamentales para todos los usuarios de la aplicación móvil Conectando, sin importar su rol (Estudiante, Empleador, Mentor o Administrador). Estas interfaces están diseñadas para garantizar un acceso rápido, claro y seguro a la plataforma, ya sea para iniciar sesión o para crear una nueva cuenta.

La pantalla de Inicio de Sesión (Log In) funciona como la puerta de entrada para los usuarios ya registrados en Conectando. Presenta campos estándar para ingresar el correo electrónico y la contraseña. Un botón destacado de Log In permite al usuario ingresar directamente a su perfil según el rol previamente seleccionado. Justo debajo, una invitación con el texto ¿Nuevo en Conectando? ¡Únete! enlaza a la pantalla de registro para aquellos que aún no tienen una cuenta.

La pantalla de Creación de Cuenta (Registrarse) está pensada para facilitar el ingreso de nuevos usuarios a la comunidad de Conectando. Aquí se recopilan datos básicos y necesarios para configurar el perfil:

- Nombre completo
- Cédula
- Teléfono
- Correo electrónico
- Contraseña

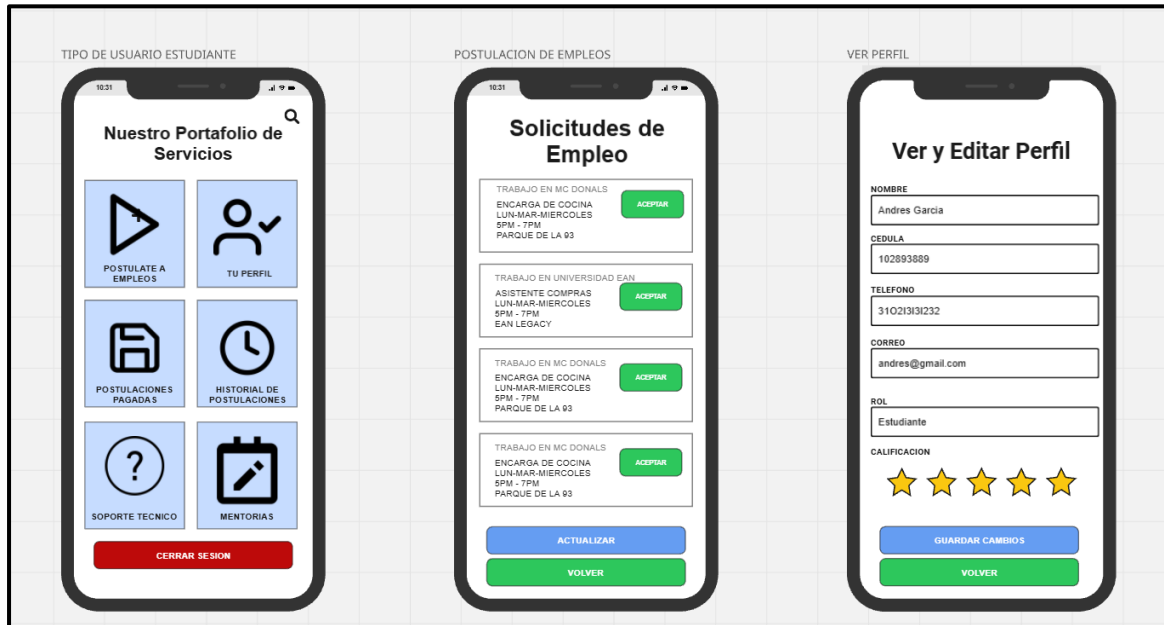
Además, se incluye un campo fundamental: la selección del Rol del usuario, donde se debe elegir entre Estudiante, Empleador, Mentor o Administrador. Esta elección define automáticamente las funcionalidades que tendrá disponibles el usuario dentro de la app, personalizando su experiencia según su perfil.

Una vez completados todos los campos, el botón Registrar permite finalizar el proceso y crear la cuenta en la plataforma. A partir de allí, el nuevo usuario podrá acceder al contenido y herramientas específicas de su rol en Conectando.

