

**FACULTAD DE INGENIERÍA Y ARQUITECTURA**

**ESCUELA DE INGENIERÍA DE SISTEMAS**

Aplicación Móvil "Estoy Contigo" plataforma Integral de Salud Mental para universitarios de la UCV en Ate

**Autores**

Arteaga Paniagua, Javier (ORCID: 0000-0003-0028-9695)

Cotrina Enrique, Victor Ramiro (ORCID: 0000-0003-0673-5115)

De La Cruz De La Cruz, Alex Rodolfo (ORCID:0000-0003-3148-5760)

Guerrero Carbajal, Roció del Pilar (ORCID: 0000-0002-9928-0975)

Reyes Diaz, Moisés Alejandro (ORCID: 0000-0003-3925-4190)

Ruidiaz Bravo, Luis Andree (ORCID: 0000-0002-8740-3237)

Sparrow Falcon, Jhon Henry (ORCID: 0000-0002-8740-3237)

Mora Palomino, Alessandro Javier (ORCID:0004-0008-0924-6789)

**Curso**

PROGRAMACIÓN DE APLICACIONES MÓVILES

**Asesor**

MENDOZA CANALES AURON EDUARDO

**LIMA - PERÚ**

**2025**

**ÍNDICE**

[**I.** **INTRODUCCIÓN** 3](#_Toc203006075)

[CONTEXTO ACTUAL 4](#_Toc203006076)

[PROBLEMÁTICA 5](#_Toc203006077)

[JUSTIFICACIÓN 6](#_Toc203006078)

[ANTECEDENTES INTERNACIONALES 8](#_Toc203006079)

[ANTECEDENTES NACIONALES 10](#_Toc203006080)

[**II.** **MARCO TEÓRICO** 12](#_Toc203006081)

[2.1 Salud emocional en estudiantes universitarios 12](#_Toc203006082)

[2.2 Tecnología móvil como herramienta para la salud mental 12](#_Toc203006083)

[2.3 Sistemas Operativos para dispositivos móviles 13](#_Toc203006084)

[2.4 Entorno de desarrollo Android Studio 14](#_Toc203006085)

[2.5 Gestor de Base de Datos 15](#_Toc203006086)

[2.6 Lenguajes de programación 18](#_Toc203006087)

[2.7 Metodología de desarrollo de software 19](#_Toc203006088)

[**III.** **DESARROLLO** 22](#_Toc203006089)

[3.1 Análisis de los requerimientos 22](#_Toc203006090)

[3.2. Diagramas de desarrollo 24](#_Toc203006091)

[3.6 Diseño 36](#_Toc203006092)

[3.7 Desarrollo de la Base de Datos 40](#_Toc203006093)

[3.8 Desarrollo de la interfaz 42](#_Toc203006094)

[3.9 App “EstoyContigo” en producción 57](#_Toc203006095)

[**INTERPRETACIÓN** 59](#_Toc203006096)

[**COMPARACIONES** 61](#_Toc203006097)

[**CONCLUSIONES** 62](#_Toc203006098)

[**REFERENCIAS BIBLIOGRÁFICAS** 63](#_Toc203006099)

# **INTRODUCCIÓN**

En los últimos años, la salud mental de los estudiantes universitarios ha empeorado significativamente, con altos niveles de estrés, ansiedad y depresión. Según la Organización Mundial de la Salud (OMS, 2024), uno de cada siete jóvenes de entre 10 y 19 años padece algún tipo de trastorno mental. Además, una publicación del Instituto Americano del Estrés (IAE, 2025) detalla que el 70% de los adolescentes identifican la ansiedad o la depresión como problemas importantes entre sus compañeros.

Es preciso resaltar que la salud mental de los estudiantes universitarios es una problemática cada vez más visible, debido a la presión académica, la sobrecarga de responsabilidades y los cambios emocionales que se presentan durante esta etapa; la soledad, el estrés y la ansiedad son algunos de los problemas más frecuentes, pero también los menos atendidos por barreras como el estigma social o la falta de acceso a atención psicológica, ante esta problemática, surge "Estoy Contigo", una aplicación móvil integral diseñada para brindar apoyo psicológico, herramientas de autocuidado y acceso a profesionales de la salud mental. Esta plataforma busca reducir las barreras de atención, fomentar el bienestar emocional y prevenir crisis en la población estudiantil.

En un contexto donde la demanda de servicios psicológicos supera la oferta en muchas universidades, la aplicación "Estoy Contigo" se posiciona como una alternativa innovadora como una herramienta digital pensada para brindar apoyo emocional diario a estudiantes universitarios a través del registro de estado de ánimo, frases motivadoras y una interacción inicial con inteligencia artificial empática para acompañar a los estudiantes en su bienestar emocional.

## CONTEXTO ACTUAL

A nivel mundial, la salud mental de los estudiantes universitarios se ha convertido en un desafío prioritario en las políticas públicas de educación. La Organización Mundial de la Salud (OMS, 2024) advierte que los trastornos mentales representan la principal causa de discapacidad entre jóvenes de 18 a 25 años, con la depresión y la ansiedad como los diagnósticos más frecuentes. En Sudáfrica, entre el 21 y el 24,5% de los estudiantes padecen trastornos de salud mental como ansiedad, estrés postraumático, trastornos alimentarios, déficit de atención e hiperactividad (Bantjes et al., 2023).

En América Latina, la situación es igualmente alarmante en Chile, el 45% de los estudiantes presentan síntomas de ansiedad, el 46% de depresión y el 53% de estrés (Barrera-Hererra y San Martín, 2021). En nuestro país el panorama se agrava por condiciones estructurales, según el Ministerio de Educación (2019) los casos más recurrentes en la comunidad universitaria son ansiedad con 82 %, estrés con 79 % y violencia con 52% siendo una cifra superior al promedio latinoamericano.

Mientras el mundo avanza hacia modelos de salud mental digital para llegar a más jóvenes, en Perú persiste una desatención crítica que requiere soluciones inmediatas. En este escenario, la aplicación "Estoy Contigo" se posiciona como una herramienta innovadora, que combina recursos tecnológicos con evidencia científica para reducir brechas y salvar vidas.

## PROBLEMÁTICA

La salud mental de los estudiantes universitarios en Perú enfrenta una problemática multifactorial y crítica, caracterizada por altos niveles de estrés, ansiedad, depresión y una alarmante falta de acceso a servicios psicológicos adecuados. Esta situación no solo afecta el rendimiento académico, sino también la calidad de vida, las relaciones interpersonales y, en casos extremos, conduce a desenlaces trágicos como el suicidio. A pesar de la creciente visibilidad del tema, las respuestas institucionales siguen siendo insuficientes, dejando a miles de jóvenes sin apoyo oportuno.

El acceso limitado a servicios de salud mental, junto con la creciente incidencia de problemas emocionales en jóvenes universitarios, crea una brecha preocupante, ya que en la actualidad la mayoría de aplicaciones existentes presentan barreras significativas como estar disponibles en idiomas extranjeros, implementar modelos de suscripción poco accesibles para estudiantes y, lo más crítico, carecen de adaptación al contexto socioemocional y las particularidades culturales de los jóvenes latinoamericanos. Así como lo señala Gómez-Gonzales et al. (2022), la mayoría de aplicaciones móviles enfocadas en salud mental carecen de un enfoque personalizado y emocionalmente conectado con el usuario, lo que reduce su impacto en la vida real.

Esta desconexión entre las necesidades reales de los universitarios y las soluciones disponibles deja a miles de estudiantes peruanos sin herramientas adecuadas para manejar su bienestar psicológico. Frente a esta realidad, resulta evidente que las soluciones tradicionales son insuficientes para atender la magnitud del problema. Se requieren enfoques innovadores que superen las barreras geográficas, económicas y sociales, aprovechando herramientas tecnológicas para llegar a más estudiantes de manera oportuna. Es por ello que la aplicación móvil "Estoy Contigo" surge como alternativas prometedoras al ofrecer servicios accesibles de terapia en línea, recursos de autocuidado y comunidades de apoyo, complementando así los esfuerzos institucionales y contribuyendo a reducir la brecha en atención de salud mental para la población universitaria peruana.

## JUSTIFICACIÓN

La salud mental en los estudiantes universitarios se ha deteriorado aceleradamente en los últimos años, convirtiéndose en una crisis global con graves repercusiones en el ámbito académico, social y personal. En Perú, esta problemática se agrava ya que solo seis universidades cuentan con Centros de Salud Mental Comunitarios, enfrentando graves limitaciones de personal y recursos, mientras los suicidios en jóvenes alcanzan uno o dos casos al día según un reportaje de La República (2025). Además, en 2024 se registraron 756 casos de depresión grave entre jóvenes de 18 a 29 años, principalmente estudiantes según Correo (2025). Ante este escenario, la aplicación “Estoy Contigo” se justifica como respuesta urgente, el cual ofrece apoyo emocional accesible, gratuito y adaptado al entorno universitario.

Así mismo, como experiencia internacional se confirma la eficacia de plataformas de salud mental digital, por ejemplo, en España, la app “Quiero cuidarme Más” ofrece atención psiquiátrica por chat y herramientas preventivas, logrando alta captación sin coste adicional (B.C. et al., 2025). Es por ello que de acuerdo a los casos que se presentan en nuestro país, la App “Estoy Contigo” se adaptará específicamente al contexto socioemocional y cultural de los universitarios peruanos, ofreciendo contenidos locales, lenguaje inclusivo y un acompañamiento empático basado en evidencia científica.

El desarrollo de la aplicación móvil “Estoy contigo” nace de la necesidad urgente de brindar soporte emocional accesible y oportuno a estudiantes universitarios, un grupo particularmente vulnerable frente a los problemas de salud mental. Diversos estudios evidencian que, durante la etapa universitaria, muchos jóvenes experimentan altos niveles de estrés, ansiedad, depresión e incluso pensamientos suicidas, debido a la presión académica, problemas familiares, económicos o la falta de redes de apoyo efectivas (Organización Mundial de la Salud [OMS], 2022).

En ese sentido, la salud emocional se ha convertido en un pilar fundamental para el rendimiento académico y el bienestar integral. Sin embargo, el acceso a servicios psicológicos profesionales aún es limitado en muchas instituciones, ya sea por falta de presupuesto, estigma social o saturación de los servicios existentes. Ante esta problemática, las herramientas digitales emergen como alternativas viables para ofrecer un primer nivel de atención emocional, que, si bien no sustituyen la terapia clínica, sí pueden servir como mecanismos de detección temprana y orientación inicial (Muñoz et al., 2021).

La app “Estoy contigo” busca facilitar el registro diario de emociones, brindar mensajes de acompañamiento positivo y sugerir recursos de autocuidado basados en principios de la psicología positiva y el enfoque cognitivo-conductual. Según Alvarado Loaiza et al. (2024), las plataformas digitales que promueven la auto observación emocional y la retroalimentación empática pueden reducir significativamente los niveles de ansiedad y fortalecer la percepción de bienestar subjetivo. Asimismo, Franco-Carrero et al. (2023) destacan que las aplicaciones que integran componentes motivacionales y humanizados tienen efectos protectores frente a la sintomatología depresiva en contextos educativos.

Además, estudios recientes sobre salud digital y e-health señalan que los estudiantes están más dispuestos a interactuar con tecnologías móviles que les permitan expresarse de forma segura y anónima (Valencia-Ortiz & Hernández, 2022). Esto resulta clave para disminuir barreras de acceso y normalizar el autocuidado emocional en espacios donde aún persiste el estigma hacia la salud mental.

Por tanto, esta propuesta no solo responde a una necesidad emocional, sino también a una necesidad de salud pública, ya que contribuye a la promoción de la salud mental y la prevención de trastornos psicológicos mediante el uso de herramientas tecnológicas accesibles, empáticas y basadas en evidencia.

## ANTECEDENTES INTERNACIONALES

A nivel internacional, se han desarrollado diversas investigaciones orientadas a evaluar el impacto de las aplicaciones móviles en la salud mental. Un ejemplo destacado es el estudio de; Linardon, Torous, Firth, Cuijpers, Messer y Fuller-Tyszkiewicz (2024), en su estudio titulado "Current evidence on the efficacy of mental health smartphone apps for symptoms of depression and anxiety. A meta-analysis of 176 randomized controlled trials", llevaron a cabo un extenso meta-análisis con el objetivo de evaluar la eficacia de las aplicaciones móviles para la salud mental en la reducción de síntomas de depresión y ansiedad. Esta investigación analizó los resultados de 176 ensayos clínicos aleatorizados (RCTs) publicados, en los cuales se incluyó una muestra combinada de más de 33,000 participantes en estudios sobre depresión y más de 22,000 en estudios sobre ansiedad. Las poblaciones estudiadas fueron diversas, incluyendo adultos jóvenes, estudiantes universitarios y población general. El análisis se realizó mediante modelos de efectos aleatorios, calculando el tamaño del efecto estandarizado con la g de Hedges, y se evaluó la heterogeneidad entre estudios utilizando el estadístico I², una medida que cuantifica el porcentaje de variación entre los estudios atribuible a la heterogeneidad real y no al azar, lo que refuerza la robustez del análisis. Además, se aplicaron análisis de subgrupos para identificar factores que pudieran influir en los resultados, como el tipo de app utilizada o el enfoque terapéutico. Los hallazgos indicaron efectos pequeños pero estadísticamente significativos en la reducción de síntomas de depresión y ansiedad generalizada, así como efectos moderados en síntomas de ansiedad social y obsesivo-compulsivos. Las aplicaciones que incluían elementos de terapia cognitivo-conductual (TCC), monitoreo del estado de ánimo o tecnología de chatbot mostraron resultados más prometedores, consolidando así el valor terapéutico de estas herramientas digitales como apoyo complementario en salud mental.

Por su parte, Linardon, Cuijpers, Carlbring, Messer y Fuller-Tyszkiewicz (2019), en su artículo titulado "The efficacy of app-supported smartphone interventions for mental health: A meta-analysis of randomized controlled trials", llevaron a cabo un meta-análisis con el objetivo de determinar la eficacia de las intervenciones psicológicas mediante aplicaciones móviles en el tratamiento de problemas de salud mental. El estudio fue realizado por investigadores afiliados a universidades de Australia, Países Bajos y Suecia, quienes recopilaron y analizaron los resultados de 66 ensayos clínicos aleatorizados (RCTs), los cuales involucraron a miles de participantes de diferentes edades y contextos. Como técnica principal se aplicó el análisis estadístico cuantitativo mediante modelos de efectos aleatorios, con el fin de evaluar el impacto de estas intervenciones sobre variables como estrés, ansiedad, depresión y autoestima. Los resultados mostraron mejoras moderadas y estadísticamente significativas en el bienestar emocional de los usuarios que utilizan aplicaciones móviles con enfoque terapéutico, consolidando así la evidencia del potencial de estas herramientas digitales como recursos complementarios en la atención psicológica.

