



**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR**

**Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC**

**APLICATIVO MÓVIL PARA LA GESTIÓN DE HÁBITOS SALUDABLES EN  
ANDROID**

**Proyecto de titulación previo a la obtención del título de:**

**Tecnóloga en Desarrollo de Software**

**Autor: Sara María Piedra Garzón**

**Tutor: Billy Andrés Navarrete Garzón**

**Quito, Ecuador**

**2025**

## **Dedicatoria**

Dedico este esfuerzo a mi familia, mis amigos y a todos los profesores que me apoyaron en este camino, por su confianza y motivación constante.

## Tabla de contenidos

Dedicatoria .....	2
Lista de tablas.....	4
Lista de figuras .....	5
1. Introducción .....	8
2. Planteamiento del problema .....	8
3. Objetivos .....	9
3.1. Objetivo General: .....	9
3.2. Objetivos Específicos:.....	9
4. Levantamiento de Requisitos y Diseño del Sistema.....	10
4.1. Requisitos funcionales:.....	10
4.2. Requisitos no funcionales .....	15
4.3. Diseño de Sistema: .....	16
5. Capítulo II .....	18
5.1. Construcción del Sistema .....	18
5.1.1. Estructura del Proyecto Android.....	18
5.1.2. Registro y Autenticación de Usuarios .....	19
5.1.3. Selección y Visualización del Personaje Motivacional .....	19
5.1.4. Gestión de Hábitos por Categoría .....	20
5.1.5. Sistema de Tareas del Usuario .....	21
5.1.6. Sistema de Notificaciones .....	22
5.1.7. Gestión de Cuenta.....	23
5.1.8. Diseño de Interfaz e Interacciones.....	25
5.1.9. Control de Orientación y Compatibilidad.....	26
5.1.10. Conexión al Backend y Seguridad .....	31
6. Capítulo III .....	32
6.1. Pruebas y Estabilización .....	33
6.1.1. Pruebas realizadas en Postman.....	33
7. Conclusiones.....	40
8. Recomendaciones .....	41
9. Referencias bibliográficas .....	42

## Lista de tablas

Tabla 1	Tabla de los personajes .....	26
Tabla 2	Tabla de Usuarios .....	27
Tabla 3	Tabla de las Tareas .....	27
Tabla 4	Tabla de Tareas Recurrentes .....	28
Tabla 5	Tabla de las tareas del Usuario.....	28
Tabla 6	Tabla de las Notificaciones .....	28
Tabla 7	Tabla para la conexión de NotificacionesUsuario .....	29
Tabla 8	Tabla de las emociones del personaje.....	29
Tabla 9	Tabla de Progreso diario del usuario .....	30
Tabla 10	Datos de .env para conexión .....	38

## Lista de figuras

Figura 1 MockUp del Login y Sign Up .....	11
Figura 2 MockUp de Ajustes.....	11
Figura 3 MockUp Selección de P. ....	12
Figura 4 MockUp Categoría Hábitos.....	13
Figura 5 MockUp de Home.....	14
Figura 6 MockUp Tareas Usuario .....	15
Figura 7 Diagrama general de casos de uso del usuario .....	17
Figura 8 Diagrama Sencillo de la base de datos .....	17
Figura 9 Inicio de Sesión y Registro .....	19
Figura 10 Categoría Hábitos .....	21
Figura 11 Tareas del usuario .....	22
Figura 12 Notificación de tareas .....	23
Figura 13 Config. Usuario .....	23
Figura 14 Edición de cuenta usuario .....	24
Figura 15 Cambio y actualización del personaje .....	25
Figura 16 Página principal .....	25
Figura 17 Diagrama de la Base de Datos.....	31
Figura 18 Conexiones Backend.....	31
Figura 19 Contraseña de los usuarios encriptadas .....	32
Figura 20 Prueba de Crear Usuario .....	34
Figura 21 Prueba de inicio de sesión.....	35
Figura 22 Prueba de consulta de tareas de la semana.....	36
Figura 23 Corrección zona horaria en el backend.....	36
Figura 24 Prueba de cambio de emoción del personaje.....	37
Figura 25 Datos insertados en Railway .....	38
Figura 26 Comprobante de conexión en Railway.....	39

## DECLARACIÓN y AUTORIZACIÓN

Yo, **Sara María Piedra Garzón** con C.I. 1726356049 autor(a) del trabajo de integración curricular intitulado: “**Aplicativo Móvil para la gestión de hábitos saludables en Android**”, previa a la obtención del título de **TECNOLOGIA SUPERIOR EN DESARROLLO DE SOFTWARE** en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, agosto 2025



Sara María Piedra Garzón

C.I. 1726356049

## **Agradecimientos**

Quiero expresar mi más sincero agradecimiento a mi familia, por su apoyo incondicional a lo largo de mi carrera. Toda su confianza, amor y palabras de aliento han sido fundamentales en cada paso de este duro camino.

A mis profesores, gracias por compartir sus conocimientos y poner su dedicación constante en mí. Cada lección ha contribuido significativamente a mi formación profesional y personal.

También deseo agradecer a mis compañeros Christopher y Antonella, con quienes trabajé en el proyecto que sentó las bases de esta tesis, ganador de un concurso en la universidad. Su colaboración y compromiso fueron claves en aquel momento, y hoy me siento honrada de dar continuidad a ese esfuerzo a través de este trabajo.

## **1. Introducción**

Este proyecto de titulación es una aplicación web interactiva diseñada para fomentar hábitos saludables en jóvenes mediante una experiencia lúdica y personalizada. A través de un personaje virtual que refleja el estado emocional y físico del usuario según sus acciones diarias, la plataforma motiva a mantener un equilibrio en áreas clave como la alimentación, la actividad física, el sueño y la salud emocional.

La aplicación incorpora funciones como listas de tareas personalizables, recompensas visuales (como fondos de pantalla), notificaciones programadas y elementos de juego que refuerzan la constancia y el autocuidado. Este proyecto no solo busca crear hábitos positivos, sino también acompañar al usuario en su proceso de crecimiento personal, promoviendo la conciencia y la reflexión diaria.

## **2. Planteamiento del problema**

En la actualidad, se ha visto la ausencia de herramientas accesibles y eficientes para poder llevar al cabo la gestión de los hábitos saludables; sobre todo para alumnos y trabajadores, en ambientes académicos y laborales de alto estrés, llegando al punto del descuido alimenticio, físico y emocional las más comunes.

Muchos estudiantes y profesionales no desarrollan itinerarios que les ayuden a organizar de manera efectiva el tiempo y las rutinas que les posibilitarían mantener los hábitos que les favorecen desde el punto de vista del bienestar físico y emocional; del contrario, el resultado será la aparición de problemas como el sedentarismo, la fatiga mental o la mala alimentación. A pesar de la creciente oferta que existe sobre aplicaciones de bienestar, las cuales prometen ser una alternativa saludable, muchas de ellas ni siquiera logran ajustarse a los requerimientos de los usuarios o carecen de una

efectiva integración con las costumbres diarias, pues llegan a ser demasiado ordinarias o con alta complejidad.

### **3. Objetivos**

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

Desarrollar una aplicación móvil interactiva que permita al usuario registrar, organizar y monitorear sus hábitos saludables de forma personalizada, motivado por un personaje animado, con base en las recomendaciones de organismos especializados como la Organización Mundial de la Salud (OMS), la Organización Panamericana de la Salud (OPS) y el Ministerio de Salud Pública del Ecuador (MSP), con el fin de mejorar su calidad de vida y bienestar integral.