Vista Rol Estudiante

Figura 5

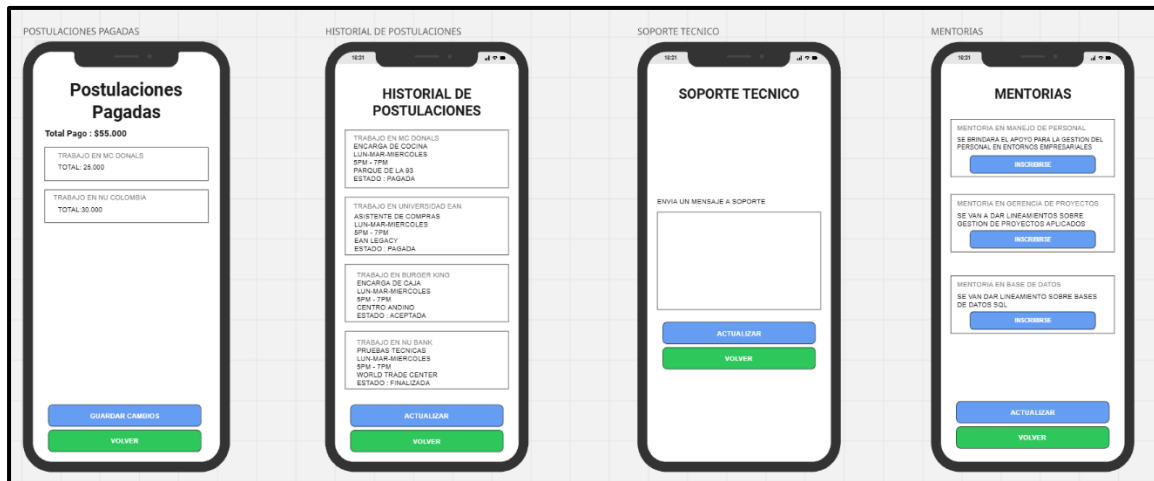
Rol Estudiante



Nota. Elaboración propia

Figura 6

Rol Estudiante Postulaciones



Nota. Elaboración propia

Este conjunto de pantallas representa toda la experiencia pensada para el perfil de Estudiante dentro de la aplicación móvil Conectando, una plataforma completa y multifuncional. La app está diseñada para conectar a los estudiantes con ofertas de trabajo flexibles por horas y brindarles recursos para su desarrollo profesional, así como permitirles interactuar con empresas, mentores y administradores.

El recorrido del estudiante inicia en la pantalla Nuestro portafolio de servicios, que funciona como su menú principal. Desde ahí, puede acceder fácilmente a funciones clave como:

- Postulación de empleos
- Mentorías
- Historial de postulaciones
- Postulaciones pagadas
- Asistencia técnica
- Cerrar sesión

En la sección de Empleos, la pantalla Solicitudes de empleo muestra una lista de trabajos por horas disponibles, con detalles como el tipo de empleo, ubicación y horarios. Cada oferta cuenta con un botón Aceptar para postularse rápidamente, y opciones para actualizar la lista o volver al menú anterior. La pantalla Ver y editar perfil permite al estudiante revisar y modificar su información personal: nombre, cédula, teléfono y correo electrónico. También incluye una calificación con estrellas, que actúa como un sistema de reputación visible para empresas y mentores. Esta pantalla es uniforme para todos los perfiles (empleador, mentor y administrador), lo que contribuye a una experiencia coherente y amigable.

Para el seguimiento de sus actividades, el estudiante cuenta con dos secciones clave:

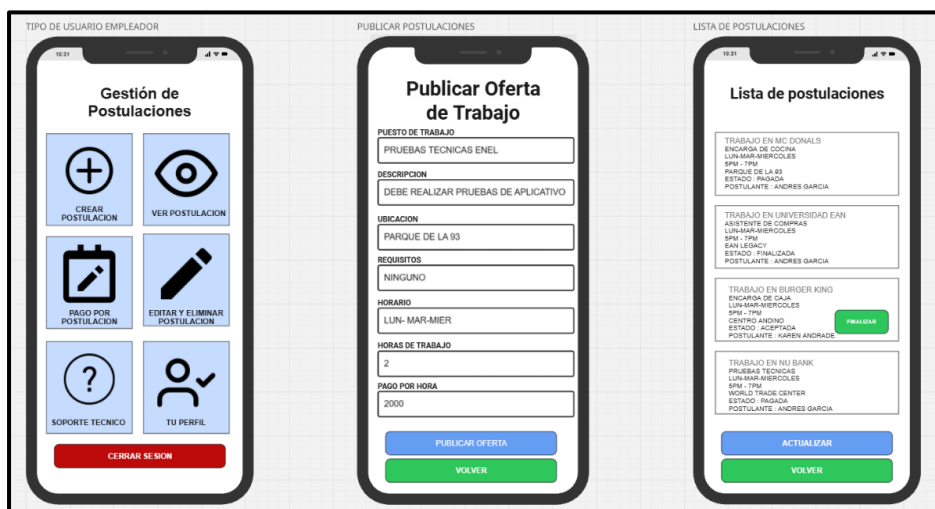
- Postulaciones pagadas, donde puede ver el total ganado y un desglose detallado de los pagos recibidos por cada trabajo.
- Historial de postulaciones, que muestra todas las ofertas a las que se ha postulado, con información sobre la empresa, el cargo, el lugar, el horario y el estado de la solicitud (pendiente, aceptada, finalizada o pagada).

En caso de requerir ayuda, puede acceder a la sección de Asistencia Técnica, desde donde puede comunicarse con el equipo de soporte para resolver dudas o problemas. Esta funcionalidad también está disponible para los demás perfiles de usuario.

Finalmente, la sección de Mentorías ofrece al estudiante espacios de aprendizaje en áreas clave como administración de proyectos, gestión de personal o bases de datos. Cada mentoría se presenta con una descripción breve y un botón Regístrate que facilita la inscripción.

