



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**SISTEMA MÓVIL ENFOCADO A COMBATIR
LA DEPRESIÓN Y SUS CONSECUENCIAS
A TRAVÉS DE LA IMPLEMENTACIÓN DE
UN BOT Y LA REPRESENTACIÓN DE
DATOS ESTADÍSTICOS**

TESIS

Que para obtener el título de
INGENIERO EN COMPUTACIÓN

P R E S E N T A

Enrique Soto Vázquez

DIRECTORA DE TESIS

M.I. Norma Elva Chávez Rodríguez



Ciudad Universitaria, Cd. Mx., 2018

DESGLOSE DE CONTENIDO.

CAPITULO 1. INTRODUCCIÓN.

1. Introducción.

CAPITULO 2. PLANTEAMIENTO DEL PROBLEMA.

2. Planteamiento del problema.

CAPITULO 3. MARCO TEÓRICO.

3. Marco Teórico.
 - 3.1 Depresión.
 - 3.1.1 Tipos de depresión.
 - 3.1.2 Tratamientos contra la depresión.
 - 3.1.3 La depresión en México y el Mundo.
 - 3.2 Bases de datos.
 - 3.2.1 Tipos de bases de datos.
 - 3.2.2 Firebase Realtime Database.
 - 3.3 Dispositivos móviles inteligentes.
 - 3.3.1 Aplicaciones móviles.
 - 3.3.2 Android.
 - 3.4 Teoría de Bots.
 - 3.5 Facebook Messenger.

CAPÍTULO 4. CONGRUENCIA METODOLÓGICA.

4. Congruencia metodológica.
 - 4.1 Matriz metodológica.
 - 4.2 Modelo de desarrollo.
 - 4.3 Etapas del modelo RAD.
 - 4.3.1 Requerimientos.
 - 4.3.2 Ciclo iterativo del prototipo.
 - 4.3.3 Cutover.

CAPÍTULO 5. DISEÑO E IMPLEMENTACIÓN.

5. Diseño e implementación.
 - 5.1 Planteamiento.
 - 5.2 Casos de uso.
 - 5.2.1 Casos de uso del usuario.
 - 5.2.2 Casos de uso del Bot.
 - 5.2.3 Casos de uso de la aplicación.
 - 5.3 Elaboración.
 - 5.3.1 Flujo lógico del Bot.
 - 5.3.2 Webhook.
 - 5.3.3 Programación del Bot.

- 5.3.4 Mensajes del Bot.
- 5.3.5 Funciones especiales del Bot.
- 5.3.6 Base de datos Firebase.
- 5.3.7 Programación de la aplicación Android.
- 5.3.8 Generación de gráficas.

CAPÍTULO 6. PRUEBAS.

- 6. Pruebas.
 - 6.1 Pruebas con el Bot.
 - 6.1.1 Interacción del Bot con el usuario.
 - 6.2 Pruebas con la base de datos.
 - 6.3 Pruebas con la aplicación móvil Android.
 - 6.3.1 Vistas horizontales.

CAPÍTULO 7. RESULTADOS.

- 7. Resultados.
 - 7.1 Resultados del Bot.
 - 7.1.1 Retroalimentación del usuario.
 - 7.2 Resultados de la aplicación móvil.
 - 7.3 Resultados del usuario.

CAPÍTULO 8. CONCLUSIONES.

- 8. Conclusiones.
 - 8.1 Objetivo planteado.
 - 8.2 Desarrollo.
 - 8.3 Implementación.
 - 8.4 Innovación.

BIBLIOGRAFÍA.

ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN.....	2
CAPÍTULO 2. PLANTEAMIENTO DEL PROBLEMA.	6
CAPÍTULO 3. MARCO TEÓRICO.....	10
3.1 DEPRESIÓN.....	10
3.1.1 TIPOS DE DEPRESIÓN.....	13
3.1.2 TRATAMIENTOS CONTRA LA DEPRESIÓN.	19
3.1.3 LA DEPRESIÓN EN MÉXICO Y EL MUNDO.	23
3.2 BASES DE DATOS.....	25
3.2.1 TIPOS DE BASES DE DATOS.....	26
3.2.2 FIREBASE REALTIME DATABASE.	29
3.3 DISPOSITIVOS MÓVILES INTELIGENTES.	31
3.3.1 APLICACIONES MÓVILES.	32
3.3.2 ANDROID.	35
3.4 TEORÍA DE BOTS.....	36
3.5 FACEBOOK MESSENGER.	38
CAPÍTULO 4. CONGRUENCIA METODOLÓGICA.	41
4.1 MATRIZ METODOLÓGICA.	42
4.2 MODELO DE DESARROLLO.	44
4.3 ETAPAS DEL MODELO RAD.....	46
4.3.1 REQUERIMIENTOS.	46
4.3.2 CICLO ITERATIVO DEL PROTOTIPO.....	54
4.3.3 CUTOVER.	57
CAPÍTULO 5. DISEÑO E IMPLEMENTACIÓN.	59
5.1 PLANTEAMIENTO.....	59
5.2 CASOS DE USO.....	61
5.2.1 CASOS DE USO DEL USUARIO.	62
5.2.2 CASOS DE USO DEL BOT.	74
5.2.3 CASOS DE USO DE LA APLICACIÓN.....	81
5.3 ELABORACIÓN.	84
5.3.1 FLUJO LÓGICO DEL BOT.	84
5.3.2 WEBHOOK.....	87

5.3.3 PROGRAMACIÓN DEL BOT.....	94
5.3.4 MENSAJES DEL BOT.	102
5.3.5 FUNCIONES ESPECIALES DEL BOT.	110
5.3.6 BASE DE DATOS FIREBASE.	122
5.3.7 PROGRAMACIÓN DE LA APLICACIÓN ANDROID.....	127
5.3.8 GENERACIÓN DE GRÁFICAS.....	139
CAPÍTULO 6. PRUEBAS.	143
6.1 PRUEBAS CON EL BOT.	143
6.1.1 INTERACCIÓN DEL BOT CON EL USUARIO.	145
6.2 PRUEBAS CON LA BASE DE DATOS.....	156
6.3 PRUEBAS CON LA APLICACIÓN MÓVIL ANDROID.	160
6.3.1 VISTAS HORIZONTALES.	171
CAPÍTULO 7. RESULTADOS	177
7.1 RESULTADOS DEL BOT.	177
7.1.1 RETROALIMENTACIÓN DEL USUARIO.	180
7.2 RESULTADOS DE LA APLICACIÓN MÓVIL.....	182
7.3 RESULTADOS DEL USUARIO.....	184
CAPÍTULO 8. CONCLUSIONES.....	187
8.1 OBJETIVO PLANTEADO.....	187
8.2 DESARROLLO.....	188
8.3 IMPLEMENTACIÓN.....	189
8.4 INNOVACIÓN.	190
BIBLIOGRAFÍA.	191

ORGANIZACIÓN FIGURAS Y TABLAS.

CAPITULO 1. INTRODUCCIÓN.

CAPITULO 2. PLANTEAMIENTO DEL PROBLEMA.

CAPITULO 3. MARCO TEÓRICO.