Por otro lado, Bress, Falk, Schier, Jaywant, Moroney, Dargis, Bennett, Scult, Volpp, Asch, Balachandran, Perlis, Lee y Gunning (2024), en su estudio titulado "Efficacy of a Mobile App-Based Intervention for Young Adults With Anxiety Disorders: A Randomized Clinical Trial", evaluaron la eficacia de una intervención basada en una aplicación móvil de terapia cognitivo-conductual (TCC) autoguiada, orientada a jóvenes adultos diagnosticados con trastornos de ansiedad. La investigación se desarrolló a través de un ensayo clínico aleatorizado con una duración de seis semanas, en el que participaron 59 personas entre 18 y 25 años, reclutadas mediante anuncios en línea y centros de salud mental ambulatorios. Los participantes fueron distribuidos en tres condiciones distintas de incentivo: una basada en la ganancia de puntos por completar sesiones (gain-framed), otra en la pérdida de puntos si no se completaban (loss-framed), y una tercera que combinaba el incentivo con apoyo social (gain–social support), permitiendo que un contacto cercano recibiera actualizaciones del progreso. Para medir el impacto del tratamiento, se aplicaron instrumentos clínicos como la Escala de Ansiedad de Hamilton (HAM-A), el Índice de Sensibilidad a la Ansiedad y la Escala de Ansiedad Social de Liebowitz. Los resultados evidenciaron una disminución significativa en los síntomas de ansiedad en todos los grupos, sin diferencias marcadas entre las condiciones, lo que respalda la efectividad general del uso de esta aplicación como apoyo terapéutico accesible para jóvenes adultos con trastornos de ansiedad.

## ANTECEDENTES NACIONALES

En el contexto nacional, Haji Taira (2020), en su tesis titulada "Diseño de aplicativo móvil para evitar el deterioro de la salud mental por el aislamiento social motivado por el Covid-19 en jóvenes de 20 a 25 años", desarrolló una investigación aplicada con enfoque cualitativo, orientada a diseñar una aplicación móvil que ayude a mitigar los efectos del aislamiento en la salud mental de jóvenes peruanos durante la pandemia. Empleó entrevistas semiestructuradas, análisis de necesidades y pruebas de usabilidad bajo el enfoque de diseño centrado en el usuario. Si bien no se detalla el tamaño de la muestra, la población objetivo fueron jóvenes entre 20 y 25 años afectados por el confinamiento. Como resultado, se obtuvo un prototipo funcional con herramientas como ejercicios de relajación, monitoreo del estado de ánimo y recursos informativos, contribuyendo así al cuidado psicológico en contextos de emergencia sanitaria.

Asimismo, Adauto Ortiz (2022), en su tesis titulada "Diseño e implementación de un aplicativo móvil de atención e información para una entidad pública prestadora de servicios de salud mental en personas con trastornos mentales de 18 años a 59 años en Lima, Perú", propuso una solución tecnológica para facilitar el acceso a información y atención en salud mental. Se trató de una investigación aplicada con enfoque cuantitativo, utilizando metodologías de ingeniería de software para el desarrollo y evaluación del aplicativo. A través de análisis de requisitos, diseño de interfaces, desarrollo de código y pruebas funcionales, se implementó una app que brinda orientación sobre trastornos como depresión, ansiedad y estrés, entre otros. Aunque la muestra específica no se detalla, se enfocó en personas adultas entre 18 y 59 años, y se concluyó que la herramienta puede complementar de manera efectiva los servicios tradicionales de salud mental.

En el contexto peruano, un estudio relevante es el desarrollado por Pacora Chávez y Bezada Sánchez (2023), que tuvo como objetivo principal crear una cápsula acústica y una aplicación móvil denominada *Moodmates* para fomentar la inclusión educativa de estudiantes universitarios con síntomas de depresión en Lima Metropolitana. Para el desarrollo del proyecto, se utilizó la metodología Toulouse Thinking, estructurada en cuatro fases: investigar, idear, desarrollar y transferir. La etapa de investigación incluyó encuestas, entrevistas y análisis DPESTA, centrado en jóvenes de entre 18 y 24 años. Se identificaron actores clave (estudiantes, personal universitario, familiares, entidades del Estado) y se aplicaron instrumentos con escala de Likert para medir la percepción sobre la salud mental en el entorno universitario. Además, se diseñó un prototipo de cabina acústica y se desarrolló un mockup funcional de la app.

Los resultados del proceso de validación fueron altamente positivos. En la primera validación, un 91.4% de los estudiantes expresó su interés en reservar una cita para utilizar el servicio. La segunda validación, centrada en feedback de usuarios finales, reveló un 100% de disposición a utilizarlo, destacando la utilidad percibida tanto de la cápsula como de la aplicación. Finalmente, la validación con expertos proporcionó respaldo técnico y económico, consolidando el potencial de implementación de la propuesta. Estos resultados respaldan la viabilidad y necesidad de soluciones tecnológicas enfocadas en la inclusión psicosocial en universidades peruanas.

## 

# **MARCO TEÓRICO**

## 2.1 Salud emocional en estudiantes universitarios

La salud emocional es un componente esencial del bienestar general, especialmente en la población universitaria, donde factores como el estrés académico, la presión familiar, la incertidumbre profesional y los cambios personales pueden desencadenar problemas como ansiedad, depresión y agotamiento. La Organización Mundial de la Salud (OMS) define la salud mental como “un estado de bienestar en el cual el individuo se da cuenta de sus propias aptitudes, puede afrontar las tensiones normales de la vida, puede trabajar de forma productiva y es capaz de contribuir a su comunidad” (OMS, 2022).

Diversas investigaciones destacan que la prevalencia de trastornos mentales en estudiantes ha aumentado significativamente. Según el Ministerio de Salud del Perú (2022), el 18% de los jóvenes de entre 18 y 29 años ha sido diagnosticado con algún problema de salud mental, y más del 50% no ha recibido atención profesional. Esta situación refleja una brecha crítica en el acceso a servicios psicológicos, lo que resalta la importancia de buscar estrategias complementarias, como el uso de herramientas tecnológicas accesibles.

## 2.2 Tecnología móvil como herramienta para la salud mental

En los últimos años, las aplicaciones móviles se han posicionado como alternativas viables para el monitoreo, la prevención y el acompañamiento en temas de salud mental. Estas herramientas permiten ofrecer un primer nivel de atención no clínica, facilitando el registro emocional, el acceso a información útil y la entrega de mensajes de apoyo.

Según Alvarado Loaiza et al. (2024), las aplicaciones móviles que permiten a los usuarios identificar y registrar sus emociones de forma cotidiana pueden reducir niveles de ansiedad y mejorar la percepción de bienestar subjetivo. Además, Franco-Carrero et al. (2023) señalan que las apps con un enfoque motivacional, empático y accesible pueden funcionar como factores protectores frente a la depresión, especialmente en contextos académicos.

## 2.3 Sistemas Operativos para dispositivos móviles

Los sistemas operativos móviles son componentes esenciales en la funcionalidad de los dispositivos inteligentes, encargados de gestionar eficientemente los recursos del sistema y las aplicaciones que en él operan. El mercado se caracteriza por la predominancia de tres plataformas principales: Android, iOS y, aunque en declive, el antiguo BlackBerry OS. La elección entre estas plataformas depende en gran medida de las necesidades individuales del usuario, considerando factores como la compatibilidad, la seguridad y el rendimiento (Bermúdez Moreno & López Hincapié, 2011).

Android: Desarrollado por Google, Android es un sistema operativo de código abierto, característica que ha contribuido a su amplia adopción por parte de diversos fabricantes. Su flexibilidad y la posibilidad de una alta personalización lo convierten en una opción atractiva para un amplio espectro de usuarios y dispositivos.

iOS: Exclusivo de Apple, iOS se distingue por ofrecer una experiencia de usuario fluida y altamente intuitiva. Se caracteriza por sus altos estándares de seguridad y la integración seamless entre los diferentes dispositivos del ecosistema Apple, creando una sinergia única para los usuarios comprometidos con la marca.

BlackBerry OS: Si bien actualmente en desuso, BlackBerry OS merece mención por su histórica importancia, especialmente en el ámbito empresarial. Su sistema de mensajería segura y su distintivo teclado físico lo posicionaron como una opción preferida para usuarios que priorizaban la privacidad y la eficiencia en la comunicación.

Otros Sistemas Operativos: Además de las tres plataformas principales, existen otros con menor cuota de mercado. HarmonyOS, desarrollado por Huawei, representa un intento de diversificación en el mercado; KaiOS, enfocado en dispositivos de gama baja, busca atender un nicho específico; mientras que sistemas como Windows Phone y versiones anteriores de BlackBerry OS han quedado obsoletos debido a la baja adopción.

**2.1.1 Sistema Operativo Android**

El sistema operativo Android, creado en 2007 y adquirido por Google, es uno de los sistemas más utilizados en dispositivos móviles como teléfonos inteligentes y tabletas. Su desarrollo se basa en el núcleo de Linux y permite una gran adaptabilidad y personalización para los usuarios y desarrolladores. Además, Android ofrece un entorno abierto para la instalación de aplicaciones, lo que lo ha convertido en la opción preferida en diversos entornos empresariales y personales (Sánchez Villafuerte, 2022, p. 27).

Ventajas:

* Código abierto que permite adaptaciones y personalización.
* Interfaz amigable y funcional, facilitando la navegación y el uso de los dispositivos.
* Uso libre de las aplicaciones de Google y compatibilidad con la nube.
* Versatilidad con accesorios y sistema de notificaciones bien organizado.
* Soporta la sincronización en la nube y la rápida transmisión de archivos a través de NFC.
* Permite la instalación de aplicaciones de diversas fuentes, ampliando la compatibilidad.

Desventajas:

* Requiere cerrar algunas aplicaciones para optimizar el consumo de batería.
* Puede tener dificultades de sincronización con otros equipos.
* Es vulnerable a ciberataques debido a su código abierto.
* Puede presentar calentamiento del dispositivo y alto consumo de batería y datos.

## 2.4 Entorno de desarrollo Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial de Google para el sistema operativo Android. Proporciona un conjunto completo de herramientas para diseñar, desarrollar y probar aplicaciones móviles. Gracias a su integración con recursos educativos y su uso extendido en la industria, Android Studio se ha convertido en una herramienta esencial para la formación de nuevos desarrolladores, especialmente en entornos educativos (Segura Canul, 2024).

Ventajas:

* Ofrece un entorno de desarrollo completo, permitiendo a los estudiantes conocer todo el ciclo de creación de aplicaciones, desde el diseño de interfaces hasta la gestión de bases de datos.
* Permite simular aplicaciones en distintos dispositivos y tamaños de pantalla, facilitando la creación de aplicaciones inclusivas y funcionales.
* Acceso a abundante documentación, tutoriales y foros, que fortalecen el aprendizaje autónomo y el pensamiento crítico.
* Prepara a los estudiantes con habilidades demandadas en el mercado laboral, como la programación en Java/ Kotlin y el diseño de UI/UX.

Desventajas:

* Su interfaz puede resultar intimidante para principiantes y requiere acompañamiento inicial para superar la curva de aprendizaje.
* Exige recursos significativos de hardware, lo que limita su uso en equipos con menos capacidad.

## 2.5 Gestor de Base de Datos

Un sistema de gestión de bases de datos (DBMS, por sus siglas en inglés) es una herramienta de software diseñada para facilitar el almacenamiento, la organización, el manejo y la recuperación eficiente de datos. Este tipo de sistema es fundamental en numerosos entornos informáticos y se emplea en una amplia gama de aplicaciones, desde soluciones simples hasta complejas plataformas empresariales.

Hay diversos tipos de sistemas de gestión de bases de datos, entre los cuales se destacan:

* **Relacionales (RDBMS)**: Organizan la información en tablas y utilizan SQL como lenguaje de consulta. Algunos ejemplos son MySQL, PostgreSQL y Microsoft SQL Server.
* **NoSQL:** Ofrecen mayor flexibilidad en la estructura de los datos, siendo adecuados para manejar grandes cantidades de información no estructurada. Ejemplos comunes incluyen MongoDB y Cassandra.
* **Bases de datos en memoria:** Diseñadas para ofrecer un rendimiento muy alto, almacenan los datos directamente en la memoria RAM. Un ejemplo representativo es Redis.
* **Bases de datos distribuidas:** Permiten gestionar grandes volúmenes de datos distribuidos en varios servidores. Un ejemplo destacado es Apache Cassandra.

Emplear un sistema de gestión de bases de datos (DBMS) brinda múltiples beneficios que mejoran significativamente la administración de la información. A continuación, se destacan algunos de los más relevantes:

* **Organización y estructura:** Facilita el almacenamiento ordenado de grandes cantidades de datos, lo que simplifica su acceso y manejo.
* **Seguridad y control de accesos:** Ofrece mecanismos de autenticación y asignación de permisos para proteger la información contra accesos no autorizados.
* **Rapidez en la recuperación de datos:** Mejora el rendimiento de las consultas, disminuyendo los tiempos de búsqueda y procesamiento.
* **Consistencia e integridad:** Asegura la exactitud y fiabilidad de los datos, reduciendo errores y duplicaciones.
* **Escalabilidad:** Se ajusta al crecimiento de los datos sin comprometer el rendimiento del sistema.
* **Acceso simultáneo y compatibilidad:** Permite que varios usuarios trabajen al mismo tiempo sin generar conflictos en la información.
* **Automatización de procesos:** Simplifica tareas como respaldos, recuperación y mantenimiento de datos de forma automática.

**2.3.1 Farebase**

Firebase es una plataforma basada en la nube que simplifica la creación de aplicaciones web y móviles, ofreciendo un conjunto de herramientas orientadas al desarrollo, escalado, monetización y análisis de datos.

Firebase sirve para:

* Crear apps de forma rápida y eficiente.
* Autenticar usuarios.
* Almacenar y sincronizar datos en tiempo real.
* Enviar notificaciones.
* Analizar el comportamiento de los usuarios.
* Escalar aplicaciones sin preocuparse por el backend.

**Ventajas de Firebase**

* Fácil integración con Android, iOS y web, base de datos en tiempo real, ideal para apps colaborativas o de mensajería.
* Escalabilidad automática, sin necesidad de administrar servidores.
* Alta seguridad con reglas personalizables.
* Análisis integrado, sin necesidad de herramientas externas.
* Sincronización offline, útil en apps que funcionan sin conexión.

## 2.6 Lenguajes de programación

A la hora de desarrollar apps móviles, existen múltiples lenguajes y frameworks, cada uno con sus características, ventajas y desventajas. La elección dependerá de la plataforma objetivo, el rendimiento deseado, la experiencia del equipo y otros factores del proyecto.

**2.4.1 Dart**

Es un lenguaje de programación altamente valorado en el entorno de desarrollo móvil por su rendimiento, su tipado fuerte y su capacidad para compilar código a aplicaciones nativas, lo cual resulta esencial para aplicaciones que El desarrollo de aplicaciones móviles modernas requiere lenguajes de programación eficientes y adaptables. Dart es un lenguaje creado por Google que permite construir aplicaciones rápidas y optimizadas para diversas plataformas, destacando por su sintaxis clara y su integración con el framework Flutter.

“Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development” (Dart, 2024).

Dart es altamente valorado en el entorno de desarrollo móvil por su rendimiento, su tipado fuerte y su capacidad para compilar código a aplicaciones nativas, lo cual resulta esencial para aplicaciones que requieren fluidez, como aquellas destinadas al acompañamiento emocional.

**2.4.2 Flutter**

Flutter es un framework de código abierto desarrollado también por Google que permite crear aplicaciones móviles, web y de escritorio desde una sola base de código. Utiliza Dart como lenguaje de programación y destaca por ofrecer widgets altamente personalizables, una excelente experiencia de usuario (UX) y un alto rendimiento en dispositivos Android e iOS.

De acuerdo con Martín-García & Gómez-Serrano (2022), Flutter es una herramienta ideal para proyectos que requieren agilidad, escalabilidad y eficiencia de recursos. Además, facilita la implementación rápida de prototipos y pruebas, lo que lo convierte en un aliado clave para proyectos centrados en el usuario, como es el caso de apps orientadas a la salud mental.