Figura 5

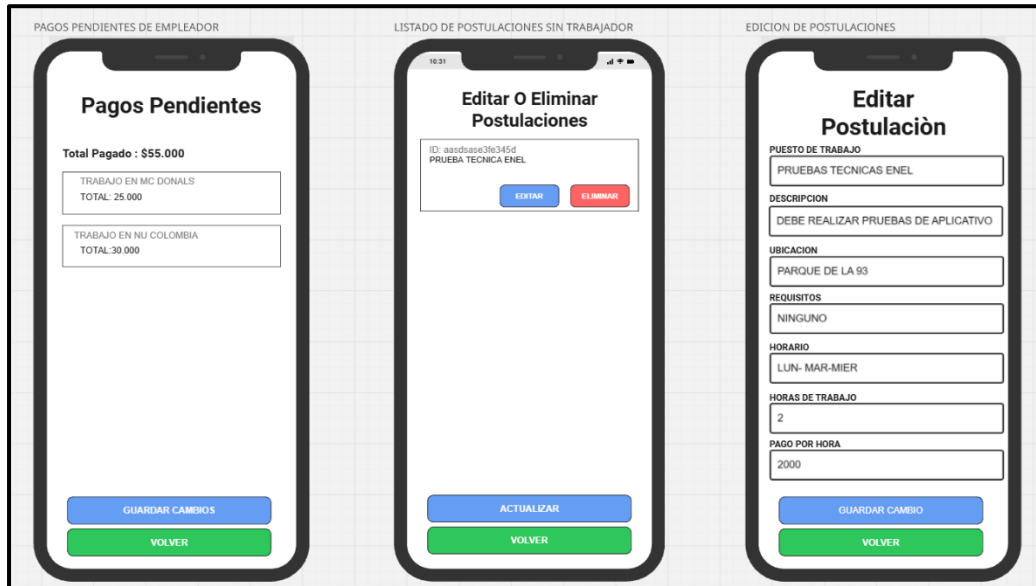
### Rol Empleador



Nota. Elaboración propia

Figura 6

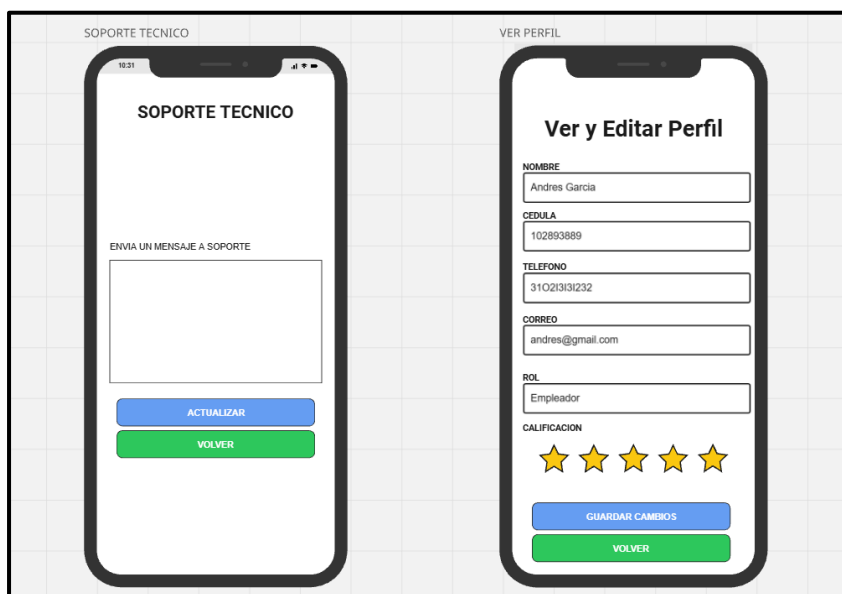
## Rol empleador pantallas



**Nota.** Elaboración propia.

Figura 7

## Ilustración 7 Soporte



**Nota.** Elaboración propia.

Este conjunto de pantallas muestra toda la experiencia diseñada para el perfil Empleador dentro de la aplicación móvil Conectando. Esta plataforma está pensada para facilitar a las empresas la contratación de talento por horas, permitiendo también la interacción con estudiantes, mentores y administradores, y creando así un ecosistema laboral y de desarrollo activo.

El recorrido del empleador comienza en la pantalla Gestión de Postulaciones, que funciona como su panel principal. Desde ahí puede realizar acciones clave como:

- Crear nuevas postulaciones para publicar ofertas laborales
- Ver la ubicación de los talentos o los lugares de trabajo
- Gestionar los pagos por cada contratación
- Editar o eliminar ofertas existentes
- Acceder a soporte técnico
- Ver y actualizar su perfil

Al momento de crear una nueva postulación, se accede a la pantalla Publicar oferta de trabajo, donde el empleador debe ingresar los detalles de la vacante: el puesto, una descripción clara, la ubicación, los requisitos, el horario (días y horas), el número de horas a trabajar y el valor por hora. Una vez llenados todos los campos, puede publicar la oferta fácilmente.

En la sección Lista de Postulaciones, puede consultar todas las ofertas publicadas. Para cada una se muestra información como el cargo, la ubicación, el horario, el nombre del postulante y el estado del proceso (pendiente, aceptada, finalizada o pagada). Esto permite hacer un seguimiento claro y organizado de cada contratación.

La pantalla Pagos Pendientes le muestra cuánto ha pagado en total y el detalle de cada pago por contratación, ayudándole a tener un control financiero dentro de la app.