Figura 3.1 Comparación de la actividad entre un cerebro normal y el de un paciente diagnosticado con depresión.

Figura 3.2 Gráfica del número de personas deprimidas en México y la frecuencia con la que se deprimen.

Figura 3.3 Gráfica del número de muertes ocasionadas por un trastorno mental entre 2011 y 2015.

Tabla 3.1 Funciones Firebase Realtime Database.

Figura 3.4 Cuota de mercado de los Sistemas Operativos Móviles a nivel mundial.

CAPÍTULO 4. CONGRUENCIA METODOLÓGICA.

Tabla 4.1 Matriz metodológica.

Figura 4.1 Modelo ágil tipo Desarrollo Rápido de Aplicaciones a utilizar.

Tabla 4.2 Requerimientos generales de la aplicación.

Tabla 4.3 Requerimientos generales de la aplicación.

Tabla 4.4 Etapas del ciclo iterativo.

CAPÍTULO 5. DISEÑO E IMPLEMENTACIÓN.

Figura 5.1 Diagrama general de casos de uso.

Tabla 5.1 Caso de uso (usuario): Interacción con el Bot.

Tabla 5.2 Caso de uso (usuario): Proporcionar consejo.

Tabla 5.3 Caso de uso (usuario): Calificar consejo.

Tabla 5.4 Caso de uso (usuario): Consultar consejo.

Tabla 5.5 Caso de uso (usuario): Generar datos.

Tabla 5.6 Caso de uso (usuario): Uso de la aplicación móvil.

Tabla 5.7 Caso de uso (usuario): Consulta de estadísticas globales.

Tabla 5.8 Caso de uso (usuario): Consulta de estadísticas personales.

Tabla 5.9 Caso de uso (Bot): Gestión de datos generados por usuario.

Tabla 5.10 Caso de uso (Bot): Gestión de consejos.

Tabla 5.11 Caso de uso (Bot): Obtención de datos personales.

Tabla 5.12 Caso de uso (Bot): Interacción con el usuario.

Tabla 5.13 Caso de uso (Bot): Vincular usuarios.

Tabla 5.14 Caso de uso (Bot): Mostrar consejo.

Tabla 5.15 Caso de uso (Bot): Proporcionar información de ayuda.

Tabla 5.16 Caso de uso (App): Despliega información estadística en gráficas.

Tabla 5.17 Caso de uso (App): Despliega información estadística en gráficas.

Figura 5.2 Diagrama de flujo para funcionamiento del Bot.

Figura 5.3 Página de Facebook entidad del Bot.

Figura 5.4 Suscripción del “webhook”.

Figura 5.5 Generación del token de acceso.

Figura 5.6 Estructuración tipo JSON de los datos.

Figura 5.7 Diagrama de flujo para actividades Android.

CAPÍTULO 6. PRUEBAS.

Figura 6.1 Pantalla que muestra la ejecución de Ngrok.

Figura 6.2 Pantalla de suscripción del “webhook”.

Figura 6.3 Pantalla de ejecución del Bot.

Figura 6.4 Interacción con el Bot en Messenger.

Figura 6.5 Interacción con el Bot en Messenger.

Figura 6.6 Interacción con el Bot en Messenger.

Figura 6.7 Interacción con el Bot en Messenger.

Figura 6.8 Interacción con el Bot en Messenger.

Figura 6.9 Interacción con el Bot en Messenger.

Figura 6.10 Interacción con el Bot en Messenger.

Figura 6.11 Interacción con el Bot en Messenger.

Figura 6.12 Interacción con el Bot en Messenger.

Figura 6.13 Interacción con el Bot en Messenger.

Figura 6.14 Interacción con el Bot en Messenger.

Figura 6.15 Inserción de usuarios.

Figura 6.16 Inserción de datos globales.

Figura 6.17 Inserción de consejos.

Figura 6.18 Gráficas Firebase.

Figura 6.19 Características del dispositivo móvil de pruebas.

Figura 6.20 Icono de la aplicación móvil.

Figura 6.21 Aplicación Móvil Android.

Figura 6.22 Aplicación Móvil Android.

Figura 6.23 Aplicación Móvil Android.

Figura 6.24 Aplicación Móvil Android.

Figura 6.25 Aplicación Móvil Android.

Figura 6.26 Aplicación Móvil Android.

Figura 6.27 Aplicación Móvil Android.

Figura 6.28 Aplicación Móvil Android.

Figura 6.29 Aplicación Móvil Android.

Figura 6.30 Navegación horizontal.

Figura 6.31 Navegación horizontal.

Figura 6.32 Navegación horizontal.

Figura 6.33 Navegación horizontal.

Figura 6.34 Navegación horizontal.

Figura 6.35 Navegación horizontal.

Figura 6.36 Navegación horizontal.

Figura 6.37 Navegación horizontal.

CAPÍTULO 7. RESULTADOS.

Figura 7.1 Implementación en plataforma Heroku.

Figura 7.2 Suscripción del webhook a Heroku.

Tabla 7.1 Retroalimentación para el Bot.

Tabla 7.2 Retroalimentación para la aplicación móvil.

Tabla 7.3 Retroalimentación para la aplicación móvil.

CAPÍTULO 8. CONCLUSIONES.

CAPÍTULO 1

INTRODUCCIÓN

CAPÍTULO 1. INTRODUCCIÓN.

El mundo del desarrollo de software actual posee un ritmo de crecimiento acelerado y más aun con el constante emerger de nuevas tecnologías como los dispositivos móviles, que consideran celulares inteligentes, tabletas, computadoras de alta tecnología, relojes inteligentes entre varios otros más por mencionar. Es por eso por lo que a nivel global el mundo se encuentra íntimamente relacionado con las tecnologías de la información y comunicación (TIC), así como el Internet y las diferentes plataformas conectadas en red que permiten la interacción entre usuarios, la difusión libre de grandes cantidades de información y una constante innovación en las interfaces de software con las que tiene contacto él usuario. Debido a todo esto y al mundo globalizado que permite la fácil adquisición de algún dispositivo inteligente, es que la mayoría de nosotros podemos tener en nuestras manos una llave que nos brinda acceso a un mundo digital de posibilidades ilimitadas.

En lo general los diferentes programas, software y aplicaciones con las que contamos en la actualidad nos permiten gestionar información de carácter relevante, así como su almacenamiento para poder realizar un análisis posterior y de este modo poder obtener múltiples soluciones a diferentes problemas que se presentan día a día con el avance de la historia humana.

Este tipo de tecnologías traen consigo un sinnúmero de aplicaciones benéficas para la sociedad, ya sea dentro del ámbito educativo, investigación con fines científicos, desarrollo en robótica, la administración pública y gubernamental, así como para las empresas en desarrollo o para procesos de gran industria.

Sin embargo, es de destacar que en tiempos recientes es la tecnología de los teléfonos inteligentes la que ha representado un cambio sustancial en la vida cotidiana de las personas, ya que estos dispositivos ofrecen características y capacidades similares a las de una computadora personal, pero todo esto al tamaño de una mano. Siendo así entonces que los dispositivos móviles nos ayudan a realizar tareas que van desde enviar un correo electrónico, hasta redactar

documentos, elaborar presentaciones, administrar nuestras tareas, o cuidar de nuestra salud.