#### **3.2. Objetivos Específicos:**

- Diseñar e implementar un sistema de autenticación y gestión de datos personalizados para el usuario.
- Integrar un personaje animado que reaccione emocionalmente a las acciones del usuario como elemento motivacional.
- Permitir la planificación, seguimiento y notificación de hábitos saludables diarios, mostrando métricas de progreso y mensajes motivacionales.

## Capítulo I

### 4. Levantamiento de Requisitos y Diseño del Sistema

En esta fase, se realiza un análisis de los requisitos funcionales y no funcionales del sistema de la aplicación HarmonyReflect, una aplicación destinada a promover hábitos saludables a través de una interfaz lúdica e interactiva. El objetivo de la movilización de requisitos fue apreciar las necesidades de los usuarios y señalar los bloques de construcción críticos del sistema.

Los requisitos se recopilaron a través de revisiones de la aplicación y análisis de tendencias en salud. Se identificaron cuatro categorías principales de hábitos a desarrollar: alimentación, actividad física, salud emocional y calidad del sueño.

#### 4.1.Requisitos funcionales:

##### 4.1.1. Registro y autenticación de usuarios:

El sistema debe permitir el registro mediante correo electrónico, nombre de usuario y contraseña. Los usuarios autenticados, después de autenticar si correo electrónico y contraseña, podrán acceder a sus datos personalizados de hábitos, personaje y progreso.

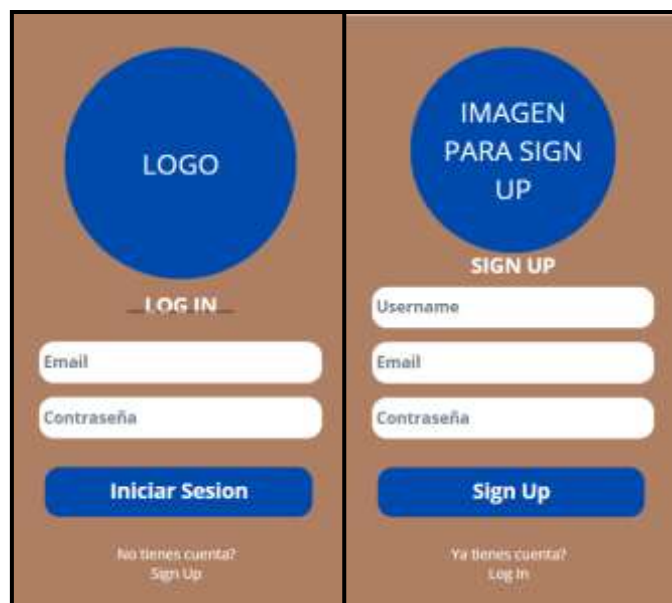


Figura 1 MockUp del Login y Sign Up

Nota. Captura de pantalla del sistema desarrollado en MockUp (2025)

#### 4.1.2. Actualización de datos del usuario:

El usuario puede cambiar su contraseña y nombre de usuario a través de la página que se tiene para Ajustes.



Figura 2 MockUp de Ajustes

Nota: Captura de pantalla del sistema desarrollado en MockUp (2025)

#### 4.1.3. Selección de personaje motivacional:

El usuario puede seleccionar entre diferentes personajes animados, los cuales reaccionan emocionalmente en función del progreso del usuario.



Figura 3 MockUp Selección de P.

Nota: Captura de pantalla del sistema desarrollado en MockUp (2025)

#### 4.1.4. Gestión de hábitos divididos por categoría:

El usuario puede consultar y marcar hábitos en las categorías:

- Alimentación, ejem: beber agua, comer frutas.
- Ejercicio, ejem: caminar, hacer estiramientos.
- Salud emocional, ejem: escribir en diario, escuchar música relajante.
- Sueño, ejem: irse a dormir temprano.



Figura 4 MockUp Categoría Hábitos

Nota: Captura de pantalla del sistema desarrollado en MockUp (2025)

#### 4.1.5. Visualización del estado emocional del personaje:

El personaje reacciona con expresiones felices o tristes según el cumplimiento de hábitos.

#### 4.1.6. Sistema de notificaciones push:

Notifica al usuario sobre tareas pendientes, el estado emocional del personaje, y frases motivacionales diarias.

#### 4.1.7. Cambio dinámico del fondo según hora y personaje:

El fondo del entorno varía visualmente entre mañana, tarde y noche, y se adapta al personaje elegido.

#### 4.1.8. Módulo de frases motivacionales (implementación en proceso):

Se muestra una frase diferente cada día para motivar al usuario.



Figura 5 MockUp de Home

Nota: Captura de pantalla del sistema desarrollado en MockUp (2025)

#### 4.1.9. Sistema de tareas como completa:

El usuario puede marcar una tarea como completada en cualquiera de las categorías de hábitos. Esta acción actualiza el progreso diario del usuario y afecta la expresión emocional del personaje motivacional.



Figura 6 MockUp Tareas Usuario

Nota: Captura de pantalla del sistema desarrollado en MockUp (2025)

#### 4.1.10. Sistema de progreso:

El usuario puede revisar su progreso diario, en una lista cada semana, que muestra las tareas registradas y las tareas realizadas, viendo un porcentaje de toda la semana.

## 4.2.Requisitos no funcionales

### 4.2.1. Usabilidad:

La interfaz debe ser intuitiva y amigable para usuarios entre 12 y 30 años.

### 4.2.2. Portabilidad:

Compatible con dispositivos Android 7.0 o superior.

#### 4.2.3. Escalabilidad:

La arquitectura debe permitir agregar nuevas funcionalidades como minijuegos o más personajes en futuras versiones.

#### 4.2.4. Rendimiento:

El tiempo de respuesta para cualquier acción no debe superar 2 segundos en condiciones normales.

#### 4.2.5. Seguridad:

Las credenciales del usuario se deben almacenar cifradas. Los datos personales se manejan de acuerdo con la normativa vigente de protección de datos.

### **4.3. Diseño de Sistema:**

A nivel de diseño, se utilizó XML para modelar casos de uso, clases y flujos de navegación. Además, se definió una estructura de base de datos relacional en MYSQL, complementada. Se optó por una arquitectura frontend en Android Studio con Java y backend basado en servicios REST, conectado a una API Node.js en la nube (Railway).