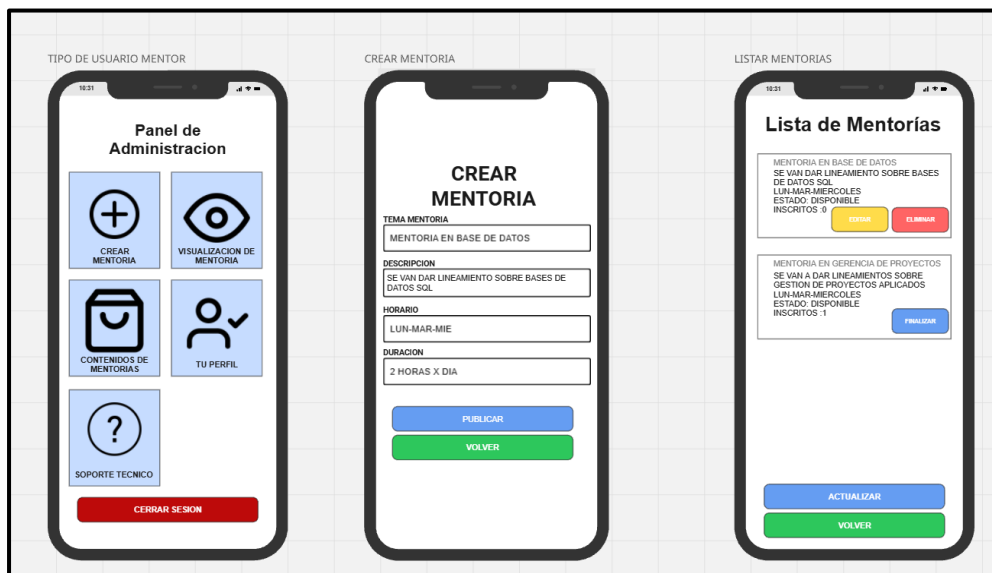
Si necesita actualizar una oferta, puede hacerlo desde Editar o Eliminar Postulaciones, buscando por ID y luego editando o eliminando la publicación. Al editarla, puede modificar todos los campos completados originalmente.

También cuenta con una sección de Soporte Técnico, desde donde puede comunicarse con el equipo de ayuda ante cualquier duda o problema. Y desde Ver y Editar Perfil, el empleador puede modificar su información personal como nombre, cédula, teléfono, correo, su rol como empleador y su calificación. Esta sección es igual para todos los perfiles (estudiante, mentor y administrador), lo que garantiza una experiencia uniforme.

### Vista Rol Mentor.

Figura 8

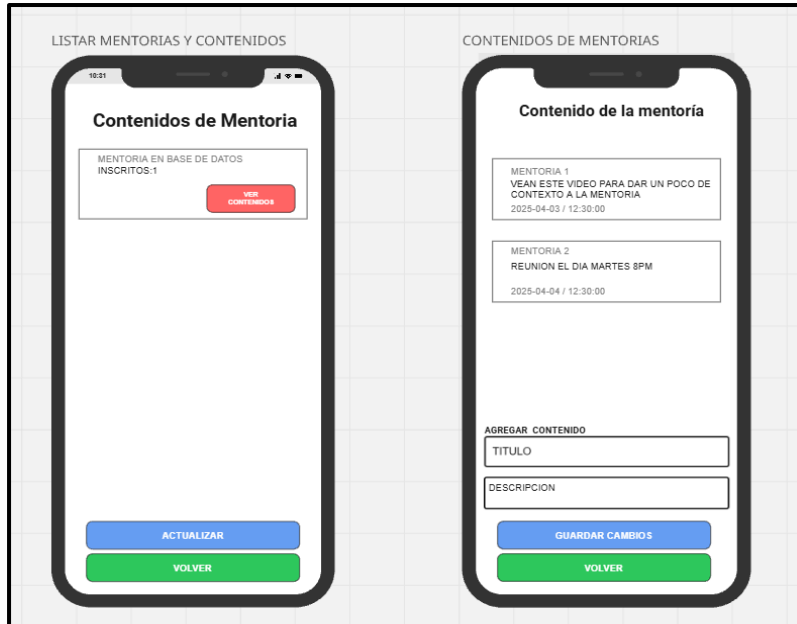
Rol Mentor



**Nota.** Elaboración propia.

Figura 9

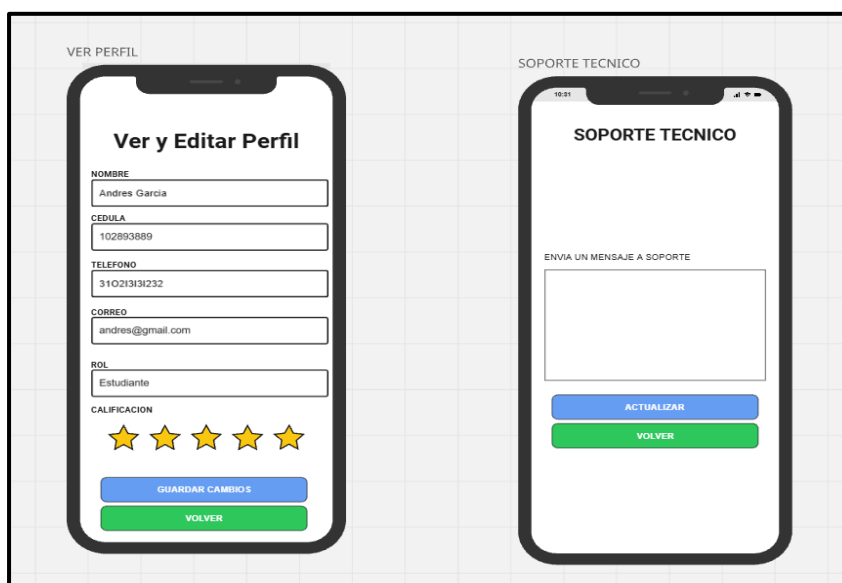
## Contenidos Mentoría



**Nota.** Elaboración propia.

Figura 10

## Soporte Mentorías



**Nota.** Elaboración propia.

Este conjunto de pantallas muestra toda la interfaz diseñada para el perfil Mentor dentro de la aplicación móvil Conectando. Esta plataforma permite que profesionales o expertos compartan su conocimiento y guíen a estudiantes, al tiempo que pueden interactuar con empresas y administradores, creando un ecosistema de aprendizaje y desarrollo integral.