El mercado de aplicaciones móviles es sumamente extenso y nos ofrece un número incontable de aplicaciones que se actualiza día con día, abarcando aplicaciones sencillas como prender un pequeño led, hasta plataformas enormes de comunicación y servicios en red. Es por eso que debemos reconocer la importancia que tienen estas tecnologías en la actualidad y aprender cómo usarlas en nuestro beneficio para aportar algún avance que satisfaga alguna necesidad de nuestra comunidad.

El presente documento busca demostrar la relevancia de las aplicaciones móviles, mediante la implementación de tecnología emergente como lo son los Bots, para combatir uno de los grandes problemas que afecta a nuestra sociedad de hoy en día que es la depresión. Cabe destacar que este es un problema que aqueja al ser humano a nivel mundial, y que es ahí donde radica precisamente la importancia de este desarrollo, ya que podrá adaptarse fácilmente a cada situación personal de cualquier usuario.

La mensajería instantánea por medio de Internet le permite al ser humano estar comunicado a cualquier hora, en cualquier lugar donde tenga acceso a la red. Es por eso que el enfoque de este desarrollo se encuentra dirigido al cómo explotar las características y capacidades de los Bots a través de un chat en línea. Un Bot puede definirse como un programa informático que es capaz de realizar tareas repetitivas de una manera eficaz sin supervisión constante de algún desarrollador, facilitando de este modo las tareas que pueden resultar tediosas y fatigantes para un humano. Adicionalmente un Bot podrá contar con una base de aprendizaje para dotarlo de inteligencia artificial que le permitirá a este ente informático realizar tareas y tomar decisiones basado en su propio conocimiento adquirido.

Los Bots tienen la cualidad de poder realizar tareas a través de internet por lo cual su uso en chats en línea se ha vuelto altamente recurrido, tanto como por empresas que buscan darles un mejor servicio a sus clientes, así como por cadenas de entretenimiento que buscan llegar a más gente con su contenido. Es aquí donde

encontramos la ventaja de utilizar un chat-bot ya que este permite una interacción con el usuario de manera natural y poco invasiva, permitiendo una recolección amplia de información para su análisis posterior.

Una vez expuestas y analizadas todas las ventajas que brindan estas herramientas tecnológicas, se llega a la conclusión de que para llegar a un número más amplio de personas e ir en el sentido del crecimiento tecnológico se desarrollará una aplicación móvil que podrá adaptarse a la mayoría de los dispositivos inteligentes. Esta aplicación móvil presentará una interfaz amigable para el usuario para su fácil manejo y navegación a través de esta, del mismo modo esta aplicación contará con la implementación de un Bot que se comunicará con el usuario a través de una aplicación de mensajería instantánea para recolectar la información que es relevante en este caso, para que tras su estudio se pueda llegar a una solución eficiente para combatir los efectos de la depresión, incentivando las relaciones interpersonales y la participación inclusiva.

CAPÍTULO 2

PLANTEAMIENTO DEL PROBLEMA

CAPÍTULO 2. PLANTEAMIENTO DEL PROBLEMA.

En la actualidad el humano como un ser social tiene que enfrentar día con día múltiples y diferentes situaciones que influyen directamente en su estado de ánimo creando dentro de sí variadas emociones que repercuten directamente en su desempeño a lo largo del día.

Con el abrumador ritmo de vida que se vive actualmente dentro de las grandes ciudades es cada vez más frecuente sufrir de alguna afección emocional como lo es la depresión. Este padecimiento afecta directamente al individuo que sufre de él ya que es un trastorno de su estado de ánimo que se puede presentar frecuente y recurrentemente, hasta llegar a convertirse en una enfermedad de carácter crónico; esta enfermedad se caracteriza por provocar en el individuo que la padece un sentimiento de tristeza, culpa, malestar, decaimiento, impotencia o baja autoestima, adicionalmente se ha comprobado que produce problemas de carácter somático, así como el desgaste en los procesos cognitivos en las personas que lo padecen y una disminución de su actividad vital habitual.

Debido a la depresión una persona puede presentar una eficiencia por debajo de lo normal en sus tareas cotidianas como desempeñar un trabajo formal, entablar conversaciones, realizar actividades deportivas, así también viéndose afectados su entorno social o familiar y un sinnúmero de actividades más. Es por eso que la depresión representa un problema de gran importancia para el ser humano no solo como una entidad individual sino también como sociedad, ya que una sociedad con un índice de depresión alto puede presentar grandes problemas para sus integrantes en su pleno desarrollo.

Con el avance de las nuevas tecnologías y la globalización acelerada que vivimos hoy en día se ha vuelto muy fácil que cualquier persona sea capaz de adquirir un dispositivo móvil inteligente (smartphone, tablet, etc); gracias a esto la difusión de diferentes contenidos a través de las redes sociales, y mediante el uso de aplicaciones se ha vuelto sumamente sencillo. Es por esto que se propone en este

proyecto el uso de estas herramientas digitales ya que el impacto que pueden llegar a producir es bastante amplio y genera un gran alcance.

La mayoría de usuarios de servicios digitales, como aplicaciones móviles, se siente estrechamente familiarizado con el uso de redes sociales como lo son Facebook, Twitter, Instagram por mencionar solo algunas, es por eso que para combatir el problema que representa la depresión se propone el uso de un Bot ya que gracias a sus cualidades y facilidades que ofrece al realizar tareas repetitivas a través de una red de información es muy útil en la recolección de información a gran escala para fines de análisis estadístico y cuantitativo. Agregado a este hecho, la implementación de un chat-bot como agente intermediario entre el usuario y la aplicación móvil ha demostrado ser una manera efectiva de relacionarse amigablemente con el usuario, ya que promueve una interfaz altamente reconocible por el usuario y de interacción sumamente intuitiva.

En un marco clínico para combatir los efectos de la depresión suelen recetarse antidepresivos y drogas experimentales de una manera indiscriminada “debido a que los médicos tienen que encontrar, mediante prueba y error, el fármaco y la dosis apropiados para tratarlos. Y en los peores escenarios, en cerca del 33% de los pacientes, ninguno de los medicamentos existentes tiene ningún efecto”¹. Por lo cual la persona que padece de este tipo de trastorno puede verse afectada por presentar resistencia a los efectos de los antidepresivos o volverse adicto a las drogas suministradas. Para atacar de manera eficiente y sin riesgos el problema de la depresión se busca gestionar los datos proporcionados por una comunidad amplia de usuarios a través de sistemas de mensajería instantánea proporcionadas por las redes sociales de mayor presencia en la actualidad. El análisis de los datos recabados durante este proceso representa debido a su naturaleza variable un reto a solucionar con el fin de proporcionar al usuario una solución óptima y adecuada para su caso específico.

Finalmente se pretende que con la implementación de la aplicación desarrollada a gran escala se logre combatir de manera efectiva los síntomas y afecciones ocasionadas por la depresión buscando de este modo incrementar la felicidad a

nivel general en nuestra sociedad y de esta manera contribuir a crear un mejor mundo donde la convivencia social sea más amena y satisfactoria.