**2.4.3 Java**

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems en la década de 1990 y actualmente mantenido por Oracle. Es uno de los lenguajes más utilizados en el mundo para el desarrollo de aplicaciones móviles nativas en Android. Su estabilidad, seguridad y portabilidad lo convierten en una opción sólida para crear aplicaciones móviles con acceso directo al hardware del dispositivo y a los servicios del sistema operativo.

Java permite estructurar aplicaciones de forma modular y mantener una alta cohesión entre clases, lo cual facilita el mantenimiento y la escalabilidad de las aplicaciones, especialmente en proyectos de salud emocional donde el contenido debe actualizarse constantemente y adaptarse a diferentes necesidades del usuario.

“Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is intended to let application developers write once, run anywhere” (Oracle, 2023).

En el contexto del desarrollo móvil para la salud mental, Java permite una interacción robusta con bases de datos locales y remotas, integración con servicios como notificaciones push o almacenamiento seguro, y una amplia disponibilidad de bibliotecas para el manejo de interfaces gráficas, gráficos de seguimiento emocional, audio y más.

## 2.7 Metodología de desarrollo de software

En el entorno actual del desarrollo de aplicaciones, especialmente en el ámbito móvil, se han consolidado diversas metodologías que ofrecen marcos estructurados para organizar, planificar y ejecutar proyectos de software. Estas metodologías se eligen según las necesidades del proyecto, el tamaño del equipo, el presupuesto y el nivel de flexibilidad requerido.

Entre las metodologías tradicionales, destaca el modelo en cascada, caracterizado por su enfoque lineal y secuencial. Este tipo de metodología es adecuada cuando los requisitos del software están completamente definidos desde el inicio, aunque su rigidez limita la capacidad de adaptación ante cambios durante el desarrollo.

En contraste, las metodologías ágiles han ganado popularidad por su capacidad de adaptarse a entornos dinámicos. Estas se basan en ciclos iterativos e incrementales, en los que se entrega valor de manera continua. Entre las metodologías ágiles más reconocidas se encuentran:

Scrum, que organiza el trabajo en sprints con entregables funcionales al final de cada iteración. Promueve la colaboración entre roles bien definidos: Product Owner, Scrum Master y equipo de desarrollo.

Extreme Programming (XP), centrada en la calidad del software mediante prácticas como la programación en parejas, el desarrollo guiado por pruebas y la integración continua.

Kanban, que utiliza tableros visuales para gestionar el flujo de tareas, permitiendo una visión clara del estado del proyecto y facilitando la mejora continua.

Asimismo, existen metodologías híbridas, que combinan elementos de enfoques tradicionales y ágiles, lo que permite mantener una planificación estructurada al tiempo que se adoptan prácticas flexibles. Estas son especialmente útiles en proyectos que requieren cierto grado de previsibilidad, sin sacrificar la capacidad de adaptación.

En el ámbito específico del desarrollo móvil, se destaca la metodología Mobile-D, la cual adapta principios ágiles a las características particulares de este tipo de aplicaciones. Su estructura incluye cinco fases: exploración, inicialización, producción, estabilización y lanzamiento. Mobile-D enfatiza la colaboración, la entrega rápida de prototipos funcionales y la interacción constante con el cliente.

Otras metodologías relevantes incluyen Rational Unified Process (RUP), de enfoque iterativo y adaptable, y enfoques centrados en el usuario como Design Thinking y Lean UX, que complementan los procesos técnicos con una atención especial al diseño y a la experiencia del usuario final.

La elección de la metodología adecuada depende de múltiples factores, por lo que muchas organizaciones optan por combinar distintas prácticas para aprovechar las ventajas de cada enfoque (Pedreira-Souto, 2021).

**2.5.1 Metodología Ágil**

La metodología ágil es un enfoque de gestión de proyectos que prioriza la flexibilidad, la colaboración y la entrega continua de valor. Se originó en 2001 con el Manifiesto Ágil, que establece cuatro valores fundamentales:

* Individuos e interacciones sobre procesos y herramientas.
* Software funcionando sobre documentación extensa.
* Colaboración con el cliente sobre negociación de contratos.
* Respuesta al cambio sobre seguir un plan rígido.

# **DESARROLLO**

## 3.1 Análisis de los requerimientos

**3.1.1 Requerimientos funcionales**

Se presenta los siguientes requerimientos funcionales para el desarrollo de la aplicación:

|  |  |  |
| --- | --- | --- |
| **#** | **Requerimientos** | **Descripción** |
| RF01 | Inicio de sesión | El usuario puede iniciar sesión con correo y contraseña. |
| RF02 | Registro de usuario | El usuario puede crear una cuenta proporcionando correo y contraseña. |
| RF03 | Acceso con redes sociales | Permite iniciar sesión con cuentas de Google y Móvil. |
| RF04 | Visualizar panel de navegación | Despliega menú lateral con acceso a Dashboard, Cuenta, Ajustes, Favoritos. |
| RF05 | Visualizar dashboard | Muestra categorías disponibles de contenido. |
| RF06 | Buscar categoría | Permite buscar categorías en el dashboard. |
| RF07 | Agregar categoría (admin) | El administrador puede crear nuevas categorías. |
| RF08 | Agregar contenido (admin) | El administrador puede subir contenido (PDF), asignarle título, descripción y categoría. |
| RF09 | Visualizar contenido por categoría | Los usuarios pueden ver los contenidos disponibles dentro de cada categoría. |
| RF10 | Ver detalle de contenido | Los usuarios pueden ver la descripción, nombre, tamaño, número de vistas y comentarios. |
| RF11 | Descargar contenido | Los usuarios pueden descargar el contenido (PDF). |
| RF12 | Marcar como favorito | El usuario puede marcar contenidos como favoritos. |
| RF13 | Ver contenido más visitado | El sistema muestra una sección con los contenidos más visitados. |
| RF14 | Ver contenido más descargado | El sistema muestra una sección con los contenidos más descargados. |
| RF15 | Eliminar categorías o contenidos (admin) | El administrador puede eliminar categorías o contenidos existentes. |

Tabla 1. Tabla de requerimientos funcionales

*Fuente: Elaboración propia*

**3.1.2 Requerimientos no funcionales**

Se presenta los siguientes requerimientos no funcionales:

|  |  |  |
| --- | --- | --- |
| **#** | **Requerimientos** | **Descripción** |
| RNF01 | Usabilidad | La interfaz debe ser intuitiva y amigable para todo tipo de usuario. |
| RNF02 | Rendimiento | Las pantallas deben cargar en menos de 3 segundos. |
| RNF03 | Seguridad | La autenticación debe ser segura y manejar cifrado de contraseñas. |
| RNF04 | Compatibilidad | La aplicación debe funcionar correctamente en dispositivos móviles Android e iOS. |
| RNF05 | Escalabilidad | La plataforma debe permitir agregar nuevas categorías y tipos de contenido en el futuro. |
| RNF06 | Accesibilidad | La app debe cumplir principios básicos de accesibilidad para personas con dificultades visuales o motoras. |
| RNF07 | Respaldo de datos | Los datos deben almacenarse de forma persistente y tener respaldo regular. |
| RNF08 | Integración con redes sociales | El sistema debe estar integrado con APIs de autenticación de Google y Facebook. |
| RNF09 | Navegación fluida | Las transiciones entre pantallas deben ser suaves y sin errores. |
| RNF10 | Disponibilidad | El sistema debe estar disponible al menos el 99% del tiempo mensual |

Tabla 2. Tabla de requerimientos no funcionales

*Fuente: Elaboración propia*

## 3.2. Diagramas de desarrollo

**3.2.1 Casos de uso**

****

Figura 1.Diagrama de casos de uso

*Fuente: Elaboración propia*

**Casos de uso**

|  |  |  |
| --- | --- | --- |
| **CU01** | Iniciar Sesión en el Sistema | |
|  | **Actores:** | Usuario |
| **Descripción:** | El caso de uso comienza cuando el usuario desea acceder a su cuenta. El usuario proporciona su correo electrónico y contraseña. El sistema valida las credenciales y, si son correctas, le concede acceso al dashboard principal. |
| **Flujo alternativo:** | Error de autenticación.  Si las credenciales son incorrectas, el sistema muestra un mensaje de error y permite al usuario intentarlo de nuevo. |
| **Requerimiento asociado:** | RF01 |

|  |  |  |
| --- | --- | --- |
| **CU02** | Registrar una Nueva Cuenta | |
|  | **Actores:** | Usuario |
| **Descripción:** | El usuario selecciona la opción de registro y completa un formulario con su correo electrónico y una contraseña. El sistema verifica que el correo no esté en uso, crea la nueva cuenta y notifica al usuario que el registro ha sido exitoso. |
| **Flujo alternativo:** | Correo existente.  Si el correo ya está registrado, el sistema informa al usuario y le sugiere iniciar sesión o usar un correo diferente. |
| **Requerimiento asociado:** | RF02 |

|  |  |  |
| --- | --- | --- |
| **CU03** | Autenticarse con Redes Sociales | |
|  | **Actores:** | Usuario |
| **Descripción:** | En la pantalla de inicio de sesión, el usuario elige autenticarse mediante un proveedor externo (Google o Móvil). El sistema lo redirige al servicio del proveedor para que autorice el acceso. Una vez autorizado, el sistema recibe la confirmación y le concede acceso al usuario. |
| **Flujo alternativo:** | N/A. |
| **Requerimiento asociado:** | RF03 |

|  |  |  |
| --- | --- | --- |
| **CU04** | Explorar y Buscar Contenido | |
|  | **Actores:** | Usuario |
| **Descripción:** | Al iniciar sesión, el sistema presenta el dashboard con todas las categorías de contenido disponibles. El usuario puede visualizar las categorías o utilizar una barra de búsqueda para filtrar y encontrar una categoría específica por su nombre. |
| **Flujo alternativo:** | Sin resultados.  Si la búsqueda no arroja resultados, el sistema muestra un mensaje indicándolo. |
| **Requerimiento asociado:** | RF04, RF05, RF06 |

|  |  |  |
| --- | --- | --- |
| **CU05** | Gestionar Categorías | |
|  | **Actores:** | Administrador |
| **Descripción:** | Este caso de uso permite al administrador realizar tres acciones principales:   * Crear: El administrador accede al panel de gestión, selecciona "Crear Categoría", ingresa un nombre y la guarda.      * Eliminar: El administrador selecciona una categoría existente y confirma su eliminación. |
| **Flujo alternativo:** | N/A. |
| **Requerimiento asociado:** | RF07, RF15 |

|  |  |  |
| --- | --- | --- |
| **CU06** | Gestionar Contenido | |
|  | **Actores:** | Administrador |
| **Descripción:** | Este caso de uso permite al administrador realizar tres acciones principales:   * Crear: El administrador selecciona "Agregar Contenido", completa un formulario (título, descripción), selecciona la categoría y sube el archivo PDF correspondiente.      * Eliminar: El administrador localiza un contenido existente y confirma su eliminación permanente del sistema. |
| **Flujo alternativo:** | N/A. |
| **Requerimiento asociado:** | RF08, RF15 |

|  |  |  |
| --- | --- | --- |
| **CU07** | Consultar Contenido | |
|  | **Actores:** | Usuario |
| **Descripción:** | El usuario selecciona una categoría para ver una lista de los contenidos disponibles en ella. Luego, hace clic en un contenido de interés para ver su página de detalle, que incluye descripción, nombre, tamaño, contador de vistas y comentarios. El sistema incrementa el contador de vistas cada vez que se accede al detalle. |
| **Flujo alternativo:** | N/A. |
| **Requerimiento asociado:** | RF09, RF10 |

**3.2.2 Diagrama de actividades**

* + **Diagrama de Actividades: Iniciar Sesión**

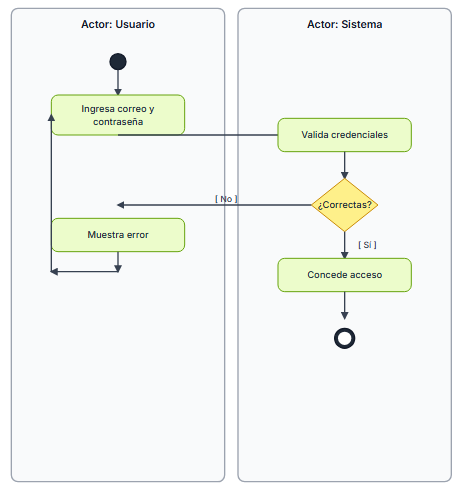
****

Figura 2. Diagrama de actividad

*Fuente: Elaboración propia*

* + **Diagrama de Actividades: Registrar Nuevo Usuario**

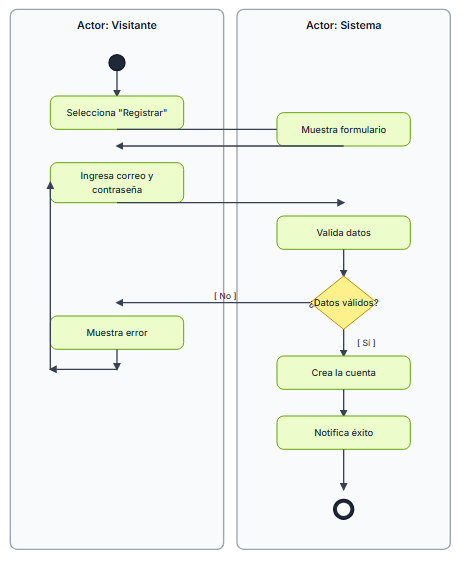
****

Figura 3. Diagrama de actividad

*Fuente: Elaboración propia*

* + **Diagrama de Actividades: Descargar Contenido**

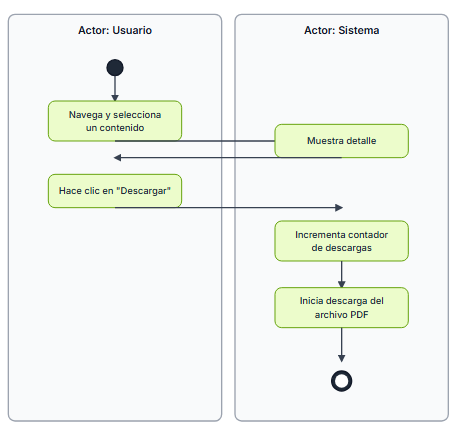
****

Figura 4. Diagrama de actividad

*Fuente: Elaboración propia*

**3.2.3 Diagrama de clases**

E El diagrama de clases del sistema móvil “Estoy Contigo” fue elaborado con base en los requerimientos funcionales del proyecto, tomando en cuenta la estructura de información necesaria para satisfacer las funcionalidades descritas. Dado que la aplicación utiliza Firebase como base de datos NoSQL, el modelo evita relaciones explícitas entre entidades y privilegia estructuras independientes con campos referenciales.

El diseño considera las siguientes clases principales: Usuario, Contenido, Categoria y ChatMessage. Además, se incorpora una nota descriptiva que representa la estructura anidada de los comentarios, los cuales se almacenan como subestructura dentro de cada contenido. La disposición de clases busca reflejar la organización lógica del sistema y la interacción entre usuarios, contenidos y mensajes.

A continuación, se detallan las clases y atributos que forman parte del modelo:

**Usuario**

Representa a cualquier persona que interactúa con la aplicación, sea un estudiante o un administrador.

* uid: Identificador único del usuario en Firebase.
* nombre: Nombre completo registrado.
* email: Dirección de correo electrónico.
* tipo: Rol del usuario (por ejemplo, “usuario” o “administrador”).
* favoritos: Lista de identificadores de contenidos marcados como favoritos.
* fechaRegistro: Fecha en que el usuario se registró en la plataforma.