El recorrido del mentor comienza en su panel de administración, que funciona como su centro de control. Desde allí puede realizar varias acciones importantes como:

- Crear nuevas mentorías
- Ver las mentorías que ya ha publicado
- Gestionar el contenido de cada sesión
- Editar su perfil
- Acceder al soporte técnico
- Cerrar sesión

Al elegir la opción de crear mentoría, el mentor accede a una pantalla donde puede ingresar todos los detalles de la sesión: el tipo de mentoría (por ejemplo, en bases de datos), una descripción clara, los días y horas en que se realizará, la duración total y la cuota que cobrará por hora o por sesión. Una vez completado todo, puede publicar la mentoría con un solo botón.

En la sección de mentorías publicadas, el mentor puede ver la lista de todas las sesiones que ha creado. Por cada una se muestra el título, la descripción, el horario, las horas de duración y la cantidad de estudiantes inscritos. También tiene botones para editar o cerrar la sesión, lo que le da control total sobre su oferta educativa.

Para complementar sus sesiones, el mentor puede acceder a la sección de contenidos de mentorías, donde puede ver y añadir materiales que acompañen el proceso de aprendizaje,

como videos, descripciones de actividades o enlaces útiles. Para agregar nuevo contenido, simplemente debe ingresar un título y una descripción, y guardarlo.

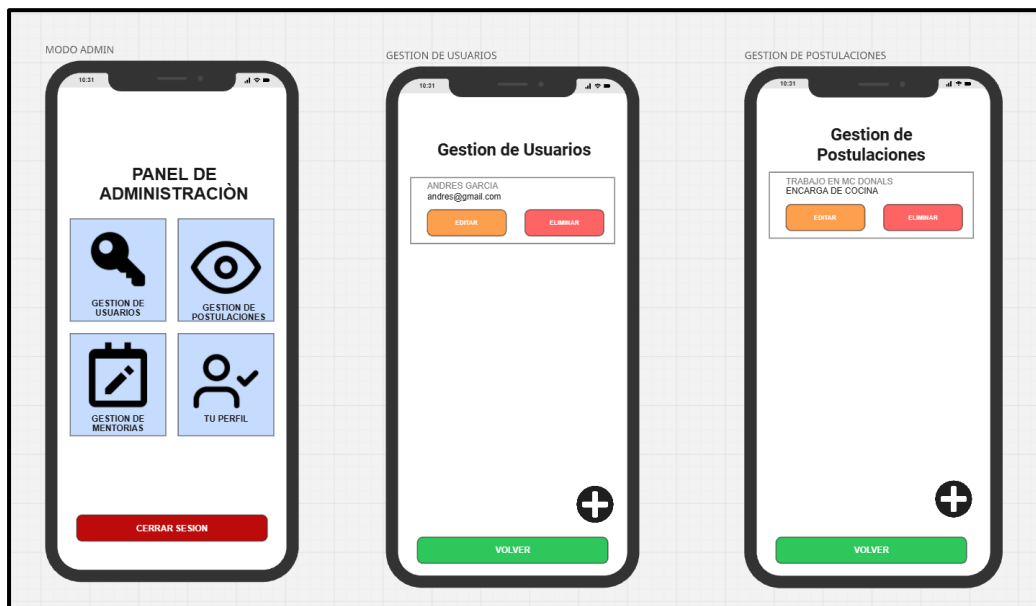
Además, el mentor puede entrar a la sección de soporte técnico si necesita ayuda, enviando un mensaje directamente al equipo de asistencia. También puede revisar o modificar su información personal en la pantalla de perfil, donde puede actualizar datos como nombre, cédula, teléfono, correo, su rol como mentor y ver su calificación dentro de la plataforma.

Con estas funciones, el mentor tiene todo lo que necesita para gestionar de forma sencilla y efectiva sus sesiones de mentoría, conectándose con estudiantes que buscan crecer personal y profesionalmente.