CAPÍTULO 3

MARCO TEÓRICO

CAPÍTULO 3. MARCO TEÓRICO.

3.1 DEPRESIÓN

El termino depresión se deriva del latín “depressio” que se interpreta como ‘opresión’, ‘encogimiento’ o ‘abatimiento’. Esto toma un gran sentido cuando hacemos un análisis de memoria propio y nos damos cuenta de que en algún momento de nuestras vidas nos hemos sentido tristes, sin ganas de hacer nada o simplemente nos sentimos pequeños dentro de este mundo que nos rodea; esta clase de sentimientos pueden presentarse con mayor o menor frecuencia, y pueden ser pasajeros o presentarse por un periodo de tiempo muy largo.

Dígase entonces que la depresión desde un punto de vista científico es un trastorno o el conjunto de síntomas que afectan el cuadro emocional de una persona de manera negativa; una vez establecido esto cabe mencionar que en el ámbito profesional los especialistas de la salud se refieren a la depresión como un término ambiguo, ya que este puede aplicarse a una cantidad numerosa de diferentes cuadros clínicos, que presentan cada uno a su vez diferentes síntomas, que van desde cuadros no significativos sin repercusiones físicas o mentales, hasta cuadros severos con síntomas que atentan directamente contra la salud de la persona que la padece, llegando a poner su vida en riesgo.

Como se puede ver la depresión es una enfermedad que todos podemos padecer, pero no por ser algo común significa que no sea grave. Es importante entonces conocer cuáles son los factores que provocan o influyen en el padecimiento de esta enfermedad. La depresión no tiene una causa única, lo que significa que son múltiples los factores que intervienen en la aparición de algún trastorno depresivo como pueden ser factores biológicos, genéticos, bioquímicos, psicológicos y psicosociales.

Algunos síntomas de la depresión suelen verse reflejados de una generación a otra, por lo que por su estrecha similitud parece existir alguna relación genética.

Por otro lado, un significativo número de evidencias y pruebas clínicas realizadas como la resonancia magnética indican que ciertos cuadros depresivos se relacionan

con alteraciones en el funcionamiento de neurotransmisión del sistema nervioso central, así como cambios en el funcionamiento de diferentes zonas del cerebro encargadas de regular el ánimo, apetito, sueño, etc.

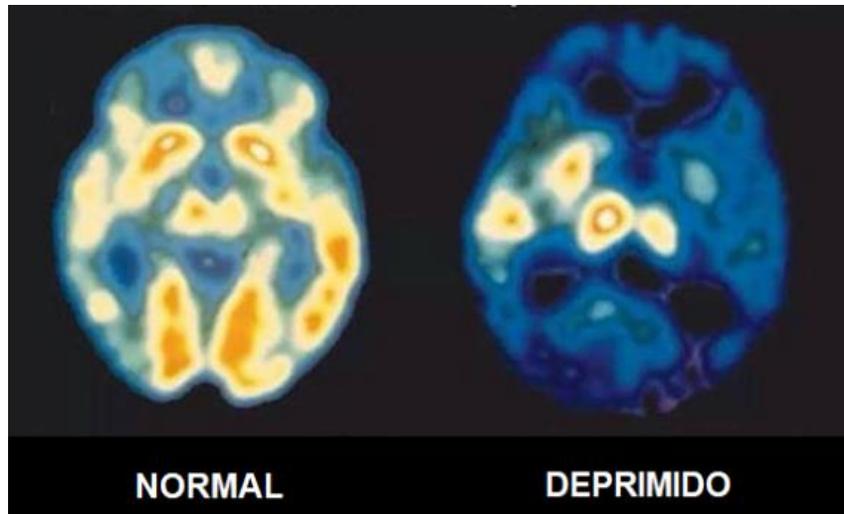


Figura 3.1 Comparación de la actividad entre un cerebro normal y el de un paciente diagnosticado con depresión. [1]

Entre otros factores causales de la depresión se pueden mencionar factores ambientales, una alimentación poco saludable o deficiente, falta de ejercicio o completa inactividad física, falta de descanso o sueño, tabaquismo, falta de vitaminas. Adicionalmente se mencionan diferentes acontecimientos o situaciones traumáticas en las que se ve envuelta una persona como el ver morir a alguien, una pérdida o ruptura amorosa, haber vivido una experiencia cercana a la muerte, entre muchas otras.

La depresión puede ocasionar además de importantes y variadas consecuencias sociales, relacionadas ya sea con un ámbito laboral o la incapacidad de desenvolverse correcta y saludablemente en un entorno social, afectaciones físicas y biológicas importantes como el desgaste de la capacidad psicomotriz, así como un deterioro en los procesos cognitivos.

Esta enfermedad afecta directamente la capacidad de quien la padece para realizar sus actividades cotidianas de manera eficiente, y produce dolor no solo para quien es víctima de este trastorno sino también para las personas sus seres queridos y

personas con las que tiene una relación cercana, sin mencionar que en el peor de los casos este padecimiento puede conducir a la muerte de la persona.

3.1.1 TIPOS DE DEPRESIÓN

Como se mencionó anteriormente la depresión representa un concepto muy ambiguo sin embargo a través del tiempo y en base a estudios se ha llegado a una forma estandarizada para clasificar la depresión en diferentes tipos con base a varias características. A continuación, se hace mención de manera breve de los diferentes tipos de trastornos depresivos incluyendo una explicación sintetizada.

Trastorno depresivo grave: también conocido como trastorno depresión mayor o depresión unipolar, se distingue por la presencia de diferentes síntomas que en su conjunto interfieren directamente con la capacidad de la persona para desempeñar su vida con normalidad.

Los criterios internacionalmente establecidos tanto el DSM-IV (El Manual diagnóstico y estadístico de los trastornos mentales) como el CIE-10 (Clasificación internacional de enfermedades, décima versión) descritos para el trastorno depresivo mayor son ⁱⁱ:

Criterio A: *La presencia de por lo menos cinco de los síntomas siguientes, durante al menos dos semanas:*

- *Estado de ánimo triste, disfórico o irritable durante la mayor parte del día y durante la mayor parte de los días*
- *Anhedonia o disminución de la capacidad para disfrutar o mostrar interés y/o placer en las actividades habituales*
- *Disminución o aumento del peso o del apetito*
- *Insomnio o hipersomnio (es decir, dificultades para descansar, ya sea porque se duerme menos de lo que se acostumbraba o porque se duerme más; véanse los trastornos en el sueño)*
- *Enlentecimiento o agitación psicomotriz*
- *Astenia (sensación de debilidad física)*
- *Sentimientos recurrentes de inutilidad o culpa*
- *Disminución de la capacidad intelectual*
- *Pensamientos recurrentes de muerte o ideas suicidas*

Criterio B: No deben existir signos o criterios de trastornos afectivos mixtos (síntomas maníacos y depresivos), trastornos esquizoafectivos o trastornos esquizofrénicos.

Criterio C: El cuadro repercute negativamente en la esfera social, laboral o en otras áreas vitales del paciente.

Criterio D: Los síntomas no se explican por el consumo de sustancias tóxicas o medicamentos, ni tampoco por una patología orgánica.