**Categoria**

Agrupa los recursos compartidos en la aplicación por temas o áreas de interés.

* id: Identificador único de la categoría.
* nombre: Título o denominación de la categoría.
* fechaCreacion: Fecha en que la categoría fue creada.

**Contenido**

Corresponde a los recursos informativos compartidos a través de la aplicación.

* id: Identificador del contenido.
* titulo: Título del recurso.
* descripcion: Breve descripción del contenido.
* categoriaId: Identificador de la categoría a la que pertenece el recurso.
* archivoUrl: Enlace directo al archivo (PDF, video, imagen, etc.).
* tamaño: Tamaño del archivo en MB o KB.
* vistas: Número de veces que el contenido ha sido visualizado.
* descargas: Número de descargas registradas.
* fechaPublicacion: Fecha en que fue publicado el recurso.
* autorUid: Identificador del usuario que publicó el contenido.
* comentarios: Campo que almacena un conjunto de comentarios en formato Map o subdocumento JSON. No se define como clase separada, ya que está embebido en el documento de contenido.

Estructura de comentarios (incluida como nota en el diagrama):

Cada comentario asociado a un contenido posee los siguientes campos:

* autorUid: Identificador del autor del comentario.
* nombreAutor: Nombre visible del autor.
* texto: Mensaje escrito.
* fecha: Fecha en que se publicó el comentario.

**ChatMessage**

Almacena el historial de interacciones vía chat, tanto del usuario como del sistema (bot o especialista).

* id: Identificador del mensaje.
* mensaje: Texto del mensaje enviado.
* esBot: Booleano que indica si el mensaje fue generado por el sistema.
* uid: Identificador del usuario al que pertenece el mensaje.
* timestamp: Fecha y hora del envío del mensaje.

Este modelo de clases refleja la organización de la información y los componentes clave que se requieren para implementar la funcionalidad completa de la aplicación “Estoy Contigo” en un entorno NoSQL, asegurando flexibilidad y escalabilidad.

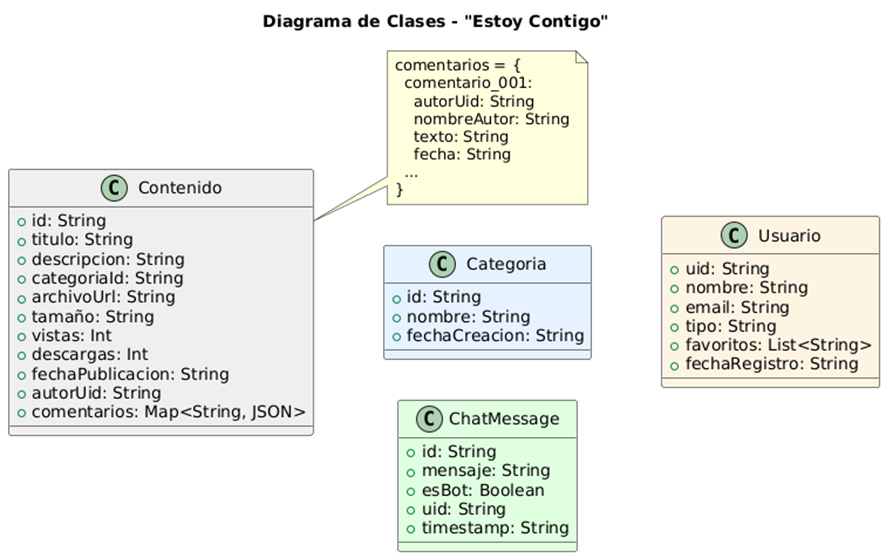


Figura 5. Diagrama de actividad

*Fuente: Elaboración propia*

**3.2.4 Diagrama de paquetes**

E El diagrama de paquetes muestra la estructura modular del sistema móvil “Estoy Contigo”, organizado en bloques lógicos que agrupan clases con responsabilidades específicas. Esta organización responde al principio de separación de responsabilidades y favorece la escalabilidad, reutilización de código y mantenibilidad general del sistema.

Cada paquete representa un nivel funcional del sistema, y las clases dentro de ellos cumplen roles definidos dentro de la arquitectura general:

**presentation**

Este paquete agrupa las pantallas e interfaces visuales con las que interactúa el usuario, así como los controladores asociados a dichas vistas. Incluye:

* PantallaLogin: interfaz para que los usuarios accedan al sistema.
* PantallaContenido: vista que permite consultar los recursos publicados.
* ControladorChat: componente encargado de manejar la lógica y eventos del chat.

**domain**

Contiene las entidades o modelos centrales del sistema. Estas clases representan los objetos reales que maneja la aplicación:

* Usuario: entidad que representa a cada persona registrada.
* Contenido: objeto que encapsula los recursos compartidos.
* ChatMessage: clase que modela un mensaje dentro del sistema de chat.
* Categoria: entidad que agrupa los contenidos temáticamente.

**data**

En este paquete se ubican los repositorios responsables de acceder, guardar y actualizar información en la base de datos. Cada repositorio se especializa en una entidad del dominio:

* UsuarioRepository: gestiona operaciones sobre usuarios.
* ContenidoRepository: maneja el acceso a los contenidos.
* ChatRepository: administra el flujo de datos del sistema de mensajería.

**firebase\_service**

Contiene los servicios encargados de interactuar directamente con Firebase. Estos encapsulan la lógica de conexión y ejecución de operaciones sobre los servicios que ofrece esta plataforma:

* FirestoreService: realiza operaciones sobre colecciones y documentos de Firestore.
* AuthService: maneja el registro e inicio de sesión con Firebase Authentication.
* StorageService: administra archivos multimedia a través de Firebase Storage.

utils

Incluye funciones auxiliares y herramientas de uso transversal. Estas clases permiten validar datos, formatear información o realizar tareas de apoyo sin acoplarse a una funcionalidad específica:

* StringValidator: validaciones comunes para cadenas de texto.
* DateHelper: funciones relacionadas con el manejo de fechas.

**Justificación de las relaciones entre paquetes**

Las flechas presentes en el diagrama indican dependencias lógicas entre paquetes. Estas relaciones son habituales en diagramas de paquetes y representan el flujo de uso entre módulos, sin que ello implique acoplamientos directos a nivel de base de datos, lo cual es coherente con la arquitectura basada en Firebase Firestore (NoSQL). A continuación, se explican:

* utils → presentation: porque las vistas requieren funciones auxiliares, como validación de texto o formato de fechas.
* utils → data: dado que los repositorios también hacen uso de funciones de ayuda al momento de guardar o procesar información.
* presentation → domain: ya que las pantallas y controladores visuales trabajan con entidades como Usuario, Contenido o ChatMessage.
* presentation → data: porque las pantallas solicitan datos o envían información a través de los repositorios.
* data → domain: los repositorios transforman datos en instancias de entidades del dominio y viceversa.
* data → firebase\_service: los repositorios delegan las operaciones de persistencia o autenticación a los servicios que interactúan directamente con Firebase.

Este diseño modular refleja las buenas prácticas en la construcción de aplicaciones móviles modernas que utilizan arquitectura limpia, promoviendo el bajo acoplamiento entre capas y favoreciendo el uso de Firebase como backend escalable.

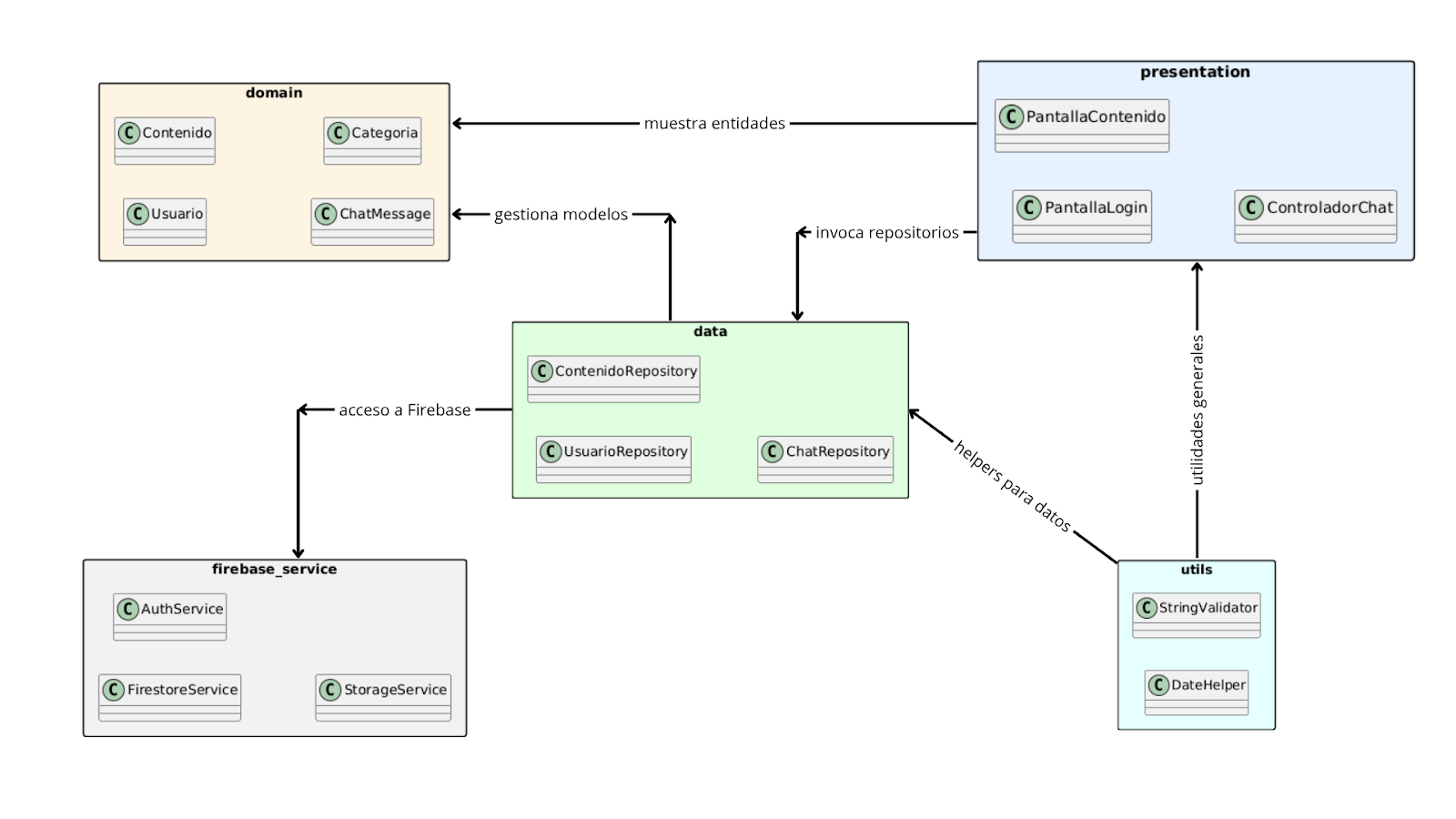


Figura 6. Diagrama de paquete.

*Fuente: Elaboración propia*

## 3.6 Diseño

**3.6.1 Diseño de la base de datos**

La Figura 05 muestra la estructura de base de datos NoSQL diseñada para Firebase, la cual facilitará la integración del aplicativo con el sistema de almacenamiento de información.

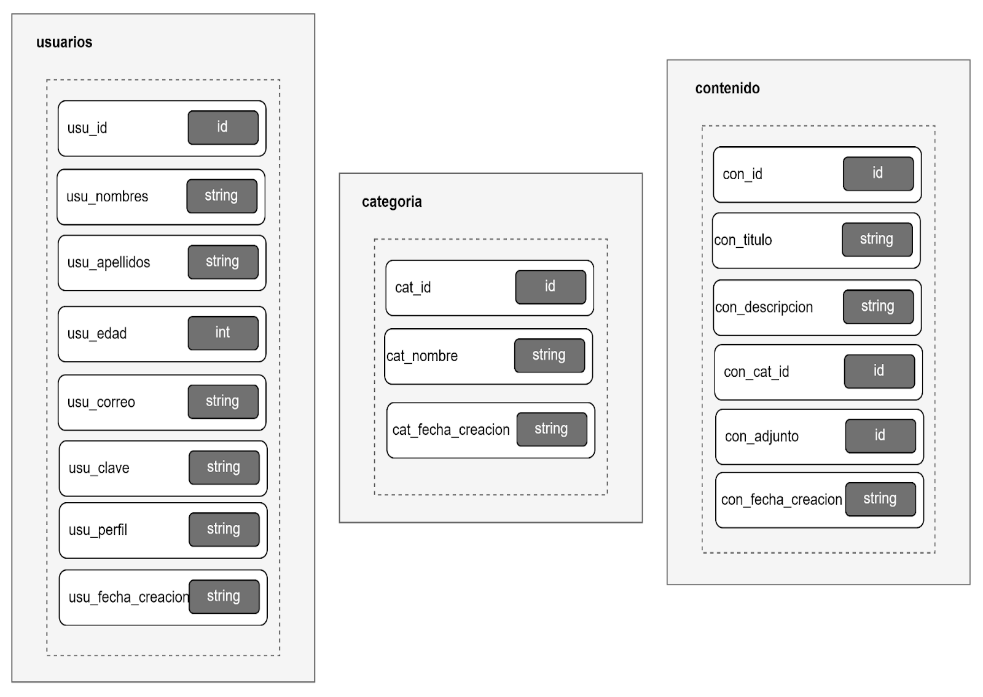
****

Figura 5. Diagrama de paquete

*Fuente: Elaboración propia*

**3.6.2 Diseño de la interfaz (Mockups)**

La elaboración de la interfaz gráfica se basó en los requerimientos del proyecto. Para el diseño de las maquetas se utilizó la plataforma Moqups, en su versión gratuita, debido a su facilidad de uso e interfaz intuitiva. Esta herramienta permite crear diversas pantallas que simulan las funcionalidades principales que tendría la aplicación móvil final.

Los diseños de las pantallas que se crearon son las siguientes:

**Cliente**

* Pantalla de inicio de sesión, registro y perfil

En la Figura 06 muestra el wireframe diseñado para la autentificación del usuario cliente, el cual le permitirá iniciar sesión usando un usuario y contraseña, a su vez le permitirá registrarse en caso no tenga una cuenta activa y finalmente le permitirá visualizar su panel de perfil cuando haya iniciado sesión.

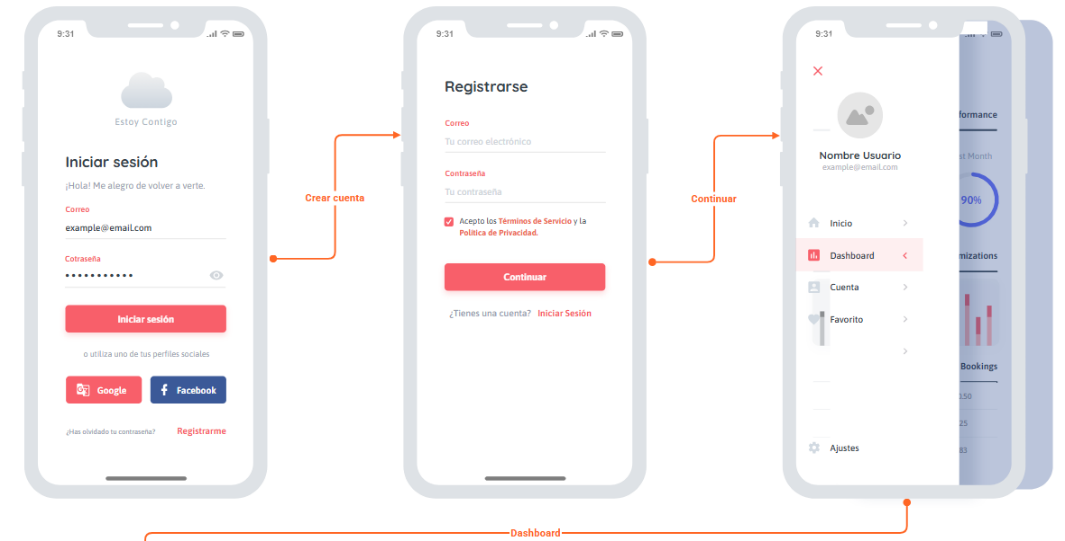


Figura 6. Interfaz de inicio de sesión, registro y perfil

*Fuente: Elaboración propia*

* Pantalla de Dashboard, listado de contenidos y detalle

En la Figura 07 se muestra la vista general de dashboard que muestra el listado de las categorías existentes, así mismo muestra los items existentes por cada categoría y finalmente el detalle de cada ítem.