### Vista Rol Administrador.

*Figura 11*

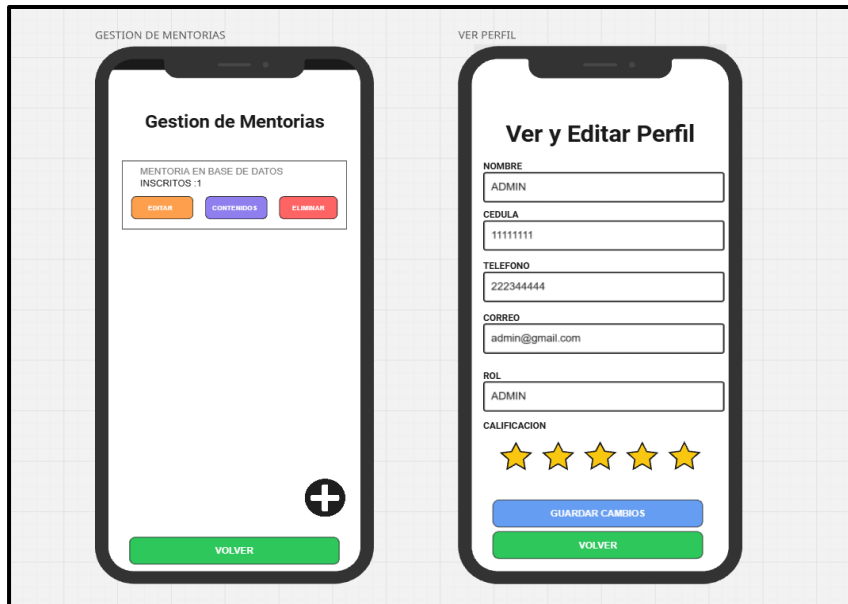
#### *Rol Administrador*



**Nota.** Elaboración propia.

Figura 12

## Gestión Mentorías



**Nota.** Elaboración propia.

Este conjunto de pantallas ilustra la experiencia diseñada para el perfil de Administrador dentro de la aplicación móvil Conectando, una plataforma integral que permite conectar estudiantes, empleadores y mentores en un ecosistema de aprendizaje y empleo por horas. El administrador es el encargado de supervisar y garantizar el buen funcionamiento del sistema, gestionando usuarios, ofertas laborales y mentorías.

El recorrido del administrador comienza en el Panel de Administración (también identificado como "Modo Admin"), que funciona como centro de control principal. Desde esta pantalla, puede acceder rápidamente a las funciones clave de gestión:

- Gestión de usuarios.
- Gestión de postulaciones.
- Gestión de mentorías.

- Ver y editar perfil.
- Cerrar sesión.

Al seleccionar Gestión de usuarios, el administrador accede a una interfaz donde puede visualizar a todos los usuarios registrados en Conectando, incluyendo estudiantes, empleadores y mentores. Por cada usuario se muestra su nombre y correo electrónico, con opciones para Editar o Eliminar su perfil. También se incluye un botón con el símbolo +, que indica la posibilidad de registrar nuevos usuarios de manera manual, otorgando control completo sobre la base de datos de usuarios de la plataforma.

En la sección de Gestión de postulaciones, el administrador puede supervisar todas las ofertas de trabajo publicadas. Se listan postulaciones con detalles como el cargo (por ejemplo, "Trabajo en McDonald's - Encargado de Cocina") y se ofrecen botones para Editar o Eliminar cada una, asegurando que todas las vacantes publicadas cumplan con los lineamientos de calidad y pertinencia. Al igual que en la gestión de usuarios, también hay una opción para añadir nuevas postulaciones.

La pantalla de Gestión de mentorías permite al administrador tener visibilidad sobre las sesiones educativas ofrecidas en la plataforma. Cada mentoría (como "Mentoría en Base de Datos") se presenta con opciones para Editar o Eliminar la información, lo que facilita mantener actualizados y relevantes los contenidos de aprendizaje. Esta funcionalidad permite garantizar una experiencia de calidad tanto para mentores como para estudiantes.

Al igual que otros perfiles, el administrador puede ingresar a la sección Ver y editar perfil, donde puede consultar y modificar sus datos personales: nombre, cédula, teléfono, correo electrónico y rol, que se identifica claramente como ADMIN. También aparece una

calificación por estrellas, que puede formar parte de un sistema interno de evaluación o ser un elemento compartido en el diseño general de la app.

### 11.5 Implementación.

*Tabla 13 Implementación*

Componente de software	Arquitectura	Funcionalidad
Rutas para vistas (registeractivity.kt)(loginactivity.kt)  (mainactivity.kt),etc.	Frontend	Controla la navegación entre pantallas (login, registro, perfil, ofertas, etc.).
Formulario de Registro/Login	Frontend	Captura datos del usuario y los envía a Firebase Authentication para validación.
Firebase Authentication	Backend	Autentica usuarios y gestiona sesiones por rol (estudiante, mentor, empleador).
Firebase Firestore	Base de datos	Almacena y gestiona información: usuarios, ofertas, mentorías y postulaciones.
Firebase Storage	Backend	Guarda archivos como hojas de vida, información asociada al perfil , etc.
Cloud Firestore Rules	Seguridad	Restringe el acceso a datos según rol y autenticación del usuario.
Cloud Functions	Backend	Realiza validaciones con Auth, envío de correos o notificaciones de credenciales.
Pantalla de Perfil (registeractivity.kt)(loginactivity.kt)  (mainactivity.kt , etc	Frontend	Muestra y permite editar la información personal del usuario.
Módulo de Publicación de Ofertas	Frontend	Permite a empleadores crear ofertas y a estudiantes postularse.
Módulo de Solicitud de Mentoría	Frontend	Permite a estudiantes solicitar mentorías y contactar mentores disponibles.
Panel de administración	Frontend / Firebase	Visualiza y gestiona usuarios, contenido y métricas por parte del administrador.

**Nota.** Elaboración propia.

## **11.7 Pruebas y QA de Software.**

### **Modelo propuesto para el aseguramiento de la calidad del software.**

Para asegurar la estabilidad, funcionalidad y experiencia del usuario en la aplicación Conectando, creada en Android Studio e integrada a Firebase Firestore, se estableció un modelo de garantía de calidad fundamentado en los siguientes elementos.