Criterio E: No se explica por una reacción de duelo ante la pérdida de una persona importante para el paciente.

Aunque un episodio de depresión grave puede suceder solo una vez en el curso de vida de una persona, esté con frecuencia dejará secuelas que serán recurrentes durante toda su vida.

El trastorno distímico: también conocido como distimia, se caracteriza principalmente por presentar síntomas de larga duración que pueden ser superiores a los dos años, aunque son de un carácter no tan grave, afectan a la persona impidiendo que se siente completamente bien. Las personas con distimia también pueden padecer uno o más episodios de depresión grave a lo largo de sus vidas.

El CIE-10 describe la distimia como “Depresión crónica del estado de ánimo que no se corresponde con la descripción o las pautas para el diagnóstico de un trastorno depresivo recurrente, episodio actual leve o moderado (F33.0, F33.1), por su gravedad o por la duración de los episodios (aunque pueden haberse satisfecho las pautas para un episodio depresivo leve en el pasado, en especial al inicio del trastorno). La proporción entre las fases recortadas de depresión leve y los períodos intermedios de comparativa normalidad es muy variable. Los enfermos tienen a menudo días o semanas en los que refieren encontrarse bien, pero durante la mayor parte del tiempo (frecuentemente durante meses seguidos) se sienten cansados y deprimidos, todo les supone un esfuerzo y nada les satisface. Están meditabundos y quejumbrosos, duermen mal y se sienten incapaces de todo, aunque normalmente pueden hacer frente a las demandas básicas de la vida cotidiana. La distimia, por lo tanto, tiene mucho en común con los conceptos de neurosis depresiva y depresión

neurótica. Si fuere necesario, puede especificarse si el comienzo es precoz (tuvo lugar durante la adolescencia o la tercera década de la vida) o tardío.”ⁱⁱⁱ

Los criterios establecidos por el CIE-10 para el diagnóstico del trastorno distímico son^{iv}:

Criterio A. *Estado de ánimo crónicamente depresivo la mayor parte del día de la mayoría de los días, manifestado por el sujeto u observado por los demás, durante al menos 2 años.*

Nota: En los niños y adolescentes el estado de ánimo puede ser irritable y la duración debe ser de al menos 1 año.

Criterio B. *Presencia, mientras está deprimido, de dos (o más) de los siguientes síntomas:*

1. *pérdida o aumento de apetito*
2. *insomnio o hipersomnia*
3. *falta de energía o fatiga*
4. *baja autoestima*
5. *dificultades para concentrarse o para tomar decisiones*
6. *sentimientos de desesperanza*

Criterio C. *Durante el período de 2 años (1 año en niños y adolescentes) de la alteración, el sujeto no ha estado sin síntomas de los Criterios A y B durante más de 2 meses seguidos.*

Criterio D. *No ha habido ningún episodio depresivo mayor durante los primeros 2 años de la alteración (1 año para niños y adolescentes); por ejemplo, la alteración no se explica mejor por la presencia de un trastorno depresivo mayor crónico o un trastorno depresivo mayor, en remisión parcial.*

Nota: Antes de la aparición del episodio distímico pudo haber un episodio depresivo mayor previo que ha remitido totalmente (ningún signo o síntoma significativos durante 2 meses). Además, tras los primeros 2 años (1 año en niños y adolescentes) de trastorno distímico, puede haber episodios de trastorno depresivo mayor

superpuestos, en cuyo caso cabe realizar ambos diagnósticos si se cumplen los criterios para un episodio depresivo mayor.

Criterio E. *Nunca ha habido un episodio maníaco, un episodio mixto o un episodio hipomaníaco y nunca se han cumplido los criterios para el trastorno ciclotímico.*

Criterio F. *La alteración no aparece exclusivamente en el transcurso de un trastorno psicótico crónico, como son la esquizofrenia o el trastorno delirante.*

Criterio G. *Los síntomas no son debidos a los efectos fisiológicos directos de una sustancia (p. ej., una droga, un medicamento) o a enfermedad médica (p. ej., hipotiroidismo).*

Criterio H. *Los síntomas causan un malestar clínicamente significativo o deterioro social, laboral o de otras áreas importantes de la actividad del individuo.*

Especificar si:

- *Inicio temprano: si el inicio es antes de los 21 años*
- *Inicio tardío: si el inicio se produce a los 21 años o con posterioridad*

Especificar (para los últimos 2 años del Trastorno distímico):

- *Con síntomas atípicos.*

Depresión psicótica: esta se presenta cuando un trastorno depresivo grave se ve acompañado por algún tipo de psicosis, se caracteriza por presentar casos de alucinaciones, así como ruptura con la realidad y alguna clase de paranoia o delirios.

Depresión posparto: el diagnóstico de esta depresión se aplica cuando una mujer que ha tenido un parto recientemente presenta un episodio de depresión grave dentro del primer mes después del parto. Se calcula que del 10 al 15 por ciento de las mujeres padecen depresión posparto luego de dar a luz ^v.

El trastorno afectivo estacional: este caracteriza por la manifestación de diversos síntomas de depresión durante los meses del invierno, cuando la temperatura ambiental disminuye al igual que el tiempo de luz solar por día. Este tipo de depresión generalmente es pasajera y en su mayoría desaparece cuando inician las temporadas de primavera y verano. “Los medicamentos antidepresivos y la

psicoterapia pueden reducir los síntomas del trastorno afectivo estacional, ya sea de forma independiente o combinados con la terapia de luz”^{vi}.

El trastorno bipolar: también conocida como enfermedad maniacodepresiva, suele presentarse en una menor escala que la depresión grave o la distimia. Este trastorno se distingue por cambios cíclicos en el estado de ánimo de la persona que sufre de esta afección que alternan entre estados de ánimo muy elevados (como manía o una inmensa felicidad e hiperactividad) y estados de ánimo muy bajo (como depresión y melancolía).

Trastorno depresivo no especificado: esta clasificación surge cuando los síntomas presentados por la persona sugieren que se trata de un cuadro de trastorno depresivo sin embargo no son suficientes para clasificarlo como alguno de los tipos anteriormente descritos.

Los ejemplos del trastorno depresivo especificados por el CIE-10 incluyen^{vii}:

1. *Trastorno disfórico premenstrual: los síntomas (p. ej., estado de ánimo acusadamente deprimido, ansiedad importante, marcada labilidad afectiva, pérdida de interés en las actividades) se presentaron con regularidad durante la última semana de la fase luteínica (y remitieron a los pocos días del inicio de las menstruaciones) en la mayoría de los ciclos menstruales del último año. Estos síntomas tienen que ser de la suficiente gravedad como para interferir notablemente en el trabajo, los estudios o las actividades habituales y estar completamente ausentes durante al menos 1 semana después de las menstruaciones.*
2. *Trastorno depresivo menor: episodios de al menos 2 semanas de síntomas depresivos, pero con menos de los cinco ítems exigidos para el trastorno depresivo mayor.*
3. *Trastorno depresivo breve recidivante: episodios depresivos con una duración de 2 días a 2 semanas, que se presentan al menos una vez al mes durante 12 meses (no asociados con los ciclos menstruales).*
4. *Trastorno depresivo pospsicótico en la esquizofrenia: un episodio depresivo mayor que se presenta durante la fase residual en la esquizofrenia.*

5. *Un episodio depresivo mayor superpuesto a un trastorno delirante, a un trastorno psicótico no especificado o a la fase activa de la esquizofrenia.*
6. *Casos en los que el clínico ha llegado a la conclusión de que hay un trastorno depresivo, pero es incapaz de determinar si es primario, debido a enfermedad médica o inducido por sustancia.*

3.1.2 TRATAMIENTOS CONTRA LA DEPRESIÓN.

Una vez se ha diagnosticado algún cuadro de depresión, es importante contrarrestar sus síntomas y consecuencias de manera inmediata para no dejar que el cuadro sintomático avance y se agrave el problema.