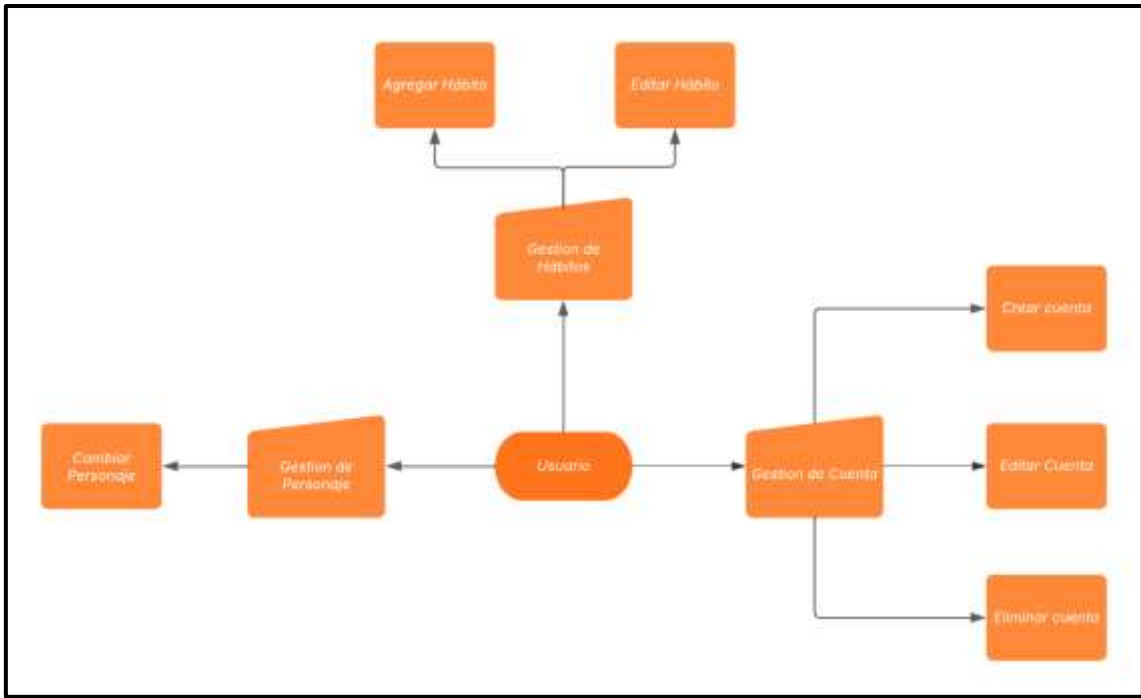


Figura 7 Diagrama general de casos de uso del usuario

Nota: Captura de pantalla de draw.io (2025)

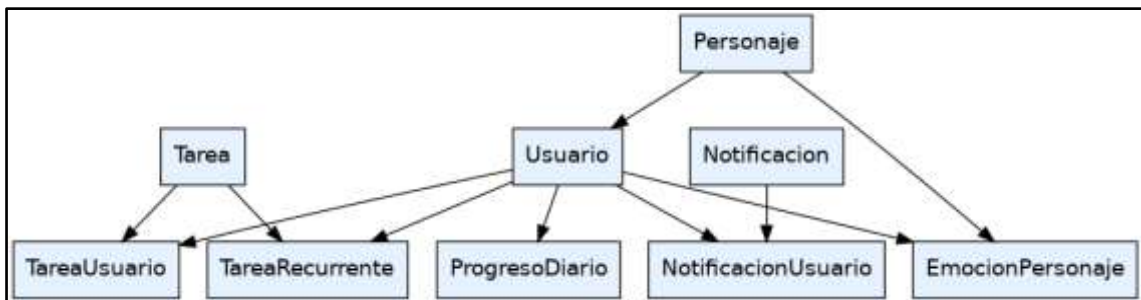


Figura 8 Diagrama Sencillo de la base de datos

Nota: Captura de pantalla de diagrama en Word (2025)

## **5. Capítulo II**

### **5.1.Construcción del Sistema**

La interfaz HarmonyReflect fue desarrollada usando la arquitectura cliente-servidor, donde la interfaz de usuario fue realizada en Android Studio, utilizando Java, al igual que el servidor que fue construido usando Node.js utilizando express. El backend estaba expuesto usando servicios REST que estaban alojados en la nube (RailWay). En esta etapa se realizaba la integración de cada uno de los módulos en cada uno de los pasos definidos en la etapa de requisitos funcionales.

#### **5.1.1. Estructura del Proyecto Android**

La estructura de la app fue mejorada por el uso del patrón de diseño MVVM, es decir Modelo-Vista-VistaModelo. La navegación fue organizada en forma de BottomNavigationViewsActivity que permite al usuario navegar a las secciones más importantes de la app: Inicio (Donde se encuentra un botón flotante para la configuración del usuario), Su Rutina de Hábitos, Opciones de Tareas, y Progreso.

La comunicación entre los fragments y el backend se lleva a cabo mediante la librería Retrofit, que proporciona una forma sencilla y segura para consumir los endpoints REST. Para serializar la respuesta de tipo JSON a objetos Java, se utilizó GsonConverterFactory , que permite mapear los datos recibidos a estructuras de datos nativas, simplificando el manejo de la información dentro de la aplicación. (Garcia, 2020)

### 5.1.2. Registro y Autenticación de Usuarios

Se creó una interfaz de registro e inicio de sesión con validaciones locales (formato de correo, nombre de usuario y longitud de contraseña). Una vez autenticado, el usuario accede a su entorno personalizado, donde se cargan sus datos desde la API.

Las credenciales se almacenan de manera segura utilizando SharedPreferences cifradas. El token de sesión, proporcionado por el servidor, permite autenticación persistente en segundo plano. Esta estrategia mejora significativamente la experiencia del usuario, asegurando la entrega oportuna de recordatorios y el correcto seguimiento de hábitos, sin necesidad de interacción constante con la aplicación. (Android Developers, 2024)

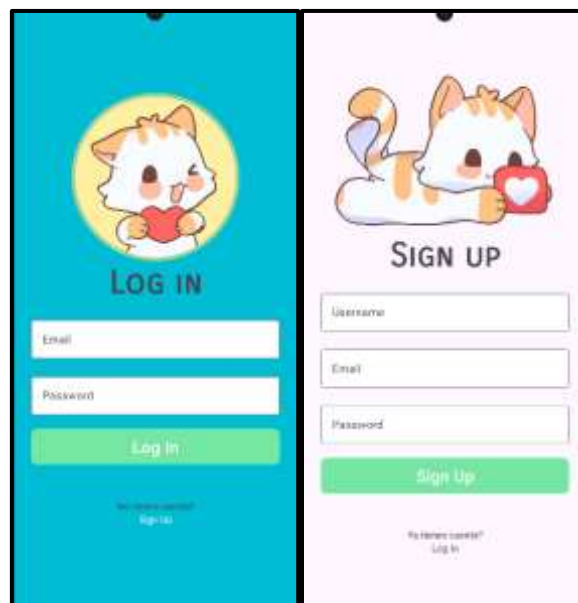


Figura 9 Inicio de Sesión y Registro

Nota: Captura de pantalla de la aplicación en Android Studio (2025)

### 5.1.3. Selección y Visualización del Personaje Motivacional

El sistema permite seleccionar entre varios personajes animados. Cada personaje tiene diferentes expresiones gráficas (feliz, triste, motivado), que se actualizan dinámicamente según el cumplimiento de hábitos, Para que esta actualización ocurra incluso cuando la aplicación no está en primer plano, se utiliza WorkManager, un componente de Android que permite ejecutar tareas en segundo plano de manera confiable. (Sindkar, 2022)

El personaje se muestra en un entorno visual que cambia según el personaje seleccionado, Con la intención de dar algo único de cada uno.

#### **5.1.4. Gestión de Hábitos por Categoría**

Los hábitos se representan como listas divididas en cuatro categorías principales: alimentación, ejercicio, salud emocional y sueño. Para separar visualmente las tareas, se utilizaron dos componentes CardView: uno para identificar las categorías y otro para mostrar las tareas almacenadas por categoría. En ambos casos, se implementó un RecyclerView para desplegar los elementos en pantalla de forma dinámica y eficiente. (AndroidDevelopers, 2025) (Android Developers, 2025)

Al marcar un hábito como realizado, se registra en el servidor mediante una petición POST. Guardando al usuario el hábito seleccionado, hora de realización y los días de la semana que se desea para el hábito.



Figura 10 Categoría Hábitos

*Nota: Captura de pantalla de la aplicación en Android Studio (2025)*

### 5.1.5. Sistema de Tareas del Usuario

Los hábitos del usuario serán registrados semanalmente, mostrando en pantalla una lista dividida en dos categorías: Tareas de Hoy y Tareas de Mañana. Solo las tareas del día actual podrán marcarse como realizadas mediante un checkbox, y una vez completadas, se eliminarán de la lista mostrada en pantalla.