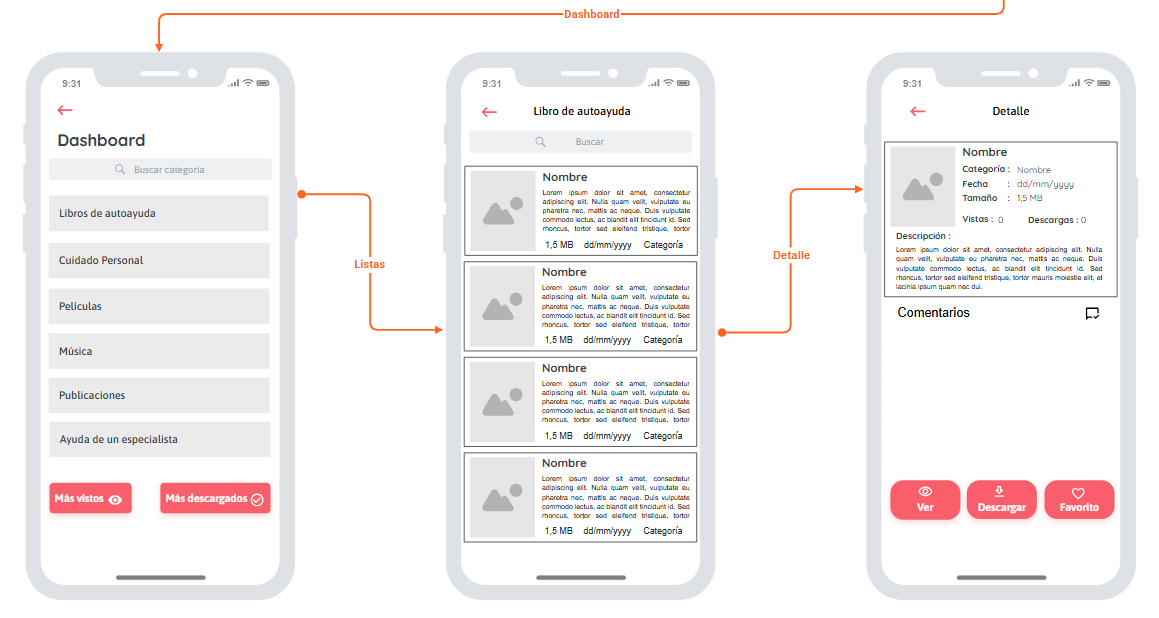
****

Figura 7. Interfaz de dashboard, listado de contenidos y detalle

*Fuente: Elaboración propia*

**Administrador**

* Pantalla de inicio de sesión y perfil

En la Figura 08, se muestra el sistema de inicio de sesión del administrador, el cual únicamente le permitirá ingresar por un usuario y contraseña asignado, una vez haya ingresado los accesos correspondientes le mostrará su perfill.

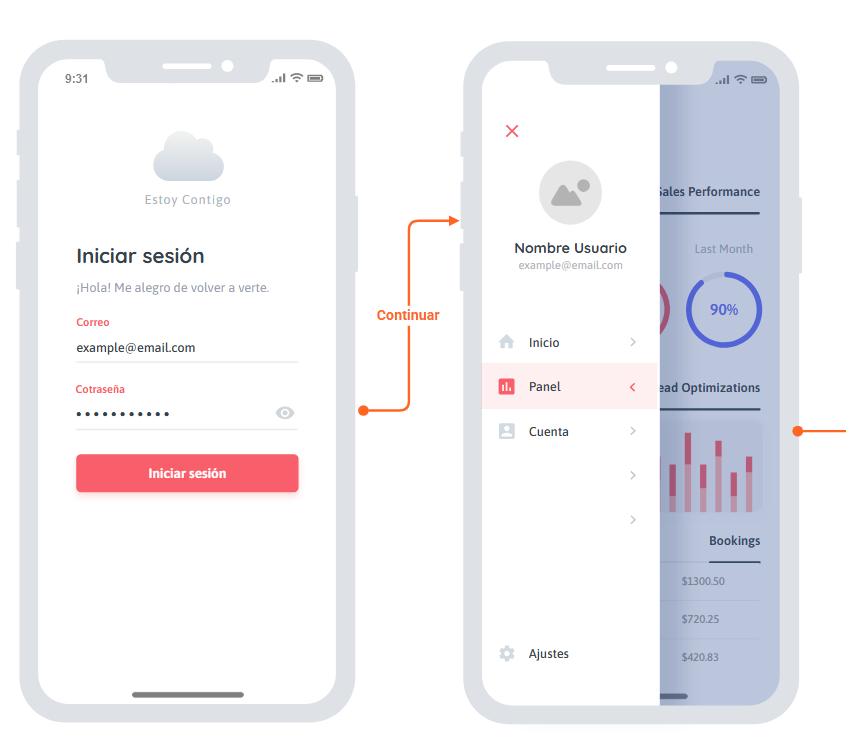


Figura 8. Interfaz de inicio de sesión y perfil

*Fuente: Elaboración propia*

* Pantalla de dashboard

En la Figura 09 se muestra el listado de las categorías existentes, en esta interfaz permitirá controlar cómo agregar más categorías, agregar detalle o eliminar de acuerdo a lo que el usuario administrador seleccione.

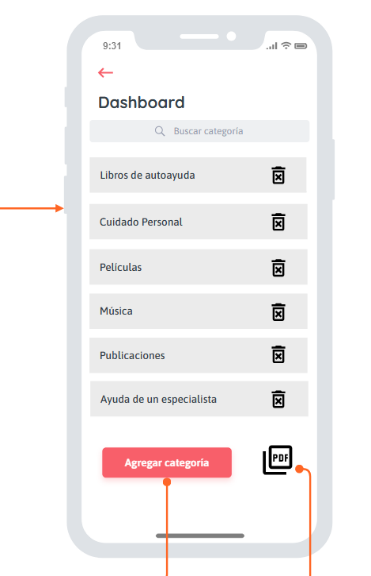


Figura 9. Interfaz de dashboard

*Fuente: Elaboración propia*

* Pantalla de agregar categoría y agregar contenido

En la Figura 10 se muestra la interfaz para agregar categoría y adicionar contenido para una categoría.

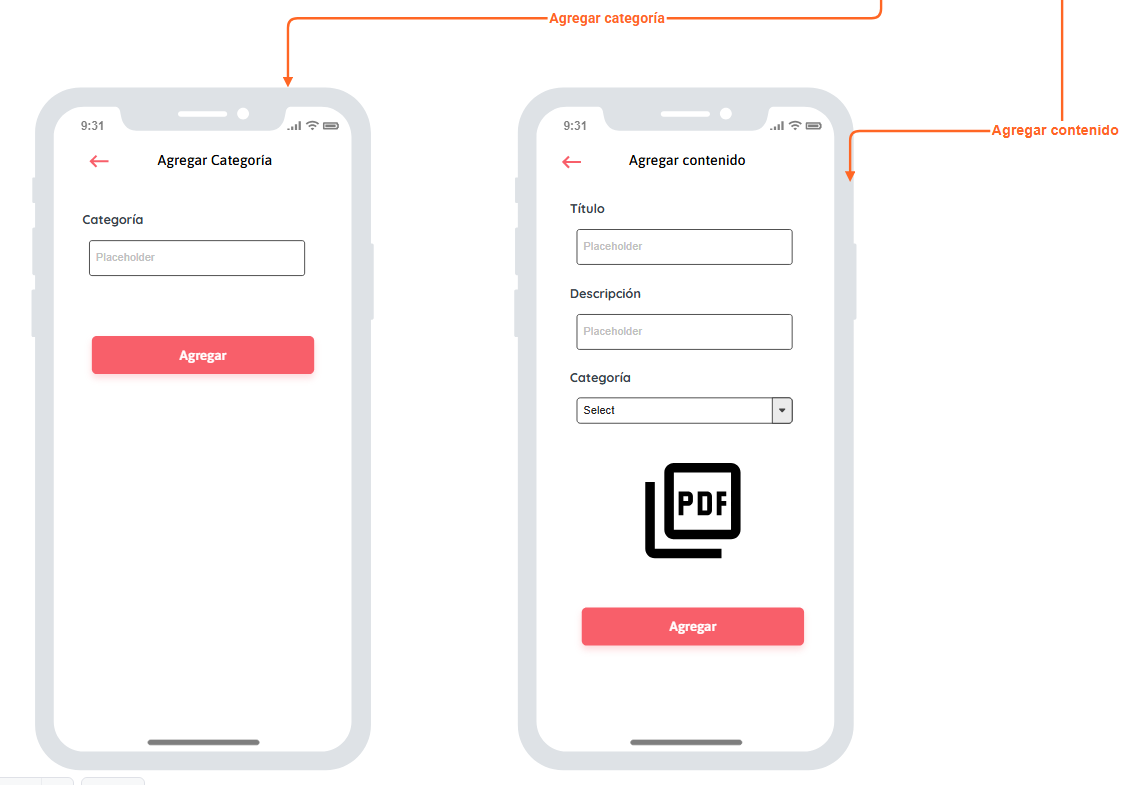


Figura 10. Interfaz de registro de categoría y contenido

*Fuente: Elaboración propia*

## 3.7 Desarrollo de la Base de Datos

En la Figura 11 se muestra la estructura de la base de datos en Firebase real time database para la aplicación "EstoyContigo". La base de datos se organiza en tres nodos principales: categoría, contenido y usuarios, cada uno con sus campos y tipos de datos.