Cómo se mencionó con anterioridad, existen múltiples tipos reconocidos de depresión, sin embargo cabe destacar que aun cuando se haya diagnosticado alguno de estos tipos de trastorno específico sin importar sus características si este padecimiento interfiere directamente con la vida del paciente incapacitando el desempeño de sus actividades cotidianas, o limitando un adecuado estado general de salud, es conveniente aplicar algún tratamiento de los múltiples ya conocidos, con el fin de ayudar al estado actual de la persona que padece el síndrome, restaurando el correcto desempeño de sus capacidades para desempeñar alguna actividad ya sea laboral o de cualquier otra índole, así como la capacidad de establecer nuevas y consistentes relaciones sociales.

Es de suma importancia conocer que existen múltiples y diferentes tipos de tratamientos para combatir los efectos del trastorno depresivo, cada uno de estos tratamientos ha demostrado algún grado de eficacia para contrarrestar la depresión, sin embargo no hay algún tratamiento específico que garantice la inmediata recuperación de la persona que presenta el cuadro de depresión, por lo que es recomendable que en caso de no notar mejoría con algún tratamiento optar por probar una serie de tratamientos alternativos.

El tratamiento adecuado para cada persona dependerá de diversos factores a considerar, y en general de un análisis previo realizado por algún especialista. Hay una extensa variedad de medicamentos antidepresivos que son recetados para combatir la depresión, sin embargo, también hay que destacar que existen terapias psicológicas que se enfocan en el proceso cognitivo conductual de la persona para tratar la enfermedad. Este tipo de soluciones terapéuticas representan una gran opción para contrarrestar los efectos de un cuadro depresivo ya que ciertos estudios han demostrado que las terapias aplicadas a algunos casos resultan ser más eficaces que un tratamiento psicofarmacológico.

Principalmente, es recomendable identificar una de las posibles causas sea debido a alguna enfermedad de carácter no psicológico, ya que al tratarla podríamos disminuir considerablemente los síntomas del cuadro depresivo o en su defecto erradicarlo por completo consiguiendo así una recuperación total del paciente.

A continuación, se realiza una breve descripción de algunos de los tratamientos para combatir los cuadros depresivos más comunes.

Tratamientos farmacológicos:

Para el tratamiento de trastornos depresivos mayores se emplea el uso de distintos fármacos de efecto psicotrópico que ayudan a disminuir los síntomas de la enfermedad.

En general estos medicamentos antidepresivos se dividen en tres diferentes clases:

- Inhibidores de la monoaminoxidasa (IMAO).
- Tricíclicos.
- De segunda generación.

En la actualidad los efectos producidos por los fármacos antidepresivos se ven relacionados a una intervención sobre la actividad de los neurotransmisores de la persona que los consume. Más específicamente estos antidepresivos producen cierto grado de inhibición de las proteínas que son encargadas de transportar la monoamina de la dopamina o serotonina. Produciendo de este modo una alteración en la sinapsis que interviene en la depresión ocasionando condiciones favorables para mejorar el estado de ánimo del paciente.

Cabe destacar que una vez comenzado el tratamiento con antidepresivos, los cambios o efectos no comenzarán a notarse inmediatamente, sino que tomarán un periodo de tiempo que puede tardar desde una semana a tres semanas. Se recomienda que, si después de transcurrido este periodo de tiempo límite de tres semanas no se nota alguna mejoría o cambio favorable, se suspenda la administración del fármaco y se opte por probar con otro diferente de la misma clase.

Este tipo de tratamientos lleva el riesgo intrínseco de sufrir alguna clase de efecto secundario debido a los componentes químicos de la medicina. Estos efectos colaterales pueden incluir entre otros la falta de sueño, dolores estomacales, cefalea, poca fuerza corporal, etc.

Tratamientos cognitivo-conductuales:

Este tipo de tratamiento parte del hecho de que el hombre basa sus sentimientos en la información que procesa del medio que lo rodea asignando un cierto significado a los estímulos que recibe de acuerdo a un conjunto de experiencias previas que ha vivido. Se entiende entonces que los trastornos depresivos son provocados por los pensamientos irracionales. Para el desarrollo de este tratamiento se busca analizar los pensamientos que se encuentran detrás de algún comportamiento negativo, para de este modo solucionar el problema psicológico.

El tratamiento consiste en terapias que evalúan de una forma lógica y realista las situaciones en las que se ve envuelto el paciente. Por lo general una persona que sufre de depresión solo presta atención a los aspectos negativos de una situación por lo que por medio de la terapia cognitivo-conductual se busca ocasionar un cambio para que el paciente aprenda a tomar en cuenta todos los aspectos e

información relevante de las situaciones que vive. De igual modo el paciente aprenderá a formular explicaciones lógicas y racionales, es decir que se detecte un pensamiento negativo e irracional de manera automática, esta persona sea capaz de cambiar ese pensamiento por uno elaborado de manera lógica y racional. Se propone probar esta clase de pensamientos racionales por medio de la realización de experimentos conductuales que ayuden a comprobar que el comportamiento, la interacción con otras personas y la forma de actuar frente a los problemas ha mejorado para bien.

Durante las primeras etapas del tratamiento es común emplear técnicas conductuales con la finalidad de alcanzar el nivel de comportamiento que tenía la persona antes de sufrir de la depresión. Después de haber logrado esto se utilizan algunas técnicas para encontrar si es que existen distorsiones cognitivas con base en los pensamientos negativos y automáticos.

Algunas de las técnicas conductuales empleadas durante la terapia son: programar ciertas actividades que refuercen una conducta asertiva de manera gradual, y la práctica de relaciones personales.

Cabe destacar que el tiempo que durará este tipo de tratamiento está en función de la gravedad del cuadro depresivo que se esté tratando, así como de la mejoría que presente el paciente durante las sesiones de terapia.

Psicoterapia Interpersonal:

La psicoterapia interpersonal aborda la depresión como una enfermedad ocasionada por una deficiencia en la capacidad de una persona para establecer relaciones personales y desenvolverse con seguridad en un ambiente social.