Figura 11 Tareas del usuario

*Nota: Captura de pantalla de la aplicación en Android Studio (2025)*

### 5.1.6. Sistema de Notificaciones

Se integró un sistema de notificaciones basado en programación con AlarmManager, enlazado a datos almacenados en MYSQL en el backend. Las notificaciones alertan sobre:

- Hábitos pendientes.
- Frases motivacionales.
- Cambios en el estado emocional del personaje.

Se asegura que las notificaciones sean personalizadas para cada usuario según sus configuraciones. Además, utiliza WorkManager para revisar constantemente las notificaciones de las tareas en segundo plano, permitiendo actualizaciones y su funcionamiento sin la necesidad de que el usuario tenga que entrar a la aplicación para su uso. (Sindkar, 2022)

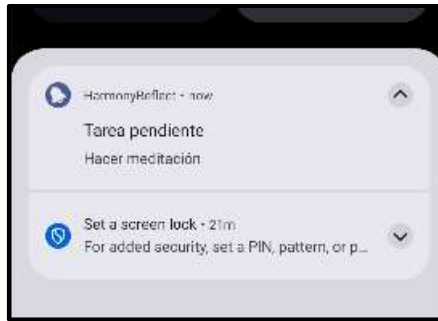


Figura 12 Notificación de tareas

Nota: Captura de pantalla de la aplicación en Android Studio (2025)

### 5.1.7. Gestión de Cuenta

El usuario puede gestionar su cuenta, modificando su contraseña y/o su nombre de usuario. Los cambios se reflejan de forma inmediata mediante una notificación emergente, tipo Toast, en la parte inferior de la pantalla, tras lo cual se cierra la vista de edición y vuelve a la pantalla principal. (Android Developers, 2024)

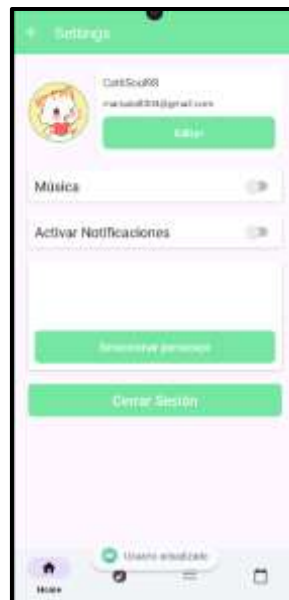


Figura 13 Config. Usuario

Nota: Captura de pantalla de la aplicación en Android Studio (2025)



Figura 14 Edición de cuenta usuario

Nota: Captura de pantalla de la aplicación en Android Studio (2025)

Además, el usuario puede navegar por los diferentes avatares disponibles y elegir el que desee. Una vez se confirme el cambio, este se almacena de forma inmediata en el dispositivo mediante SharedPreferences y se persiste en el servidor mediante una llamada REST, mostrando una notificación emergente, nuevamente tipo Toast, que informa del éxito de la operación. A continuación, la vista de selección se cierra y la interfaz principal se actualiza para reflejar el nuevo personaje y su habitación correspondiente. (Android Developers, 2024)

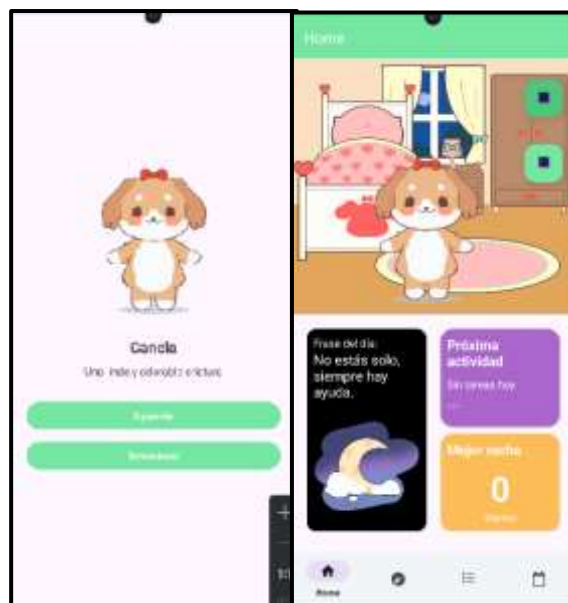


Figura 15 Cambio y actualización del personaje

Nota: Captura de pantalla de la aplicación en Android Studio (2025)

### 5.1.8. Diseño de Interfaz e Interacciones

El diseño de interfaz fue realizado en XML, utilizando ConstraintLayout y LinearLayout para organizar los elementos visuales. Se aplicaron CardViews para destacar las secciones y mejorar la estética. (Android Developers, 2024)

La pantalla principal se muestra de forma vertical, mostrando al personaje, próxima tarea a realizar y una frase por día para motivar (seleccionada de forma al azar). Además, una imagen que cambia dependiendo del horario local: amanecer, mediodía y noche.



Figura 16 Página principal

Nota: Captura de pantalla de la aplicación en Android Studio (2025)

Se aplicaron animaciones leves con `ViewPropertyAnimator` y se integró una paleta de colores amigable con el usuario joven, siguiendo criterios de accesibilidad visual.

### 5.1.9. Control de Orientación y Compatibilidad

Para asegurar una experiencia consistente, se restringió la rotación de pantalla a modo vertical desde el `AndroidManifest.xml`, definiendo `android:screenOrientation="portrait"` para todas las actividades. (Android Developers, 2024)

La app fue diseñada para ser compatible con dispositivos Android 7.0 en adelante, optimizando rendimiento mediante el uso de imágenes comprimidas y control de memoria en cargas de `ImageView`.

### 5.1.10. Base de datos

La base de datos fue creada mediante `MYSQL Workbench`, permitiendo un manejo adecuado de datos y sus diferentes entidades. Siendo subida a `RailWay` para su despliegue en la nube.

<b>Tabla Personaje</b>		
<b>Dato</b>	<b>Tipo de Dato</b>	<b>Restricciones</b>
idPersonaje	INT	AUTO_INCREMENT, PRIMARY KEY
nombre_personaje	VARCHAR(100)	NOT NULL
descripcion	VARCHAR(255)	

*Tabla 1 Tabla de los personajes*

*Nota: Tabla basada en el diagrama simple (2025)*

<b>Tabla Usuario</b>		
<b>Dato</b>	<b>Tipo de Dato</b>	<b>Restricciones</b>
idUsuario	INT	AUTO_INCREMENT, PRIMARY KEY
nombre	VARCHAR(100)	NOT NULL
email	VARCHAR(100)	UNIQUE, NOT NULL
password	VARCHAR(100)	NOT NULL
fecha_registro	DATE	NOT NULL
idPersonaje	INT	FOREIGN KEY → Personaje(idPersonaje), ON DELETE SET NULL

*Tabla 2 Tabla de Usuarios*

*Nota: Tabla basada en el diagrama simple (2025)*

<b>Tabla Tarea</b>		
<b>Dato</b>	<b>Tipo de Dato</b>	<b>Restricciones</b>
idTarea	INT	AUTO_INCREMENT, PRIMARY KEY
nombre_tarea	VARCHAR(100)	NOT NULL
categoria	ENUM('Alimentación','Ejercicio','Sueño','SaludEmocional')	NOT NULL
activa	BOOLEAN	DEFAULT TRUE