#### **1. Estrategia de Pruebas**

Se estableció un método de evaluación combinado, centrado tanto en la verificación técnica como en la experiencia del usuario final:

##### **Pruebas instrumentadas (Espresso)**

Se utilizaron para simular interacciones reales del usuario con la interfaz. Esto permitió verificar el comportamiento de actividades como el registro de usuarios, inicio de sesión, creación y edición de ofertas o mentorías dando como resultado un funcionamiento correcto.

##### **Pruebas de integración**

Se verificó la correcta interacción entre los módulos de la app y los servicios de Firebase (Firestore, Auth, Storage), asegurando que la información se guarde y se visualice adecuadamente sin afectar el uso de esta.

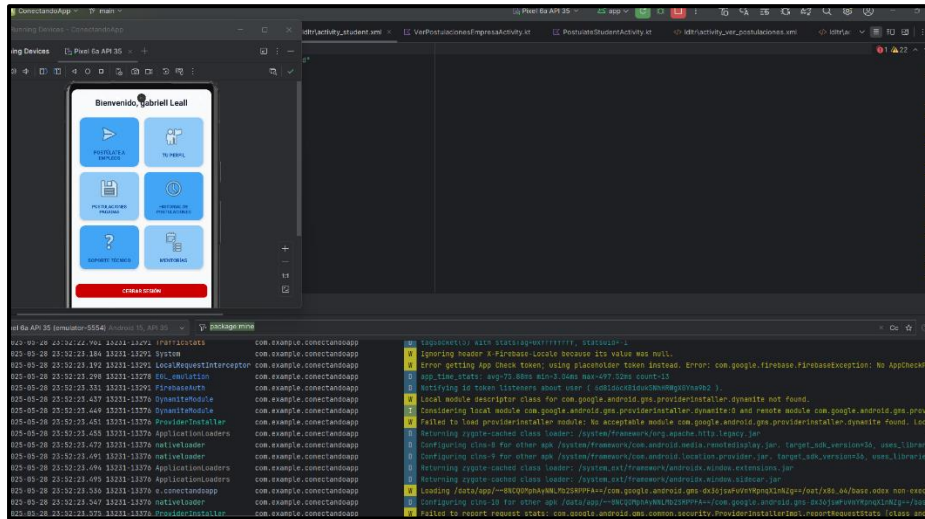
##### **Evidencia.**

Se realiza el inicio de sesión al perfil estudiante donde podemos ver que el usuario fue encontrado por el proceso de Firebase Auth para adicionalmente, ir a su sección de usuarios y

darle la instrucción de acceder al tipo de pantalla que corresponde al rol de estudiante mostrando su información.

Figura 13

Evidencia Prueba.



Nota. Elaboración propia.

## Pruebas funcionales

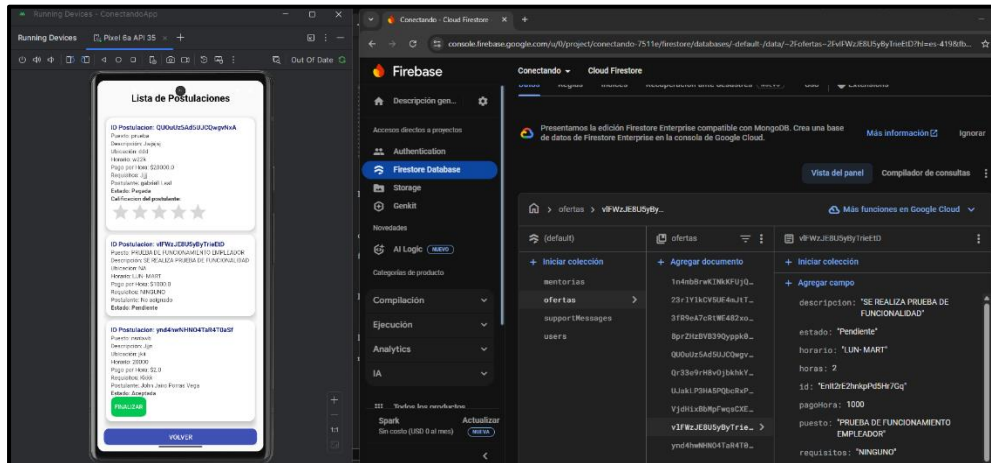
Se evaluaron los flujos clave del sistema: creación de ofertas, inscripción a mentorías, envío de mensajes, y gestión de roles. Todas las pruebas se realizaron usando dispositivos físicos y emuladores, así como también se realizó la ejecución de smoke test.

## Evidencia.

Como podemos ver el funcionamiento para esta prueba fue realizado para el rol de empleador, en el cual se creó una oferta de prueba donde evidenciamos la correcta llegada de los datos y la posterior visualización en la pantalla cumpliendo una de las diferentes pruebas realizadas para la validación del aplicativo en su parte funcional.

Figura 14

Evidencia prueba.



**Nota.** Elaboración propia.

## Pruebas de usabilidad

Se realizaron pruebas con usuarios reales para evaluar la navegación, comprensión de la interfaz, tiempos de carga y mensajes de error. Se recogieron sugerencias y se aplicaron mejoras en la interfaz.

## 2. Herramientas Utilizadas

- **Android Studio** – Entorno principal de desarrollo y ejecución de pruebas.
- **Espresso** – Herramienta de pruebas UI instrumentadas en Android.
- **Emuladores Android** – Para pruebas en diferentes versiones y resoluciones de pantalla.
- **Dispositivos físicos** – Para pruebas de rendimiento y conectividad real.