El tipo de disfunciones que tiene en cuenta son ^{viii}:

- *Un duelo no resuelto. La asunción de una pérdida es un proceso difícil y doloroso y no siempre se realiza de forma adecuada.*
- *Cuando los papeles en la relación no están bien establecidos, se dan disputas en los roles. Por ejemplo, las discusiones en la pareja sobre los papeles de cada cual, las discusiones entre padres e hijos, etc. Este problema persiste por falta de habilidades para conseguir lo que quiere y también porque se tienen expectativas que están lejos de la realidad de lo que se pueden conseguir.*
- *Los problemas que surgen en las transiciones de un rol social a otro pueden estar en el origen de la depresión. Por ejemplo, cuando se cambia de situación en el trabajo tanto por una promoción como por pasar al paro o a la jubilación.*
- *Otra disfunción que considera esta terapia es el déficit interpersonal de habilidades necesarias de comunicación que pueden llevar al aislamiento social.*

Al final este tipo de terapia se enfoca en mejorar de manera significativa la capacidad del paciente para desarrollarse en sociedad entablando relaciones personales significativas, enseñando métodos eficaces para este fin.

Actividad física:

Se recomienda a la persona que sufre de depresión realizar alguna actividad física o practica algún deporte ya que en ciertos casos se ha visto que complementar algún tratamiento contra la depresión con ejercicio y acondicionamiento físico resulta ser de gran efectividad para contrarrestar los síntomas de depresión.

“Entre otros efectos el ejercicio físico estimula la producción de proteínas que reparan las neuronas y aumentan su proliferación, regula los niveles de serotonina y otros neurotransmisores, aumenta el flujo sanguíneo cerebral y estimula la liberación de endorfinas.”^{ix}

3.1.3 LA DEPRESIÓN EN MÉXICO Y EL MUNDO.

Como ya se ha visto la depresión es una enfermedad que puede afectar a cualquier persona sin importar, su edad, género o situación social. Es por eso que al ser una enfermedad que engloba a toda la sociedad vale la pena analizar cómo repercute en nuestro entorno, tanto nacional como a nivel mundial.

De acuerdo con la Encuesta Nacional de Hogares realizada en el año 2015, elaborada en México por el Instituto Nacional de Estadística y Geografía (INEGI), en ese año se contabilizaron un total de 104.95 millones de habitantes con una edad superior a los 7 años en todo el país, de los cuales un número de 57.2 millones dijeron haberse sentido preocupados o nerviosos por lo menos alguna vez.

En México este tipo de sentimientos presenta una variación con respecto a con qué frecuencia se presentan en las personas. A continuación, se muestra una gráfica realizada con base a la información proporcionada por la Encuesta Nacional de Hogares realizada en el año 2015 por el INEGI en México, donde se compara el número de personas que se han sentido deprimidas ya sea diariamente, alguna vez durante la semana, por lo menos una vez al mes y finalmente, aunque sea una vez al año; adicionalmente se proporciona la información de qué número de persona toma alguna clase de antidepresivo para controlar su estado de ánimo.

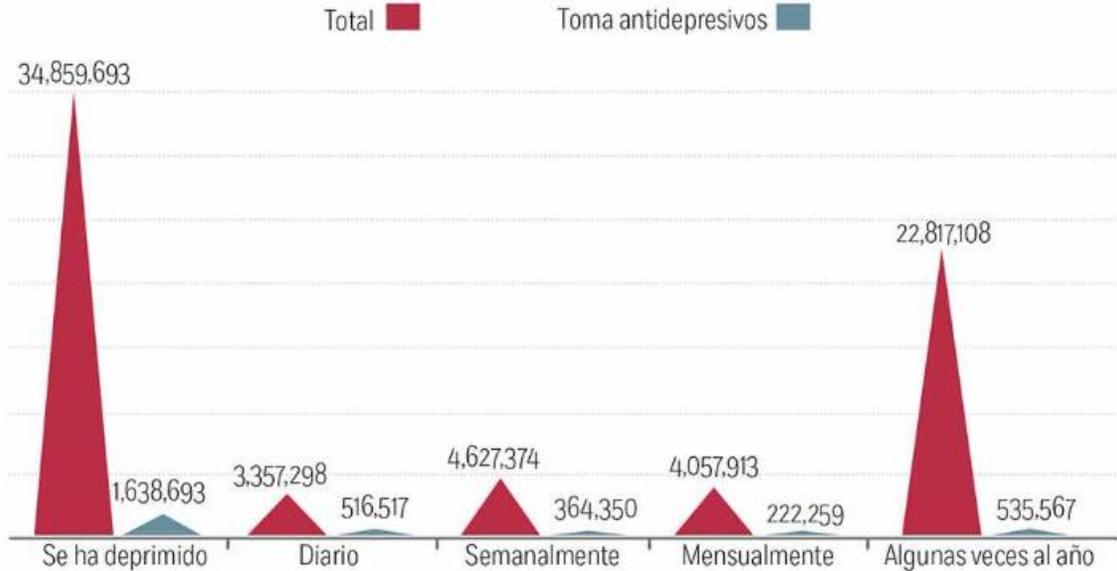


Figura 3.2 Gráfica del número de personas deprimidas en México y la frecuencia con la que se deprimen. [2]

El periódico Excélsior, que es uno de gran distribución y credibilidad en México, nos menciona dentro de un artículo relacionado a la depresión en México que “Al menos 34.5 millones de personas en México se han sentido alguna vez deprimidas; de ellas, 14.48 millones son hombres y 20.37 millones son mujeres; de ese total, sólo 1.63 millones de personas toman antidepresivos”^x. Este dato resulta alarmante el total de personas que se han sentido deprimidas representa un gran número de la población total de México que oscila los 127.5 millones de habitantes. Adicionalmente podemos notar que en el país son las mujeres quienes se ven más afectadas por este tipo de trastornos depresivos.

En el caso de México, según los datos publicados por el INEGI, las estadísticas de mortalidad muestran que entre los años 2011 y 2015 debido a los trastornos mentales fallecieron alrededor de 23,335 personas.

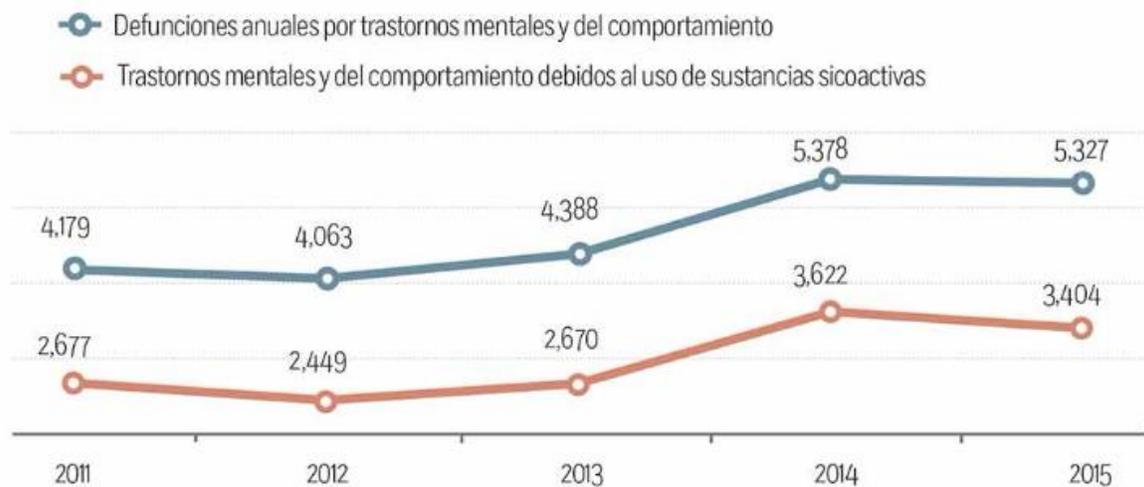


Figura 3.3 Gráfica del número de muertes ocasionadas por un trastorno mental entre 2011 y 2015. [2]

La depresión es una enfermedad que afecta de igual modo a la sociedad alrededor del mundo, según datos dados a conocer por la Organización Mundial de la Salud (OMS), en el 2015 esta enfermedad afectó a más de 322 millones de habitantes de todo el mundo, lo que se puede traducir a un equivalente del 4,4 por ciento de toda la población mundial. Además, cabe destacar que este porcentaje ha incrementado con respecto a anteriores evaluaciones realizadas.