*Tabla 3 Tabla de las Tareas*

*Nota: Tabla basada en el diagrama simple (2025)*

<b>Tabla TareaRecurrente</b>		
<b>Dato</b>	<b>Tipo de Dato</b>	<b>Restricciones</b>
idTareaRecurrente	INT	AUTO_INCREMENT, PRIMARY KEY
idUsuario	INT	FOREIGN KEY → Usuario(idUsuario), ON DELETE CASCADE

idTarea	INT	FOREIGN KEY → Tarea(idTarea), ON DELETE CASCADE
hora	TIME	NOT NULL
lunes	BOOLEAN	DEFAULT FALSE
martes	BOOLEAN	DEFAULT FALSE
miercoles	BOOLEAN	DEFAULT FALSE
jueves	BOOLEAN	DEFAULT FALSE
viernes	BOOLEAN	DEFAULT FALSE
sabado	BOOLEAN	DEFAULT FALSE
domingo	BOOLEAN	DEFAULT FALSE

Tabla 4 Tabla de Tareas Recurrentes

Nota: Tabla basada en el diagrama simple (2025)

<b>Tabla TareaUsuario</b>		
<b>Dato</b>	<b>Tipo de Dato</b>	<b>Restricciones</b>
idTareaUsuario	INT	AUTO_INCREMENT, PRIMARY KEY
idUsuario	INT	FOREIGN KEY → Usuario(idUsuario), ON DELETE CASCADE
idTarea	INT	FOREIGN KEY → Tarea(idTarea), ON DELETE CASCADE
fecha	DATE	NOT NULL
realizada	BOOLEAN	DEFAULT FALSE
hora_realizacion	TIME	

Tabla 5 Tabla de las tareas del Usuario

Nota: Tabla basada en el diagrama simple (2025)

<b>Tabla Notificacion</b>		
<b>Dato</b>	<b>Tipo de Dato</b>	<b>Restricciones</b>
idNotificacion	INT	AUTO_INCREMENT, PRIMARY KEY
mensaje	VARCHAR(255)	NOT NULL
hora_envio	TIME	NOT NULL
tipo	ENUM('RecordatorioTarea','Motivacional')	NOT NULL
recurrente	BOOLEAN	DEFAULT TRUE
activa	BOOLEAN	DEFAULT TRUE

Tabla 6 Tabla de las Notificaciones

Nota: Tabla basada en el diagrama simple (2025)

<b>Tabla NotificacionUsuario</b>		
<b>Dato</b>	<b>Tipo de Dato</b>	<b>Restricciones</b>
idNotificacionUsuario	INT	AUTO_INCREMENT, PRIMARY KEY
idUsuario	INT	FOREIGN KEY → Usuario(idUsuario), ON DELETE CASCADE
idNotificacion	INT	FOREIGN KEY → Notificacion(idNotificacion), ON DELETE CASCADE
fecha_programada	DATE	NOT NULL

Tabla 7 Tabla para la conexión de NotificacionesUsuario

Nota: Tabla basada en el diagrama simple (2025)

<b>Tabla EmocionPersonaje</b>		
<b>Dato</b>	<b>Tipo de Dato</b>	<b>Restricciones</b>
idEmocion	INT	AUTO_INCREMENT, PRIMARY KEY
idUsuario	INT	FOREIGN KEY → Usuario(idUsuario), ON DELETE CASCADE
idPersonaje	INT	FOREIGN KEY → Personaje(idPersona je), ON DELETE CASCADE
fecha	DATE	NOT NULL
emocion	ENUM('Feliz','Triste','Neutro','Enojado','Motivado')	NOT NULL
causa	VARCHAR(255)	
unique_emocion_por_dia	-	UNIQUE (idUsuario, fecha)

Tabla 8 Tabla de las emociones del personaje

Nota: Tabla basada en el diagrama simple (2025)

<b>Tabla ProgresoDiario</b>		
<b>Dato</b>	<b>Tipo de Dato</b>	<b>Restricciones</b>
idProgresoDiario	INT	AUTO_INCREMENT, PRIMARY KEY
idUsuario	INT	FOREIGN KEY → Usuario(idUsuario),

		ON DELETE CASCADE
fecha	DATE	NOT NULL, UNIQUE (idUserio, fecha)
tareas_realizadas	INT	DEFAULT 0
tareas_asignadas	INT	DEFAULT 0
porcentaje_cumplimiento	DECIMAL(5,2)	GENERATED ALWAYS AS ((tareas_realizadas / tareas_asignadas) * 100) STORED

Tabla 9 Tabla de Progreso diario del usuario

Nota: Tabla basada en el diagrama simple (2025)

Aquí se mostrarán las diferentes tablas creadas en nuestra base de datos:

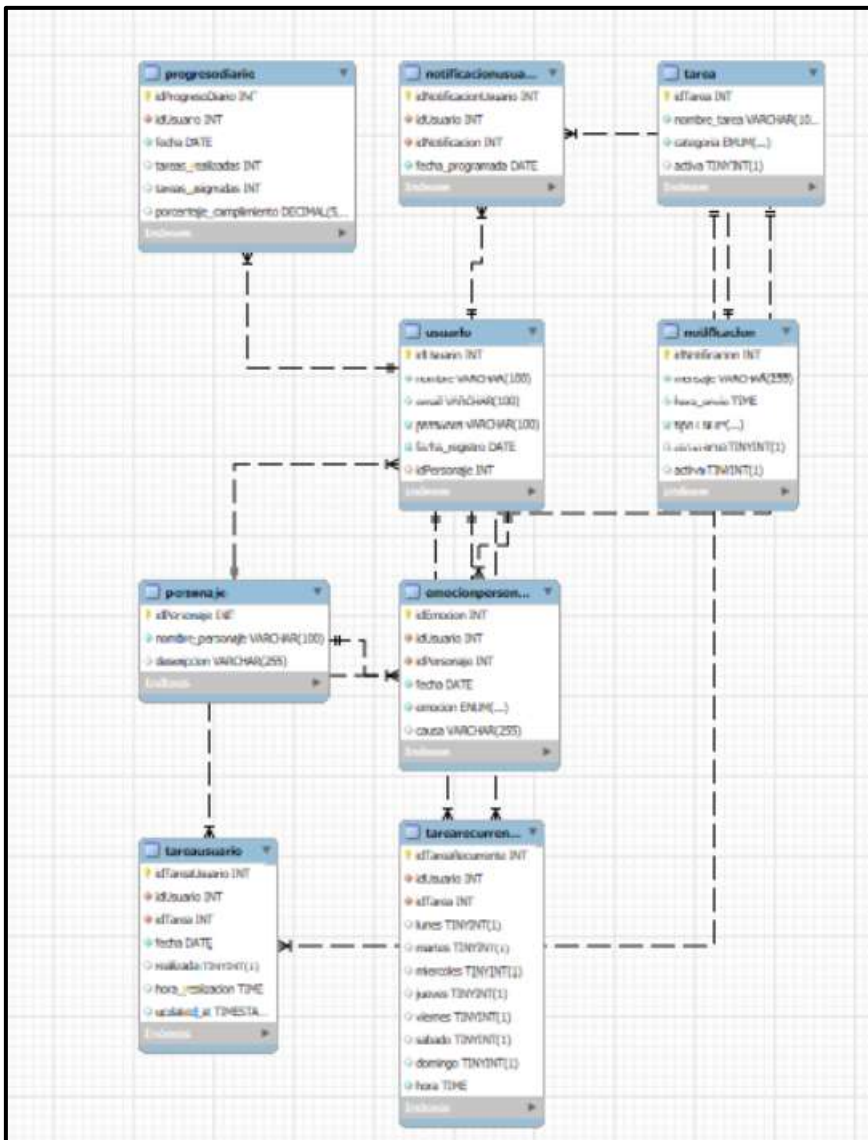


Figura 17 Diagrama de la Base de Datos

Nota: Captura de pantalla del diagrama en MYSQL Workbench (2025)