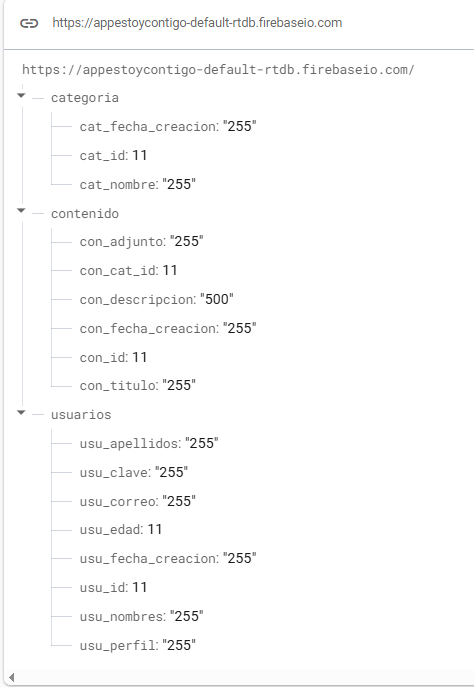
****

Figura 11. Interfaz de registro de categoría y contenido

*Fuente: Elaboración propia*

## 3.8 Desarrollo de la interfaz

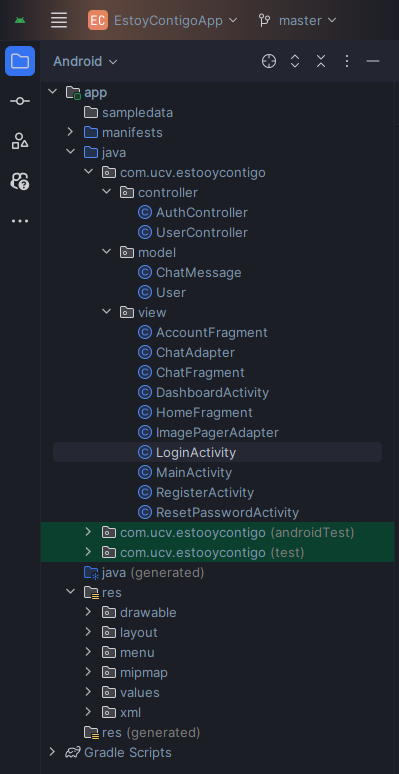
****

Figura 12. Estructura del Proyecto

*Fuente: Elaboración propia*

**Las ventanas.xml que se usa para visualizar el proyecto:**

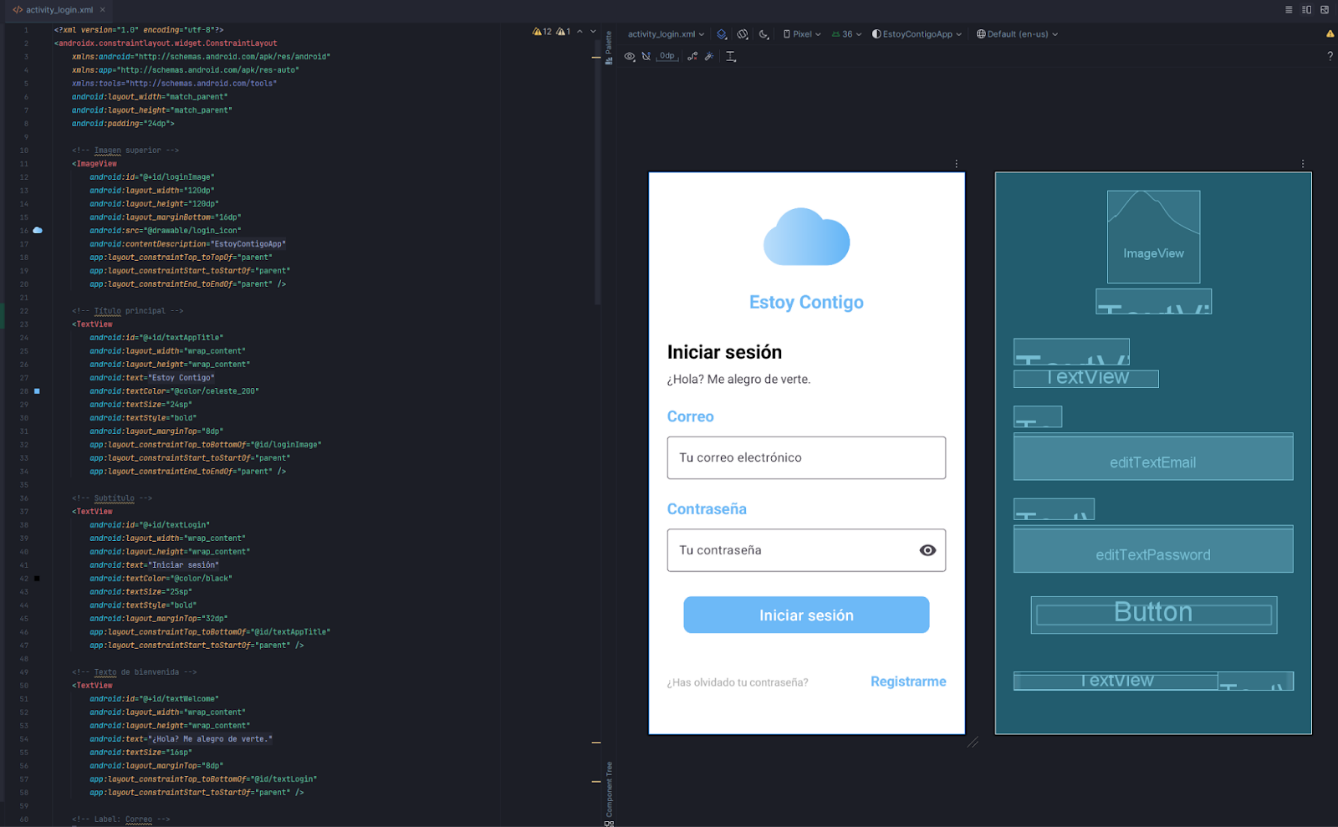
****

Figura 14. activity\_login.xml

*Fuente: Elaboración propia*

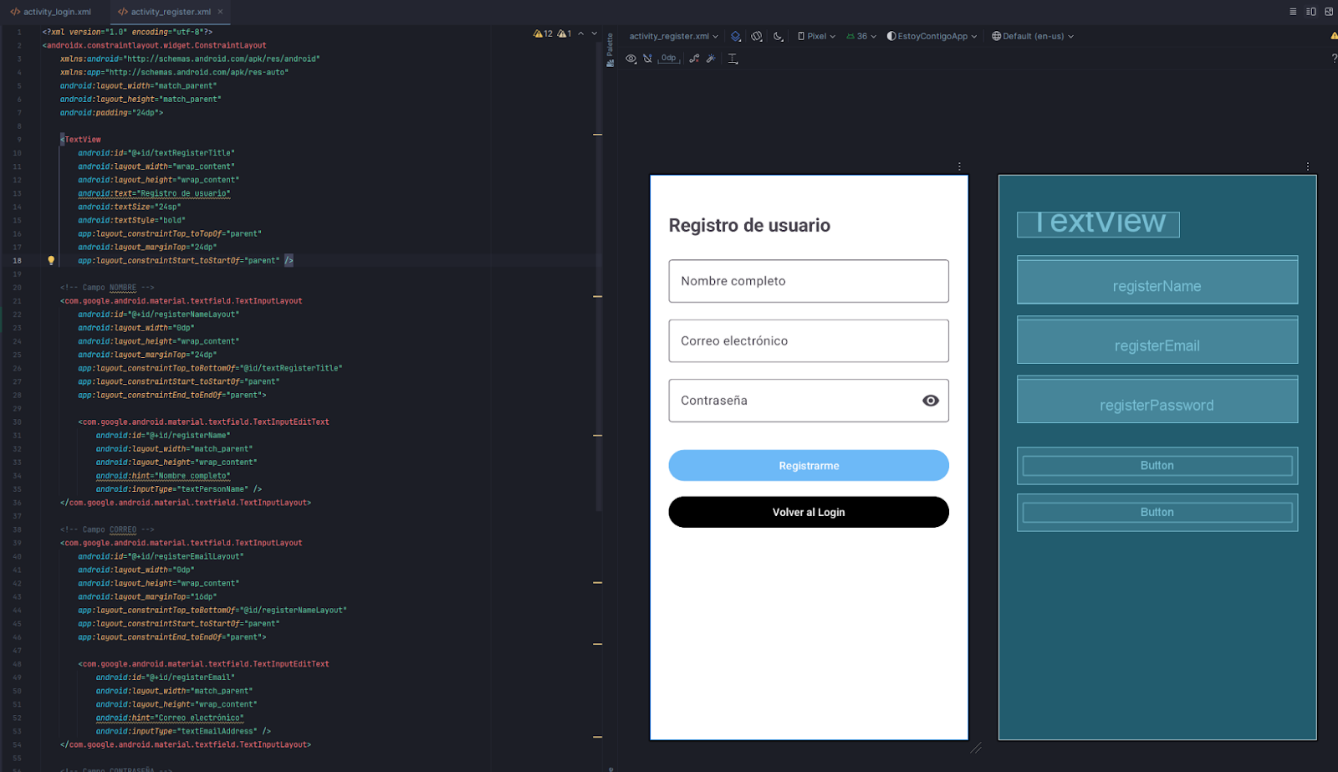
****

Figura 15. register\_activity.xml

*Fuente: Elaboración propia*

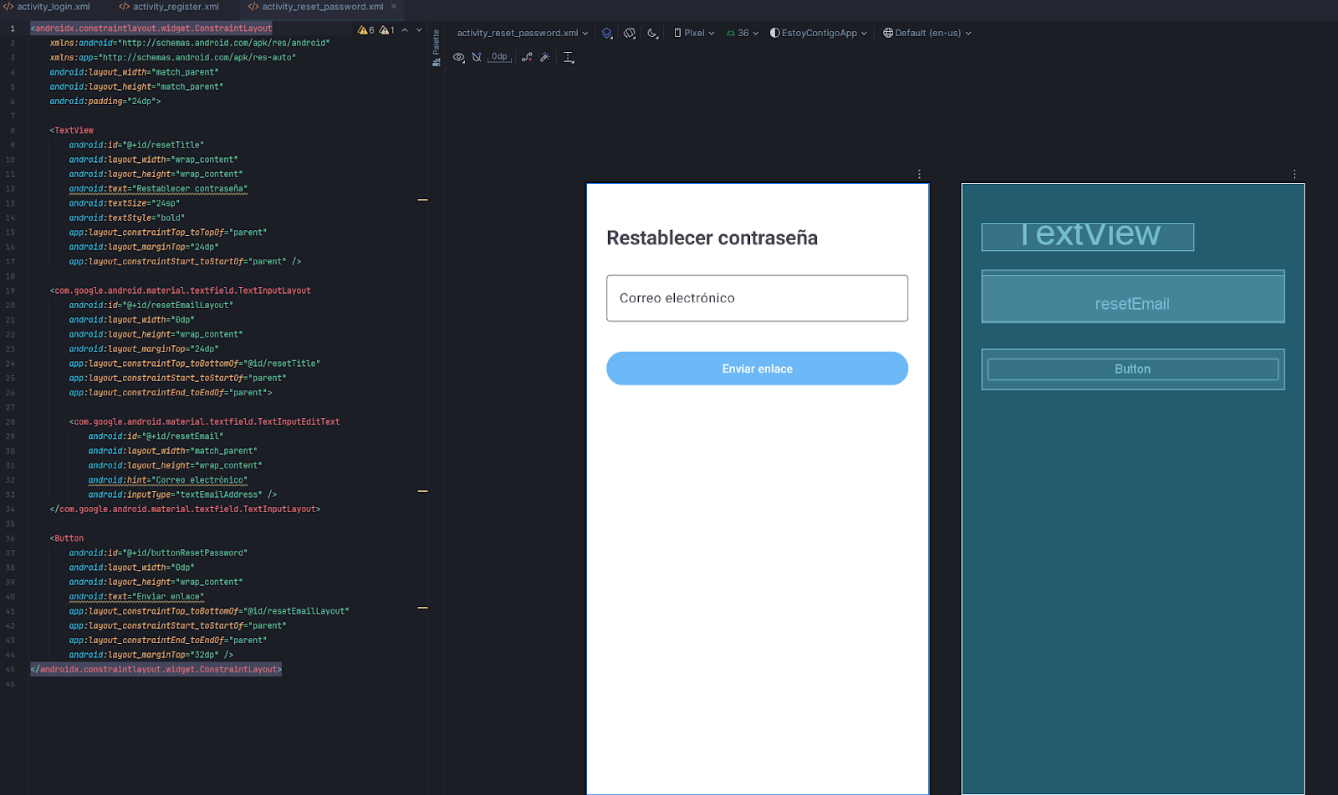
**

Figura 16. activity\_reset\_pasasword.xml

*Fuente: Elaboración propia*

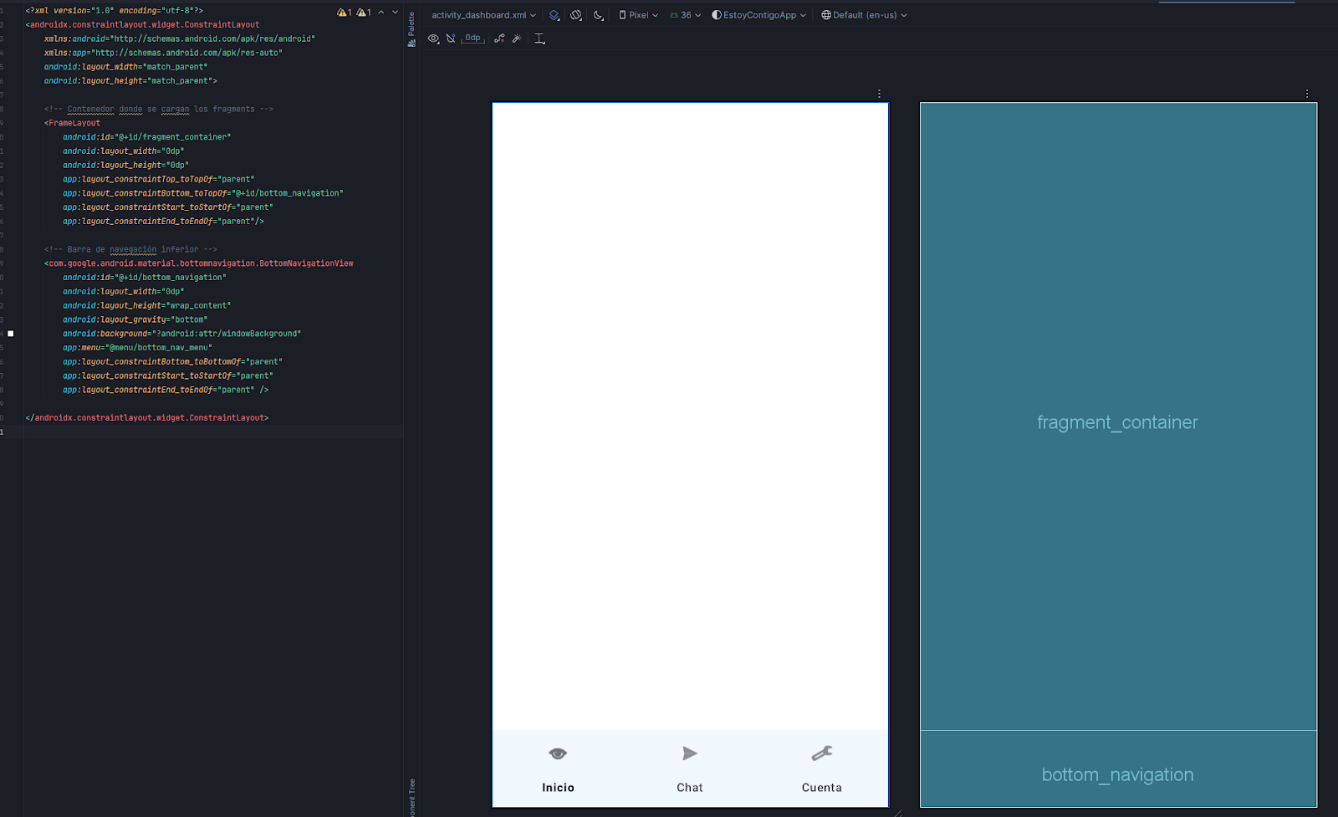
****

Figura 17. activity\_dashboard.xml

*Fuente: Elaboración propia*

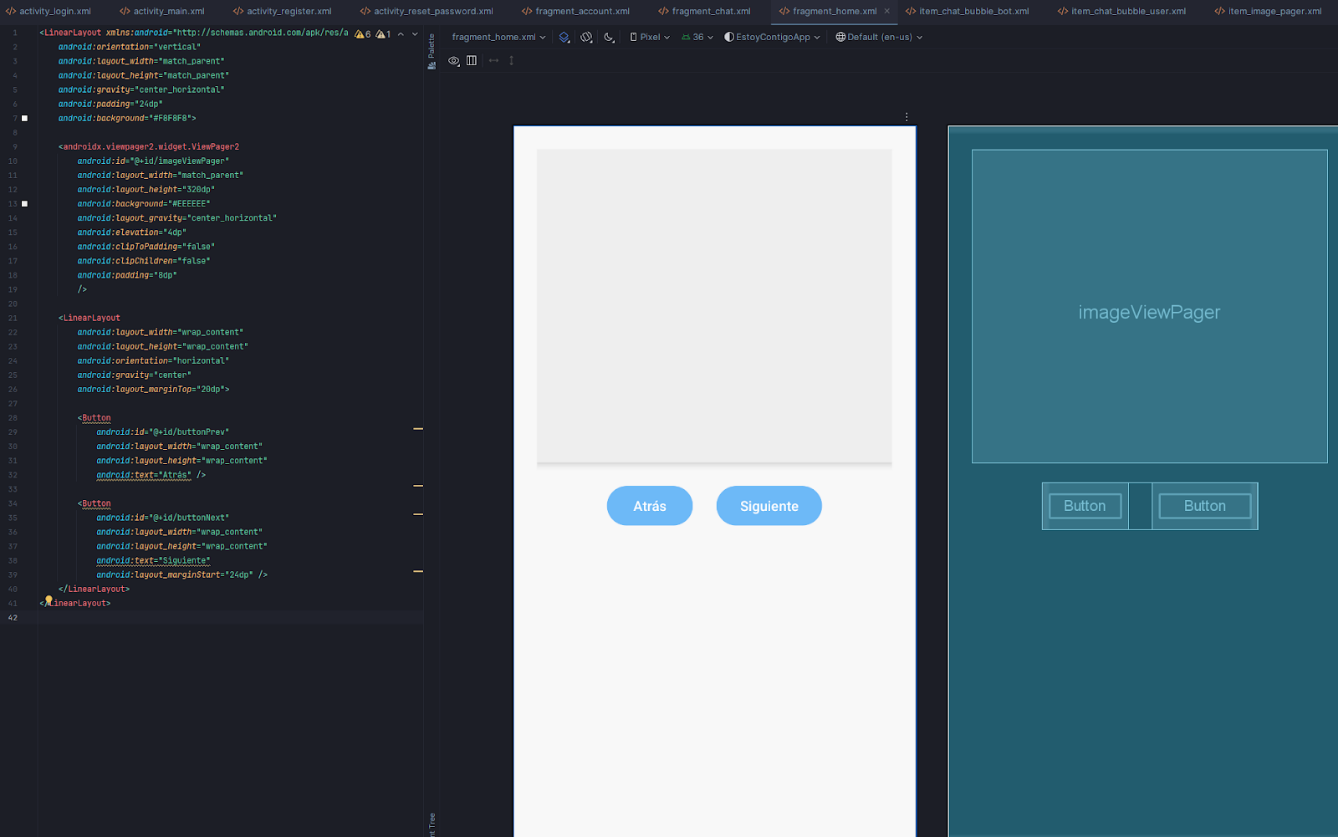
**

Figura 18. framgent\_home.xml

*Fuente: Elaboración propia*

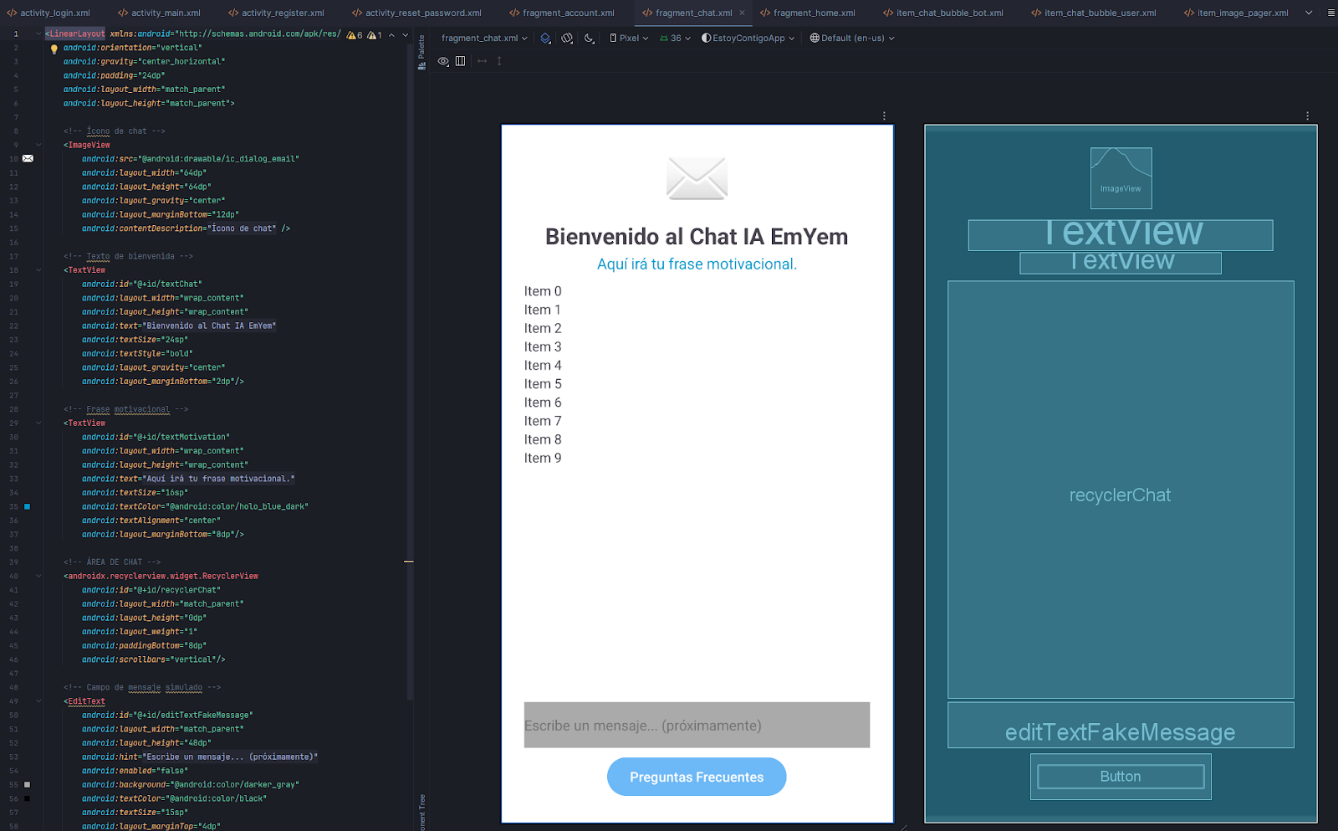
**

Figura 18. fragment\_chat.xml

*Fuente: Elaboración propia*

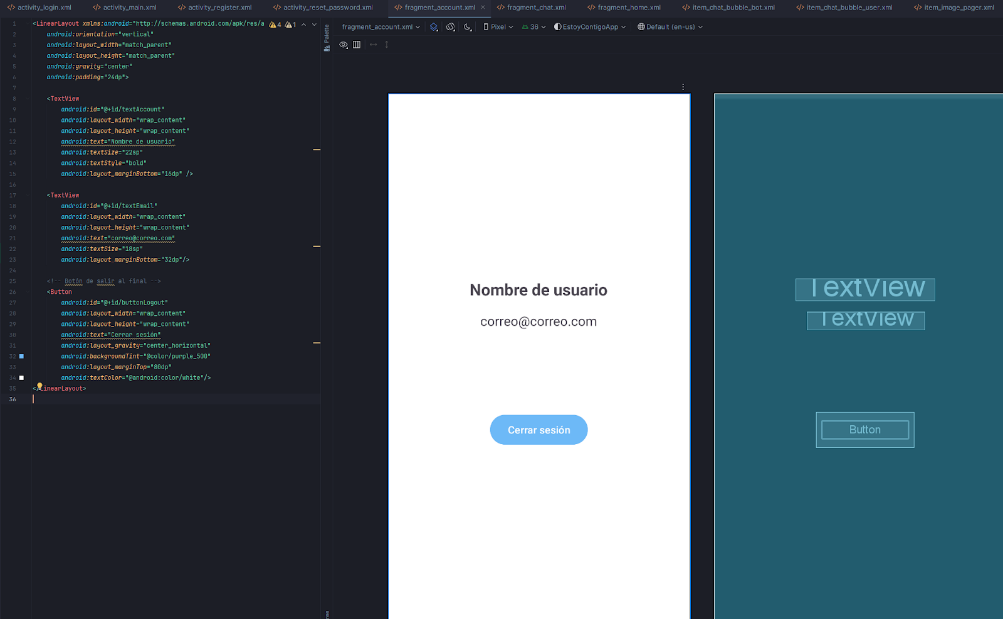


Figura 18. fragment\_account.xml

*Fuente: Elaboración propia*

**Estructura MVC, contenido del paquete Controller:**

AutchController.java

***public class* AuthController {**

***public static void* resetPassword(Context context, String email) {**

**FirebaseAuth.*getInstance*().sendPasswordResetEmail(email)**

**.addOnCompleteListener(task -> {**

***if* (task.isSuccessful()) {**

**Toast.*makeText*(context, "Se envió un enlace para restablecer tu contraseña", Toast.LENGTH\_LONG).show();**

**} *else* {**

**Toast.*makeText*(context, "Error al enviar el correo. Inténtalo de nuevo", Toast.LENGTH\_SHORT).show();**

**}**

**});**

**}**

***public static void* registerUser(Context context, String name, String email, String password) {**

**FirebaseAuth auth = FirebaseAuth.*getInstance*();**

**auth.createUserWithEmailAndPassword(email, password)**

**.addOnCompleteListener(task -> {**

***if* (task.isSuccessful()) {**

**FirebaseUser user = auth.getCurrentUser();**

**// Guardar nombre y correo en Realtime Database**

***if* (user != *null*) {**

**String uid = user.getUid();**

**HashMap<String, Object> userData = *new* HashMap<>();**

**userData.put("name", name);**

**userData.put("email", email);**

**FirebaseDatabase.*getInstance*().getReference("users")**

**.child(uid)**

**.setValue(userData)**

**.addOnCompleteListener(dbTask -> {**

***if* (dbTask.isSuccessful()) {**

**Toast.*makeText*(context, "Registro exitoso. Verifica tu correo antes de iniciar sesión.", Toast.LENGTH\_LONG).show();**

**Intent intent = *new* Intent(context, LoginActivity.*class*);**

**context.startActivity(intent);**

***if* (context *instanceof* Activity) {**

**((Activity) context).finish();**

**}**

**} *else* {**

**Toast.*makeText*(context, "No se pudo guardar el usuario en la base de datos.", Toast.LENGTH\_SHORT).show();**

**}**

**});**

**}**

**// Enviar correo de verificación**

***if* (user != *null*) {**

**user.sendEmailVerification();**

**}**

**} *else* {**

**Toast.*makeText*(context, "Error al registrar: " + task.getException().getMessage(), Toast.LENGTH\_LONG).show();**

**}**

**});**

**}**

***public static void* loginUser(Context context, String email, String password) {**

**FirebaseAuth auth = FirebaseAuth.*getInstance*();**

**auth.signInWithEmailAndPassword(email, password)**

**.addOnCompleteListener(task -> {**

***if* (task.isSuccessful()) {**

**FirebaseUser user = auth.getCurrentUser();**

***if* (user != *null*) {**

***if* (user.isEmailVerified()) {**

**// ¡Correo verificado! Llevar al dashboard**

**Toast.*makeText*(context, "Bienvenido", Toast.LENGTH\_SHORT).show();**

**Intent intent = *new* Intent(context, DashboardActivity.*class*);**

**context.startActivity(intent);**

**// Limpiar la pila de actividades  // Agrega los flags para limpiar el back stack**

**intent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);**

**// Si vienes de un Activity, puedes cerrarlo para no volver atrás**

***if* (context *instanceof* Activity) {**

**((Activity) context).finish();**

**}**

**} *else* {**

**// Si el correo no está verificado**

**Toast.*makeText*(context, "Por favor, verifica tu correo electrónico antes de iniciar sesión.", Toast.LENGTH\_LONG).show();**

**auth.signOut();**

**}**

**}**

**} *else* {**

**Toast.*makeText*(context, "Error al iniciar sesión: " + task.getException().getMessage(), Toast.LENGTH\_LONG).show();**

**}**

**});**

**}**

**}**

UserController.java

*public class* UserController {

*public interface UserDataCallback* {

*void* onDataReceived(User user);

   }

*public static void* getCurrentUserData(*final UserDataCallback* callback) {

       FirebaseUser firebaseUser = FirebaseAuth.*getInstance*().getCurrentUser();

*if* (firebaseUser == *null*) {

           callback.onDataReceived(*null*);

*return*;

       }

       DatabaseReference ref = FirebaseDatabase.*getInstance*()

               .getReference("users")

               .child(firebaseUser.getUid());

       ref.addListenerForSingleValueEvent(*new* ValueEventListener() {

           @Override

*public void* onDataChange(DataSnapshot snapshot) {

               User user = snapshot.getValue(User.*class*);

               callback.onDataReceived(user);

           }

           @Override

*public void* onCancelled(DatabaseError error) {

               callback.onDataReceived(*null*);

           }

       });

   }

}

**Estructura MVC, contenido del paquete Model:**

ChatMessage.java

***public class* ChatMessage {**

***private* String message;**

***private boolean* isBot; // true si es respuesta, false si es pregunta/usuario**

***public* ChatMessage(String message, *boolean* isBot) {**

***this*.message = message;**

***this*.isBot = isBot;**

**}**

***public* String getMessage() { *return* message; }**

***public boolean* isBot() { *return* isBot; }**

**}**

User.java