### **3. Resultados del Proceso de QA.**

Gracias a este proceso, fue posible identificar errores con antelación y corregirlos antes de que afectaran a los usuarios. También se mejoró el rendimiento general de la aplicación, lo que resultó en una experiencia más fluida y estable.

## **12. Conclusiones.**

La ejecución de este proyecto representa el resultado de un trabajo en equipo , comprometido que no habría sido posible sin la constancia, responsabilidad, y el compromiso de cada uno de los integrantes. Esta colaboración permitió afrontar las diferentes situaciones que se presentaron con el paso del tiempo facilitando el aprendizaje continuo a nivel personal , profesional y académico.

Asimismo , la creación de un Producto Mínimo Viable (MVP) funcional fue posible mediante la utilización de Android Studio y Firebase como las herramientas principales de desarrollo e implementación desde el Backend hasta el Front End. Lo cual permitió validar técnica y conceptualmente la propuesta planteada luego de un proceso arduo de implementación y desarrollo, demostrando con ello su viabilidad en torno a la proyección de lo que podría ser en un futuro una solución escalable, segura y centrada en las necesidades reales de algunos estudiantes universitarios.

La implementación de sistemas de autenticación y gestión de roles a través de Firebase Authentication, junto con la aplicación de reglas de seguridad en Firestore, garantizó la protección de los datos personales y la integridad del sistema, lo cual es fundamental para generar confianza en los usuarios. Así como también la utilización de Firestore para el almacenamiento y la preservación de la integridad de los datos.

De igual manera , la adopción de principios de metodologías ágiles contribuyó significativamente a la organización del trabajo. Teniendo en cuenta que la segmentación del desarrollo en fases facilitó la gestión del tiempo a pesar de algunos contratiempos que se presentaron y permitió entregar funcionalidades clave de manera eficiente y oportuna.

La estrategia propuesta por la aplicación, orientada a conectar a los estudiantes universitarios con empleos por horas y oportunidades de autoconocimiento asociados a la capacidad de poder obtener una mentoría en la plataforma, representa una iniciativa innovadora con un potencial transformador a nivel social, ya que responde a una de las problemáticas más relevantes de la juventud colombiana 'el desempleo'.

Finalmente, la documentación técnica y académica desarrollada durante el proyecto evidencia el conocimiento adquirido por el equipo, y asegura que el MVP cumpla con los objetivos inicialmente planteados, con ansias de una futura evolución y posible implementación a mayor escala.

### 13. Bibliografía.

- Carrasco Ruano, Y. T. (2018). Los estudiantes universitarios y el aspecto socio económico. *Conciencia Digital*, 1(2), 6–15. <https://doi.org/10.33262/concienciadigital.v1i2.946>
- Departamento Administrativo Nacional de Estadística. (2023). *Boletín técnico: Mercado laboral de la juventud*. <https://www.dane.gov.co/index.php/estadisticas-por-tema/mercado-laboral/mercado-laboral-de-la-juventud>
- Rodríguez, L. (2023, November 16). *El 30% de los jóvenes en Colombia deja la universidad por dificultades económicas*.
- Aguilar, & Ramón Alfredo. (2021). *El trabajo a través de plataformas digitales: ¿autónomos o subordinados?. Una visión de derecho comparado*. <https://elibro-net.bdbiblioteca.universidadean.edu.co/es/lc/bibliotecaeaan/titulos/249321>
- Banco Mundial. (2023). *Tasa de desempleo, jóvenes (15-24 años) (% de la población activa)*. <https://datos.bancomundial.org/>
- Berard, V. (2022). *Las 10 mejores aplicaciones de empleo en Australia*. <https://grabjobs.co/>
- Departamento Administrativo Nacional de Estadística. (2024, August 12). *Mercado laboral de la Juventud Trimestre abril - junio 2024*. <https://www.dane.gov.co/files/operaciones/GEIH/bol-GEIHMLJ-abr-jun2024.pdf>
- OIT, & Marcel Crozet. (2025, February 24). *Jóvenes en América Latina y el Caribe: entre la informalidad y la falta de oportunidades*. <https://news.un.org/es/story/2025/02/1536731>

Amazon Web Services. (n.d.-a). *¿En qué consiste Scrum?* Retrieved March 8, 2025, from <https://aws.amazon.com/es/what-is/scrum/>

Amazon Web Services. (n.d.-b). *¿Qué es el ciclo de vida del desarrollo de software (SDLC)?* Retrieved March 8, 2025, from <https://aws.amazon.com/es/what-is/sdlc/>

Carolina César Pipenburg. (2024, June 3). *¿Qué modelo de ciclo de vida del software es el más adecuado para tu proyecto?* <https://intelequia.com/es/blog/post/qu%C3%A9-modelo-de-ciclo-de-vida-del-software-es-el-m%C3%A1s-adecuado-para-tu-proyecto>

Claire Drumond. (n.d.). *Qué es scrum y cómo empezar.* Retrieved March 8, 2025, from <https://www.atlassian.com/es/agile/scrum>

Edgar. (2022, January 15). *¿Qué es el ciclo de vida de desarrollo de sistemas/software?* <https://www.sergestec.com/que-es-el-ciclo-de-vida-de-desarrollo-de-sistemas-software/>