### 5.1.11. Conexión al Backend y Seguridad

La conexión al backend se realiza vía HTTP o HTTPS dependiendo del entorno. Debido a que, en pruebas locales se usa la IP del servidor local (10.0.2.2 para emulador y la IP LAN para dispositivos reales). (Android Developers, 2024)

```
public class RetrofitAdapter {
    private static Retrofit retrofit = null; //usages

    public static Retrofit getClient() {
        if (retrofit == null) {
            retrofit = new Retrofit.Builder()
                .baseUrl("http://10.0.2.2:3000/") // IP para emulador Android
                // .baseUrl("http://192.168.0.34:3000/") // Si usas dispositivo real
                // .baseUrl("https://alibabacloud.com.ec/") // Si ya está desplegado en la nube
                .addConverterFactory(GsonConverterFactory.create())
                .build();
        }
        return retrofit;
    }
}
```

Figura 18 Conexiones Backend

Nota: Captura de pantalla de la aplicación en Android Studio (2025)

La integración con el backend se realiza mediante el uso de la librería Retrofit, una herramienta robusta y ampliamente utilizada en Android para la gestión de solicitudes HTTP. Esta librería facilita la comunicación cliente-servidor al permitir el manejo eficiente de peticiones y respuestas en formato JSON. (Square, 2024)

Para producción, se utiliza un dominio público con cifrado SSL. El backend implementa control de acceso por token, validación de entradas y encriptación de contraseñas mediante bcrypt permitiendo que las contraseñas sean seguras y no puedan visualizarse en la base de datos. (Provos, 2020)

idUsuario	nombre	email	password	fecha_registro	idPersonaje
3	MarySue34	mariaorangexd@gmail.com	\$2b\$10\$O2HfV5urP7AWm94x2e/9d./XQx7FueI...	2025-06-26	2
4	Tonoto004	sakuripandita@gmail.com	\$2b\$10\$h865uhP0.2g0U7VASQ7YEOmXBH0anP...	2025-06-26	1
11	SaritaP	saritapiedrag@gmail.com	\$2b\$10\$YAJar2mesclZZM47N16INO8yG5.UR0a...	2025-06-27	1
17	Pandita Kawaii	elenita.piedrag@gmail.com	\$2b\$10\$UqavYtKRzDq4cxjzUGCb/ue6eTz0AFO...	2025-06-27	1
19	CattiSoul98	mariadoll004@gmail.com	\$2b\$10\$ilIvy.QpK1otYRDmgcsHHOSxGuUdVb5e...	2025-07-28	2
NULL	NULL	NULL	NULL	NULL	NULL

Figura 19 Contraseña de los usuarios encriptadas

Nota: Captura de pantalla de la aplicación en Android Studio (2025)

## 6. Capítulo III

## **6.1.Pruebas y Estabilización**

Tras completar el desarrollo de los módulos principales de la aplicación, se lleva a cabo las pruebas para su estabilización, con el propósito de validar el correcto funcionamiento de cada componente y asegurar una experiencia de usuario estable en los distintos dispositivos. Tomando en cuenta que la aplicación puede ser instalada por medio de un apk.

### **6.1.1. Pruebas realizadas en Postman.**

Postman es una plataforma integral que permite trabajar con APIs. Facilita el desarrollo, las pruebas y la documentación de las diferentes APIs creadas, mediante solicitudes HTTP, permitiendo analizar su respuesta.

Se realizaron las primeras pruebas de integración de la API en esta plataforma con el objetivo de comprobar la correcta conexión de los endpoints Rest.

### **6.1.2. Crear Usuario:**

Al momento de crear usuario, se revisó que las contraseñas queden cifradas en la base de datos para mayor protección. Y el sistema se encarga de llenar al usuario con el

personaje predeterminado (Michu), con su respecta emoción inicial (feliz).

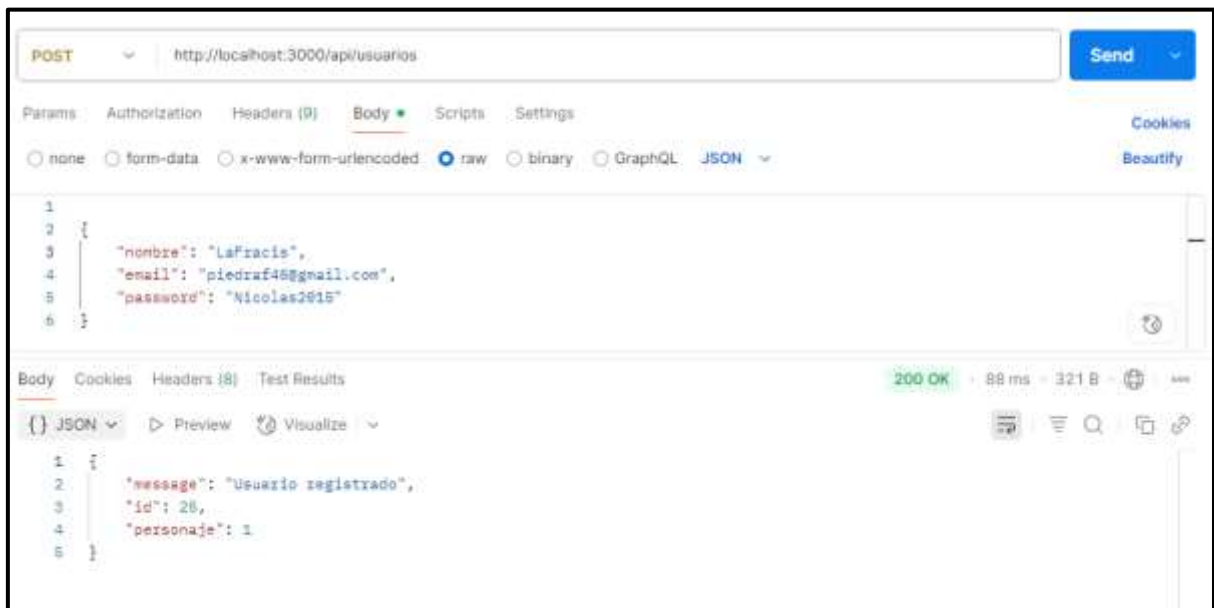


Figura 20 Prueba de Crear Usuario

Nota: Captura de pantalla de prueba en postman (2025)

### 6.1.3. Inicio de Sesión:

La prueba de inicio de sesión fue un éxito, llegando a crear el token debido para el uso de su funcionalidad. Además, de enviar el id del usuario para cargar la información en la aplicación.



toLocaleDateString('sv-SE') para sincronizar la hora en formato UTC con la hora local.