***public class* User {**

***private* String name;**

***private* String email;**

***public* User() {} // Constructor vacío requerido por Firebase**

***public* User(String name, String email) {**

***this*.name = name;**

***this*.email = email;**

**}**

***public* String getName() { *return* name; }**

***public* String getEmail() { *return* email; }**

***public void* setName(String name) { *this*.name = name; }**

***public void* setEmail(String email) { *this*.email = email; }**

**}**

**Estructura MVC, contenido del paquete View:**

LoginActivity.java

***public class* LoginActivity *extends* AppCompatActivity {**

***private* EditText editTextEmail, editTextPassword;**

***private* Button buttonLogin;**

***private* TextView textRegister, textForgotPassword;**

**@Override**

***protected void* onCreate(Bundle savedInstanceState) {**

***super*.onCreate(savedInstanceState);**

**setContentView(R.layout.*activity\_login*);**

**editTextEmail = findViewById(R.id.*editTextEmail*);**

**editTextPassword = findViewById(R.id.*editTextPassword*);**

**buttonLogin = findViewById(R.id.*buttonLogin*);**

**textRegister = findViewById(R.id.*textRegister*);**

**textForgotPassword = findViewById(R.id.*textForgotPassword*);**

**buttonLogin.setOnClickListener(v -> {**

**String email = editTextEmail.getText().toString().trim();**

**String password = editTextPassword.getText().toString().trim();**

***if* (email.isEmpty() || password.isEmpty()) {**

**Toast.*makeText*(*this*, "Completa todos los campos", Toast.LENGTH\_SHORT).show();**

***return*;**

**}**

***if* (!android.util.Patterns.EMAIL\_ADDRESS.matcher(email).matches()) {**

**editTextEmail.setError("Formato de correo inválido. Ejemplo: usuario@correo.com");**

***return*;**

**}**

***if* (!password.matches("^[a*-*zA*-*Z0*-*9!@#\\$%\\^&\\\*]{6,20}$")) {**

**editTextPassword.setError("Solo se permiten letras, números y !@#$%^&\*. 6-20 caracteres.");**

***return*;**

**}**

**// Lógica de login.**

**AuthController.*loginUser*(*this*, email, password);**

**});**

**textRegister.setOnClickListener(v -> {**

**Intent intent = *new* Intent(LoginActivity.*this*, RegisterActivity.*class*);**

**startActivity(intent);**

**});**

**textForgotPassword.setOnClickListener(v -> {**

**Intent intent = *new* Intent(LoginActivity.*this*, ResetPasswordActivity.*class*);**

**startActivity(intent);**

**});**

**}**

**}**

**RegisterActivity.java**

***public class* RegisterActivity *extends* AppCompatActivity {**

***private* EditText editTextName, editTextEmail, editTextPassword;**

***private* Button buttonRegister;**

**@Override**

***protected void* onCreate(Bundle savedInstanceState) {**

***super*.onCreate(savedInstanceState);**

**setContentView(R.layout.*activity\_register*);**

**editTextName = findViewById(R.id.*registerName*);**

**editTextEmail = findViewById(R.id.*registerEmail*);**

**editTextPassword = findViewById(R.id.*registerPassword*);**

**buttonRegister = findViewById(R.id.*buttonRegister*);**

**Button buttonBackToLogin = findViewById(R.id.*buttonBackToLogin*);**

**buttonBackToLogin.setOnClickListener(v -> {**

**finish(); // Esto vuelve a la pantalla anterior (LoginActivity)**

**});**

**buttonRegister.setOnClickListener(v -> {**

**String name = editTextName.getText().toString().trim();**

**String email = editTextEmail.getText().toString().trim();**

**String password = editTextPassword.getText().toString().trim();**

***if* (name.isEmpty() || email.isEmpty() || password.isEmpty()) {**

**Toast.*makeText*(*this*, "Completa todos los campos", Toast.LENGTH\_SHORT).show();**

***return*;**

**}**

***if* (!name.matches("^[A*-*Za*-*zÁÉÍÓÚáéíóúñÑ ]{3,40}$")) {**

**editTextName.setError("Solo se permiten letras y espacios. 3-40 caracteres.");**

***return*;**

**}**

***if* (!android.util.Patterns.EMAIL\_ADDRESS.matcher(email).matches()) {**

**editTextEmail.setError("Formato de correo inválido. Ejemplo: usuario@correo.com");**

***return*;**

**}**

***if* (!password.matches("^[a*-*zA*-*Z0*-*9!@#\\$%\\^&\\\*]{6,20}$")) {**

**editTextPassword.setError("Solo se permiten letras, números y !@#$%^&\*. 6-20 caracteres.");**

***return*;**

**}**

**AuthController.*registerUser*(*this*, name, email, password);**

**});**

**}**

**}**

ResetPasswordActivity.java

***public class* ResetPasswordActivity *extends* AppCompatActivity {**

***private* EditText resetEmail;**

***private* Button buttonReset;**

**@Override**

***protected void* onCreate(Bundle savedInstanceState) {**

***super*.onCreate(savedInstanceState);**

**setContentView(R.layout.*activity\_reset\_password*);**

**resetEmail = findViewById(R.id.*resetEmail*);**

**buttonReset = findViewById(R.id.*buttonResetPassword*);**

**buttonReset.setOnClickListener(v -> {**

**String email = resetEmail.getText().toString().trim();**

***if* (email.isEmpty()) {**

**resetEmail.setError("Ingresa tu correo electrónico");**

***return*;**

**}**

***if* (!android.util.Patterns.EMAIL\_ADDRESS.matcher(email).matches()) {**

**resetEmail.setError("Formato de correo inválido. Ejemplo: usuario@correo.com");**

***return*;**

**}**

**AuthController.*resetPassword*(*this*, email); // Llama al controlador**

**});**

**}**

**}**

## 3.9 App “EstoyContigo” en producción

|  |  |  |
| --- | --- | --- |
| Descargar e instar el APK | Pantalla de inicio de sesión | Pantalla de registro |
|  |  |  |
| Verificación de cuenta | Iniciar sesión | Pantalla principal |
|  |  |  |
| Pantalla principal de chat | Conversación con Bot | Configuración de cuenta |
|  |  |  |

Tabla 3. Tabla de paso del App en producción

*Fuente: Elaboración propia*

# **INTERPRETACIÓN**

**Interpretación técnica y funcional**

El desarrollo de la aplicación móvil “Estoy Contigo” se abordó desde un enfoque integral, que combina buenas prácticas de ingeniería de software, diseño centrado en el usuario y pertinencia social. La aplicación fue diseñada para funcionar como una herramienta de acompañamiento emocional dirigida a estudiantes universitarios, incorporando funciones claves como el acceso a contenido motivacional, interacción con chatbot, y gestión personalizada de usuarios.

Desde el punto de vista técnico, se implementó una arquitectura basada en el patrón Modelo-Vista-Controlador (MVC), favoreciendo la separación de responsabilidades entre la lógica de presentación, la lógica de negocio y la gestión de datos. Esta estructura modular permitió desarrollar un sistema limpio, escalable y fácilmente mantenible (Alonso & Mendoza, 2023). A nivel de modelado, se emplearon diagramas UML (casos de uso, clases, actividades y paquetes) que permitieron representar el comportamiento esperado del sistema, sus componentes principales y las interacciones entre los distintos actores.

El diseño de interfaz fue desarrollado mediante la herramienta Moqups para prototipado, y posteriormente trasladado a archivos XML en Android Studio. Las vistas creadas incluyen pantallas de inicio de sesión, registro, dashboard, visualización de contenido, chat y perfil de usuario, priorizando la experiencia de usuario, accesibilidad y diseño intuitivo.

**Interpretación sociotécnica y de impacto**

La implementación de la aplicación “Estoy Contigo” no solo responde a una necesidad técnica, sino también social. En el contexto universitario peruano, muchos estudiantes no acceden fácilmente a servicios de salud mental, ya sea por estigmas sociales, falta de recursos o barreras institucionales. En ese sentido, esta aplicación actúa como un mecanismo de apoyo preventivo, brindando herramientas accesibles y emocionalmente conectadas, lo que incrementa su valor e impacto social.

Este enfoque se alinea con los hallazgos de Linardon et al. (2024), quienes identificaron que las aplicaciones móviles con elementos de monitoreo emocional y técnicas cognitivo-conductuales logran reducir niveles de ansiedad y depresión en jóvenes. De manera similar, Bress et al. (2024) destacaron que el soporte emocional mediante interacciones personalizadas con apps mejora la salud mental, sobre todo cuando se incluyen funciones como bots empáticos y contenido motivador, elementos presentes en “Estoy Contigo”.

A nivel nacional, se evidencia consistencia con los resultados obtenidos por Pacora Chávez y Bezada Sánchez (2023), quienes desarrollaron una app para estudiantes con síntomas de depresión, concluyendo que la adaptación cultural del contenido y el diseño centrado en el usuario son factores clave en la aceptación y efectividad de estas soluciones tecnológicas.

En contraste, estudios como el de Gómez-Gonzales et al. (2022) señalaron que muchas apps extranjeras fracasan en su implementación en países como Perú por carecer de elementos culturalmente adaptados. En cambio, “Estoy Contigo” fue desarrollada tomando en cuenta el contexto lingüístico, social y emocional de estudiantes peruanos, logrando mayor vinculación emocional y compromiso por parte del usuario.

En resumen, el desarrollo de “Estoy Contigo” integró correctamente aspectos técnicos, funcionales y sociales. La aplicación representa un aporte valioso en el campo de la salud mental preventiva, especialmente en entornos académicos vulnerables. Su diseño modular, el uso de herramientas actuales (Firebase, Android Studio) y la adaptación cultural del contenido garantizan no solo su efectividad, sino también su sostenibilidad y aceptación entre los usuarios objetivos.

# **COMPARACIONES**

El desarrollo y aplicación de la herramienta “Estoy Contigo” demostró ser efectiva para mejorar el acompañamiento emocional de los estudiantes universitarios, al proporcionar recursos accesibles, motivacionales y adaptados al contexto peruano. Este resultado coincidió con lo reportado por Linardon et al. (2024), quienes identificaron efectos estadísticamente significativos en la reducción de ansiedad y depresión mediante apps móviles que integran elementos como el monitoreo emocional y la terapia cognitivo-conductual.

Asimismo, los resultados del proyecto se alinearon con los hallazgos de Bress et al. (2024), quienes evidenciaron que las intervenciones con aplicaciones basadas en TCC disminuyeron notablemente los síntomas de ansiedad en jóvenes adultos, especialmente cuando incluían interacciones personalizadas y apoyo social. En la aplicación “Estoy Contigo”, el uso de un chatbot empático y la posibilidad de acceder a contenido emocionalmente conectado aportaron beneficios similares.

Anivel nacional, los resultados del proyecto también fueron comparables con los obtenidos por Pacora Chávez y Bezada Sánchez (2023), quienes desarrollaron una app y cápsula acústica para estudiantes con síntomas de depresión. Ambos estudios concluyeron que la personalización del contenido y el diseño centrado en el usuario fueron clave para la aceptación y eficacia de la herramienta, especialmente en contextos universitarios.

Por otro lado, en contraste con aplicaciones extranjeras analizadas por Gómez-Gonzales et al. (2022), que carecieron de adaptación cultural y no lograron un vínculo emocional sólido con el usuario, la app “Estoy Contigo” superó esta limitación al diseñarse específicamente para el contexto sociocultural de los estudiantes peruanos, mejorando la usabilidad, pertinencia y compromiso emocional.

En síntesis, el presente proyecto confirmó los hallazgos de la literatura internacional y nacional, y además aportó un valor añadido al incorporar características culturales y lingüísticas locales que aumentaron su efectividad como herramienta preventiva y de acompañamiento emocional.

# **CONCLUSIONES**

La aplicación móvil “Estoy Contigo” constituye una herramienta innovadora ya que es muy funcional para el acompañamiento emocional de estudiantes universitarios, especialmente en contextos donde el acceso a atención psicológica es limitado, por ello, su enfoque preventivo el cual es basado en tecnologías accesibles y recursos interactivos, permite a los jóvenes gestionar su bienestar mental los estudiantes universitarios.

El proyecto tiene un enfoque contextualizado a la realidad peruana, lo que representa una de sus principales fortalezas que muy poco se ven ya que a diferencia de otras aplicaciones extranjeras que no consideran las particularidades culturales y emocionales de los estudiantes locales, debido a esto esta app se construye con base en sus necesidades específicas, costumbres, idioma y entorno para poder mejorar la satisfacción del usuario o del universitario en si.

Desde el punto de vista técnico, la aplicación demuestra un desarrollo sólido, utilizando tecnologías modernas como Firebase y Android Studio, lo que asegura eficiencia, escalabilidad al contar con una interfaz intuitiva para el usuario, además, cuenta con la implementación de un modelo MVC y una base de datos NoSQL en tiempo real para poder así garantizar una experiencia fluida, segura y personalizada.

Las funcionalidades implementadas abordan diferentes dimensiones del bienestar psicológico, como el test emocional, la posibilidad de programar citas con psicólogos, el uso de un chatbot empático y el acceso a contenido motivador de manera la cual esta herramienta no solo ayuda a los estudiantes sino también identifica su estado emocional para luego buscar soluciones óptimas del caso.

# **REFERENCIAS BIBLIOGRÁFICAS**

Alvarado Loaiza, G., Rodríguez, M., & Chacón, L. (2024). Intervenciones digitales en el manejo de la ansiedad en jóvenes universitarios. Revista Latinoamericana de Psicología Digital, 8(1), 45-60.

Franco-Carrero, P., Martínez, A., & Guzmán, J. (2023). Aplicaciones móviles como herramientas para el afrontamiento emocional en contextos académicos. Psicología y Educación, 29(2), 113-128.

Muñoz, R. F., Leykin, Y., Barrera, A. Z., & Pérez-Stable, E. J. (2021). Digital interventions for mental health: From innovation to implementation. Clinical Psychology: Science and Practice, 28(3), e12345.

Organización Mundial de la Salud (OMS). (2022). Salud mental de los adolescentes y jóvenes: datos y cifras. https://www.who.int/es/news-room/fact-sheets/detail/mental-health-of-adolescents

Valencia-Ortiz, J., & Hernández, D. (2022). Aceptación de tecnologías móviles para el bienestar emocional en estudiantes universitarios. Revista de Psicología y Tecnología, 5(2), 91-104.

Organización Mundial de la Salud (OMS). La salud mental de los adolescentes [en línea]. 2024. [Consultado el 22 de abril de 2025]. Disponible en:

<https://www.who.int/es/news-room/fact-sheets/detail/adolescent-mental-health>

The American Institute of Stress. Stress in Teens and Young Adults. 2025. [Consultado el 22 de abril de 2025].

Disponible en: <https://www.stress.org/who-gets-stressed/teens-young-adults/>

Organización Mundial de la Salud (OMS). La salud mental de los adolescentes [en línea]. 2024. [Consultado el 22 de abril de 2025]. Disponible en: <https://www.who.int/news-room/fact-sheets/detail/mental-health-at-work>

BANTJES, Jason et al. The mental health of university students in South Africa: Results of the national student survey. Journal of Affective Disorders [en línea]. 2023 [consultado el 24 de abril de 2025]. ISSN 0165-0327. Disponible en: <https://doi.org/10.1016/j.jad.2022.10.044>

BARRERA-HERERRA, Ana y Yusting SAN MARTÍN. Prevalencia de Sintomatología de Salud Mental y Hábitos de Salud en una Muestra de Universitarios Chilenos. Psykhe (Santiago) [en línea]. 2021, 30(1) [consultado el 24 de abril de 2025]. Disponible en: <https://doi.org/10.7764/psykhe.2019.21813>

Minedu y el Minsa trabajan con 21 universidades públicas en el cuidado de la salud mental. (s. f.). Noticias - Ministerio de Educación - Plataforma del Estado Peruano. 2019. Disponible en:

<https://www.gob.pe/institucion/minedu/noticias/52741-minedu-y-el-minsa-trabajan-con-21-universidades-publicas-en-el-cuidado-de-la-salud-mental>

GÓMEZ-GONZALES, Roxana; RAMÍREZ, Mariana; PEÑA, Lucía, 2022. Análisis de la usabilidad y efectividad de apps de salud mental en jóvenes latinoamericanos. Revista Latinoamericana de Psicología Digital. [Consulta: 21 abr. 2025].

Pacora Chávez, P. M., & Bezada Sánchez, A. C. (2023). Cápsula acústica y aplicación móvil para impulsar la inclusión educativa de los estudiantes con depresión en universidades de Lima Metropolitana [Trabajo de investigación de bachillerato, Escuela de Educación Superior Tecnológica Toulouse Lautrec]. <https://repositorio.tls.edu.pe/handle/20.500.12826/1020>

Linardon, Jake; Torous, John; Firth, Joseph; Cuijpers, Pim; Messer, Mariel; Fuller-Tyszkiewicz, Matthew. *Current evidence on the efficacy of mental health smartphone apps for symptoms of depression and anxiety. A meta-analysis of 176 randomized controlled trials*. *World Psychiatry* [en línea]. 2024, vol. 23, no. 1, pp. 139–149. [Consulta: 24 abr. 2025]. Disponible en:<https://onlinelibrary.wiley.com/doi/full/10.1002/wps.21183>

Linardon, Jake; Cuijpers, Pim; Carlbring, Per; Messer, Melanie; Fuller-Tyszkiewicz, Matthew. *The efficacy of app-supported smartphone interventions for mental health problems: A meta-analysis of randomized controlled trials*. *World Psychiatry* [en línea]. 2019, vol. 18, no. 3, pp. 325–336. [Consulta: 21 abr. 2025]. Disponible en:<https://doi.org/10.1002/wps.20673>

Bress, Jennifer N.; Falk, Avital; Schier, Maddy M.; Jaywant, Abhishek; Moroney, Elizabeth; Dargis, Monika; Bennett, Shannon M.; Scult, Matthew A.; Volpp, Kevin G.; Asch, David A.; Balachandran, Mohan; Perlis, Roy H.; Lee, Francis S.; Gunning, Faith M. *Efficacy of a Mobile App-Based Intervention for Young Adults With Anxiety Disorders: A Randomized Clinical Trial*. *JAMA Network Open* [en línea]. 2024, vol. 7, no. 8, e2428372. [Consulta: 24 abr. 2025]. Disponible en:<https://jamanetwork.com/journals/jamanetworkopen/fullarticle/2822451>

Alvarez Suel, Diana Elizabet; Flores Quispe, Anggie Anthuane; Lamas Hernández, Valeria Belén; Liñán Morales, Karen Inés; Shibuya Panduro, Sofía de Fátima. *Aplicación móvil Mente Sana*. Proyecto de grado, Universidad Peruana de Ciencias Aplicadas, 2021. Disponible en:[https://repositorioacademico.upc.edu.pe/handle/10757/656841​](https://repositorioacademico.upc.edu.pe/handle/10757/656841%E2%80%8B)

Adauto Ortiz, Rodrigo Alexander. *Diseño e implementación de un aplicativo móvil de atención e información para una entidad pública prestadora de servicios de salud mental en personas con trastornos mentales de 18 años a 59 años en Lima, Perú*. Tesis de bachiller, Pontificia Universidad Católica del Perú, 2022. Disponible en:<http://hdl.handle.net/20.500.12404/23896>

ALVARADO LOAIZA, Carlos; MARTÍNEZ, Andrea, 2024. Emotions&Care: Impacto de una app para el seguimiento emocional de estudiantes universitarios. Revista de Psicología Aplicada. [Consulta: 21 abr. 2025].

FRANCO-CARRERO, Gabriela; TORRES, Samuel, 2023. Uso de Headspace en la reducción del estrés académico en estudiantes de medicina. Journal of Mental Health Innovation. [Consulta: 21 abr. 2025].

CHÁVEZ GUTIÉRREZ, Mónica Alejandra, 2023. Uso de una app de frases motivadoras para fortalecer la autoestima en jóvenes universitarios. Tesis de licenciatura. Cusco: Universidad Nacional de San Antonio Abad del Cusco. [Consulta: 21 abr. 2025]. <https://repositorio.uap.edu.pe/jspui/bitstream/20.500.12990/13275/1/Tesis_autoestima_rendimiento_acad%C3%A9mico_estudiantes_facultad_ingenier%C3%ADa_civil_universidad_nacional_SanLuis%20Gonzaga_Ica%20.pdf>

HERNÁNDEZ-APARCANA, Julio; MENDOZA, Luis; GARCÍA, Diego, 2025. Evaluación del impacto de las aplicaciones móviles en el bienestar emocional de estudiantes universitarios peruanos. Psicología Educativa y Social. [Consulta: 21 abr. 2025]. <https://www.cienciaspedagogicas.rimed.cu/index.php/ICCP/article/download/565/656/5506>