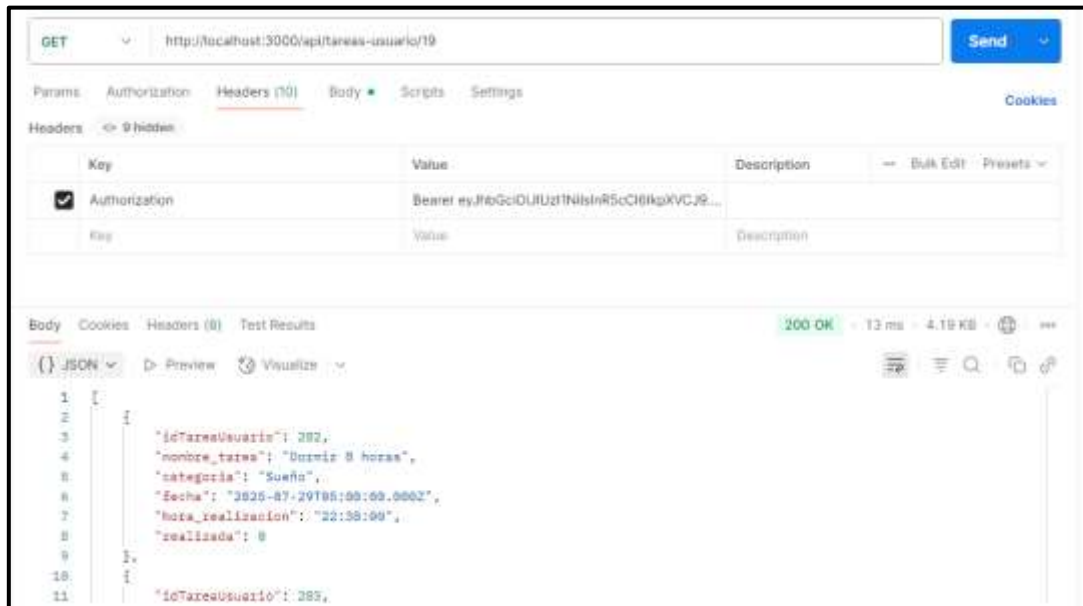


Figura 22 Prueba de consulta de tareas de la semana

Nota: Captura de pantalla de prueba en postman (2025)

```
// Actualizar tarea recurrente y regenerar toda la semana
const updateTareaRecurrente = (req, res) => {
  const { idTareaRecurrente } = req.params;
  const { idUsuario, idTarea, hora, dias } = req.body;
  const diasSemana = ['domingo', 'lunes', 'martes', 'miercoles', 'jueves', 'viernes', 'sabado'];
  const today = new Date();
  const fechaHoy = today.toLocaleDateString('sv-SE');
```

Figura 23 Corrección zona horaria en el backend

Nota: Captura de pantalla de prueba en postman (2025)

### 6.1.5. Carga de personaje:

Permite al usuario actualizar las emociones de su personaje, además de actualizar su personaje a otro de su preferencia, siendo actualmente Michu y Canela los personajes para seleccionar.

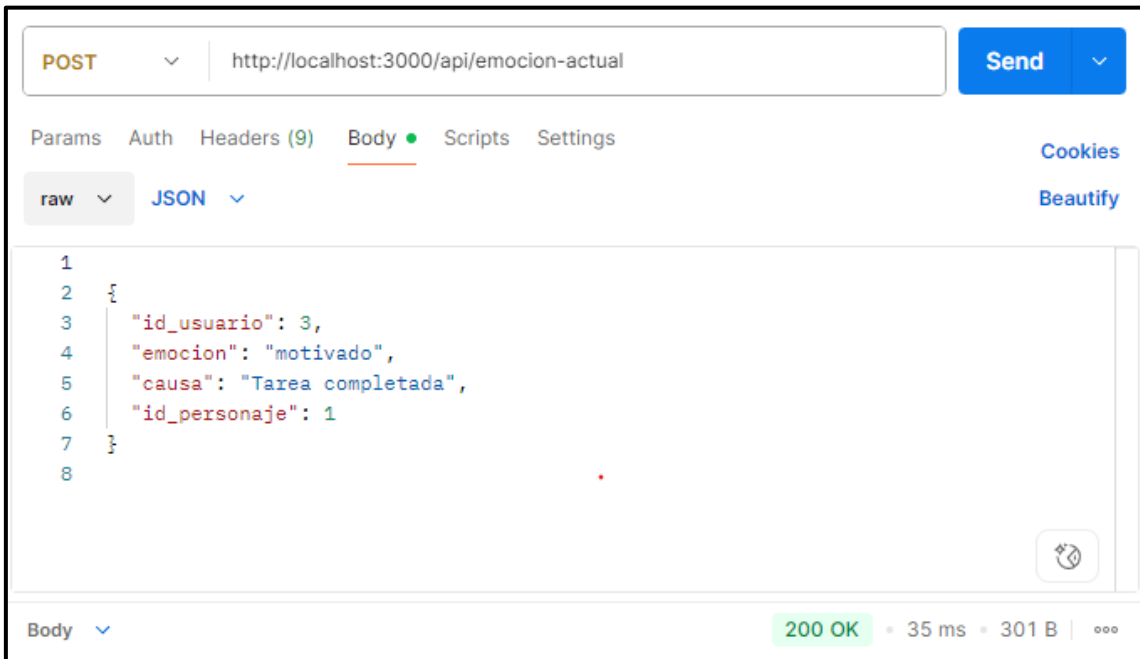


Figura 24 Prueba de cambio de emoción del personaje

Nota: Captura de pantalla de prueba en postman (2025)

## 6.2. Conexión de Railway:

Railway es una plataforma en la nube que facilita el despliegue de aplicaciones y servicios sin necesidad de una configuración compleja de servidores. Permite a desarrolladores subir proyectos (backend, bases de datos, APIs, etc.) y luego ser desplegarlos. Así que, para la integración de la API y la base de datos, se tuvo que hacer pequeños cambios.

Se modifica el archivo `.env` para establecer la conexión con la base de datos desplegada en Railway, utilizando los parámetros proporcionados por la plataforma. En este caso, el archivo queda configurado de la siguiente manera:

<b>DB_HOST</b>	<b>shortline.proxy.rlwy.net</b>
<b>DB_PORT</b>	27049
<b>DB_USER</b>	root

<b>DB_PASS</b>	*****
<b>DB_NAME</b>	railway
<b>PORT</b>	3000

Tabla 10 Datos de .env para conexión

Posteriormente, el proyecto fue almacenado en un repositorio en GitHub y desplegado en Railway, configurando las mismas variables de la tabla en la sección Variables de la plataforma.

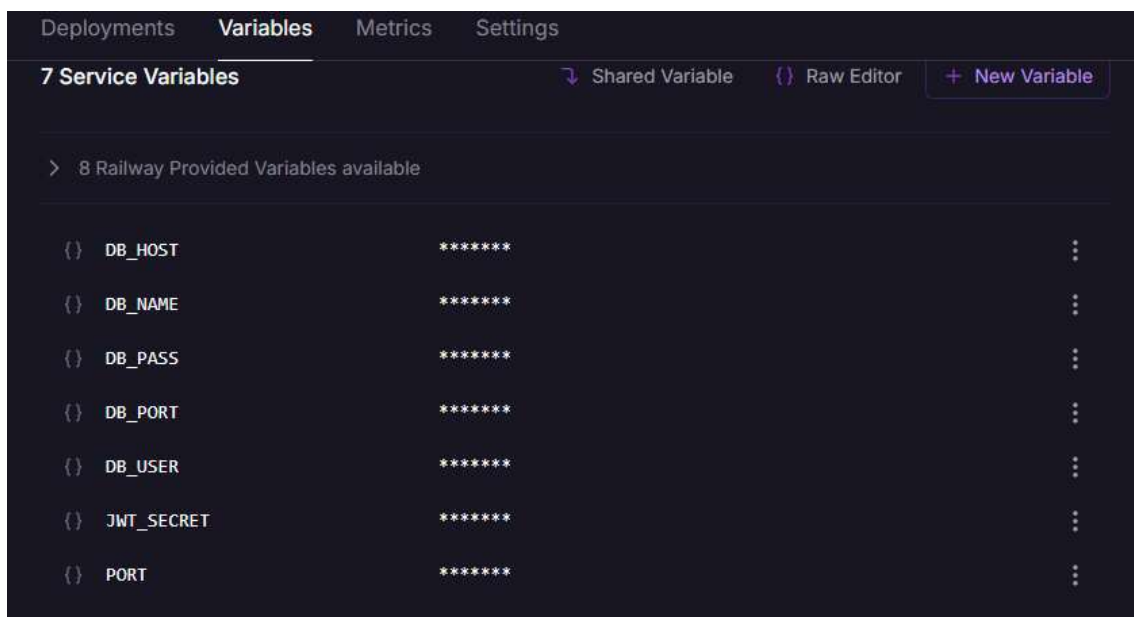
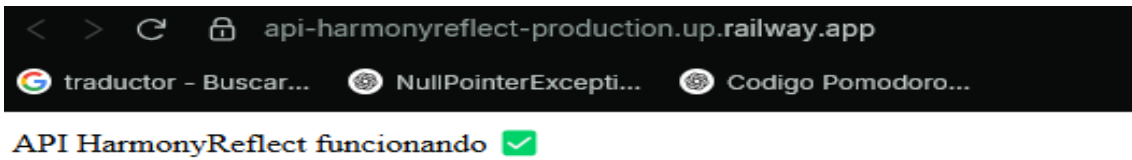


Figura 25 Datos insertados en Railway

Nota: Captura de pantalla de prueba en Railway (2025)



*Figura 26 Comprobante de conexión en Railway*

*Nota: Captura de pantalla de prueba de conexión (2025)*

## **7. Conclusiones**

El desarrollo de la aplicación HarmonyReflect permite cumplir satisfactoriamente con el objetivo general planteado, al crear una herramienta interactiva y personalizada que registra, organiza y monitorea hábitos saludables, utilizando un personaje animado como elemento motivacional. A través de la integración de un sistema de autenticación seguro, la gestión de datos personalizados y la implementación de notificaciones y métricas de progreso, se logra proporcionar al usuario un entorno digital intuitivo y atractivo.

Además del entorno personalizado y visual con el que el usuario puede interactuar, el personaje animado brinda a los usuarios la oportunidad de incentivar al usuario a comprometerse mejor con hábitos saludables. Asimismo, la programación adecuada de las tareas del usuario y la sincronización fluida de datos y tiempo crean una experiencia amigable para el usuario.

Para concluir, HarmonyReflect cumple con las expectativas de experiencia del usuario al proporcionar apoyo emocional y motivacional más allá de las simples listas de tareas. Además, muestra su efectividad comprobada para fomentar hábitos saludables, cumpliendo con los objetivos específicos definidos dentro del marco de un diseño centrado en el usuario.

## 8. Recomendaciones

- Desarrollar módulos independientes para que las actualizaciones de uno no requieran reinstalar o modificar todo el sistema, reduciendo el tiempo de despliegue y facilitando el mantenimiento.
- Integrar sistemas de monitoreo y almacenamiento de logs para detectar rápidamente fallos después del despliegue, evitando largos tiempos de diagnóstico.
- Realizar una aplicación o página web para administradores con el fin de dar facilidad para la integración de nuevos personajes, escenarios, tareas. Y mucho más importante, administración de los usuarios.
- En caso de que el número de personajes, animaciones y escenarios aumente significativamente, se recomienda migrar estos recursos a un servicio de almacenamiento en la nube. Con el fin de, reducir el tamaño de la aplicación instalada en el dispositivo y mejorar la velocidad de actualización.
- Se debe considerar a futuro la posibilidad de subir la aplicación a la plataforma Play Store u similares, para la facilidad de actualizaciones de la aplicación.

## 9. Referencias bibliográficas

- Android Developers. (2024, julio 26). *Background work*. Android Developers: <https://developer.android.com/guide/topics/data/data-storage/shared-preferences>
- Android Developers. (2024, diciembre 20). *Desarrollar una interfaz de usuario con vistas*. Android Developers: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwiAnKzhsdaOAXeTTABHSP6GhQQFnoECBsQAQ&url=https%3A%2F%2Ftranslate.google.com%2Ftranslate%3Fu%3Dhttps%3A%2F%2Fdeveloper.android.com%2Fstudio%2Fwrite%2Flayout-editor%26hl%3Des%26sl%3De>
- Android Developers. (2024). *Notifications overview*. Android Developers: <https://developer.android.com/guide/topics/ui/notifiers/toasts>
- Android Developers. (2024). *Run Apps on the Android Emulator*. Android Developers: <https://developer.android.com/studio/run/emulator-networking>
- Android Developers. (2024, junio). *Support different display sizes*. Android Developers: [https://developer.android.com/guide/practices/screens\\_support?hl=es-419](https://developer.android.com/guide/practices/screens_support?hl=es-419)
- Android Developers. (2025, mayo 27). *Cómo crear listas dinámicas con RecyclerView*. Android Developers: <https://developer.android.com/develop/ui/views/layout/recyclerview?hl=es-419>
- Android Developers. (2025, mayo 21). *Como programar alarmas*. Android Developers: [https://developer.android.com/develop/background-work/services/alarms/schedule?hl=es-419#:~:text=Alarms%20\(based%20on%20the%20AlarmManager,timers%20or%20continuously%20running%20services](https://developer.android.com/develop/background-work/services/alarms/schedule?hl=es-419#:~:text=Alarms%20(based%20on%20the%20AlarmManager,timers%20or%20continuously%20running%20services).
- Android Developers. (2025, febrero 10). *Programacion de tareas con WorkManager*. Android Developer: [https://developer.android.com.translate.google/topic/libraries/architecture/workmanager?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=es&\\_x\\_tr\\_hl=es&\\_x\\_tr\\_pto=tc](https://developer.android.com.translate.google/topic/libraries/architecture/workmanager?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)
- AndroidDevelopers. (2025, mayo 15). *CardView*. AndroidDevelopers: <https://developer.android.com/reference/androidx/cardview/widget/CardView>
- Cursos Android ANT. (2024, mayo 30). *Curso Retrofit: Cómo consumir web services de un API en Android Studio y Kotlin(API REST/ENDPOINT)*. Youtube: [https://www.youtube.com/watch?v=xm0GhiLD\\_gw](https://www.youtube.com/watch?v=xm0GhiLD_gw)
- Garcia, A. (2020, julio 7). *¿Cómo consumir una API Rest desde android?* EDteam: <https://ed.team/blog/como-consumir-una-api-rest-desde-android>
- Provos, N. (2020). *bcrypt password hashing for your software and your servers*. Openwall: <https://www.openwall.com/crypt>

Sindkar, P. (16 de Marzo de 2022). *Tutorial de Android WorkManager: Primeros pasos*. flexiple:  
[https://flexiple-com.translate.goog/android/android-workmanager-tutorial-getting-started?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=es&\\_x\\_tr\\_hl=es&\\_x\\_tr\\_pto=tc&\\_x\\_tr\\_hist=true](https://flexiple-com.translate.goog/android/android-workmanager-tutorial-getting-started?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc&_x_tr_hist=true)

Square. (2024). *Run Apps on the Android Emulator*. SquareGitHub:  
<https://square.github.io/retrofit>

## **Anexos**

Enlace a Drive donde se guarda el Manual de Usuario:

[https://docs.google.com/document/d/14wM-S5ffNdeC3BPyQTpB9Yjcd9LUn\\_46/edit?usp=sharing&oid=117381692771526728890&rtpof=true&sd=true](https://docs.google.com/document/d/14wM-S5ffNdeC3BPyQTpB9Yjcd9LUn_46/edit?usp=sharing&oid=117381692771526728890&rtpof=true&sd=true)