Según la OMS “La depresión es la principal causa de problemas de salud y discapacidad en todo el mundo”.

3.2 BASES DE DATOS.

Se entiende por base de datos a un grupo de datos que por compartir alguna característica o relación en común pueden ser almacenados de una manera ordenada para algún propósito futuro.

Un ejemplo tangible de lo que es una base de datos puede ser una biblioteca, ya que esta puede contener múltiples documentos almacenados y ordenados por tópicos, nombres, años etc.

Las bases de datos representan una alternativa a la actual problemática de almacenamiento de grandes cantidades de información digital que se generan día con día más aún con el uso de los servicios en línea que son de carácter global, siendo ésta una alternativa virtual eficiente para dicho propósito.

Para el manejo de este tipo de datos se han desarrollado e implementado diferentes softwares conocidos como sistemas gestores de bases de datos que facilitan al administrador de la base de datos con la que se esté trabajando realizar operaciones de almacenaje, ordenamiento o consulta de una manera eficaz y fluida. Estos gestores de bases de datos tendrán diferentes características y cualidades dependiendo del desarrollador y del tipo de la base de datos con la que se esté trabajando.

Usualmente las bases de datos tienen su aplicación dentro del ámbito empresarial o institucional debido a que este tipo de organizaciones requieren almacenan grandes cantidades de información con el mayor orden posible manteniendo estos datos con una alta disposición para su uso inmediato. Estas bases de datos pueden ser de carácter público o privado ya que cierta información puede ser consultada por personas ajenas a alguna institución mediante servicios en línea.

Las bases de datos tienen la capacidad de almacenar diferentes tipos de información, ya sean tipos de datos binarios, imágenes, texto, sonido, etcétera, sin embargo, esta clase de información que puede ser de carácter personal y de un alto valor está protegida por la Ley Federal de Transparencia y Acceso a la Información Pública Gubernamental, que se encuentra actualmente vigente en México.

3.2.1 TIPOS DE BASES DE DATOS.

Se determina el tipo de las bases de datos con base en los diferentes rubros que las caracterizan, tomando en cuenta el propósito y entorno para el cual fueron creadas.

Bases de datos estáticas

Este tipo de bases de datos tienen como objetivo principal el almacenamiento de datos de lectura, los cuales cumplen el propósito de fungir como fuente de consulta para el posterior análisis de dicha información y con ello los desarrolladores sean capaces de recabar herramientas que les permitan realizar estudios sobre dichos datos.

Bases de datos dinámicas

Las bases de datos dinámicas son propiamente bases que contienen información cambiante y tienen un uso constante, es decir, que le permiten al desarrollador realizar cambios o modificaciones a dicha información, estas modificaciones siguen el principio de las operaciones básicas tales como eliminación, actualización, consulta y alta de datos.

Directorios

Un directorio es una recopilación de datos que administra información, agrupando dichos datos en un depósito virtual que cumple con el objetivo de mantener ordenada la información para que el acceso a ésta sea sencillo y eficaz.

Los directorios más comunes encuentran su clasificación en dos formas, personales o empresariales. Los primeros con contenido de datos de carácter personal y privado (Nombre, edad, teléfono, domicilio, etc) y los segundos haciendo referencia a la información de la empresa (datos de facturación, razón de la empresa, etc).

Bases NoSQL

Dentro del mundo de las bases de datos existen las que son de tipo no relacional que también son llamadas bases de datos tipo "NoSQL", estos en general presentan sistemas de gestión de bases de datos diferentes de los modelos clásicos SGBDR (Sistema de Gestión de Bases de Datos Relacionales), principalmente se diferencian de estos últimos por el hecho de que no utilizan lenguaje SQL para realizar consultas o programación. Por otro lado, cabe destacar que en este tipo de bases de datos los datos que se almacenan no implementan estructuras como tablas o relaciones. Generalmente las bases de datos tipo NoSQL se escalan de manera horizontal y esto les permite una eficiencia considerable al manejar datos masivos. Sin embargo, cabe mencionar que de ser requerido este tipo de bases de

datos pueden implementar lenguajes de consulta SQL, aunque no resulta ser lo óptimo.

La mayoría de los desarrolladores que implementan este tipo de bases de datos suelen conceptualizar este modelo como una clase de almacenamiento estructurado, ya que requiere de una estructuración sofisticada para explotar las cualidades que ofrece una base de datos de este tipo. Las bases de datos no relacionales pueden clasificarse conforme a la forma que tienen de almacenar los datos siendo las más comunes las categorías de llaves, implementación de BigTable y bases de datos orientadas a grafos.

Las bases de datos tipo de tipo NoSQL, resultan ser relativamente nuevas ya que tuvieron su auge debido a su implementación en las principales plataformas de redes sociales como Facebook, Twitter, Google e Instagram. Los desarrolladores de este tipo de aplicaciones prefieren el uso de las bases de datos no relacionales sobre las tradicionales SGBDR, ya que las cualidades de tiempo real que ofrecen optimizan el funcionamiento de las redes sociales donde se realizan movimientos de grandes cantidades de datos de forma constante y que necesitan contar con datos actualizados. Debido a que las bases de datos tradicionales ocupan mucho tiempo en procesos de análisis y coherencia, estas son poco recurridas en los desarrollos innovadores de tecnologías para redes sociales.

Por lo general las bases de datos de tipo NoSQL se encuentran altamente optimizadas para realizar operaciones en las que se recuperan datos altamente dinámicos y también para agregar nuevos datos a la base. Haciendo una comparativa con los modelos tradicionales de bases de datos relacionales, las ganancias son significativas en el aspecto de rendimiento y escalabilidad, más aún cuando se trata de modelos de datos orientados a redes sociales que requieren cualidades de datos en tiempo real y manejo masivo de datos.

Conforme a lo establecido anteriormente se presenta a continuación una lista que numera las ventajas y desventajas que posee el uso de una base de datos de tipo NoSQL.

Ventajas

- Responden de manera efectiva a la escalabilidad horizontal que muy frecuentemente es recurrida por las empresas.
- Manejo optimizado para cantidades masivas de datos.
- Velocidad de consulta.
- Un escalamiento relativamente sencillo.
- No generan cuellos de botella.
- Nuevas plataformas que ofrecen su servicio de base de datos en la nube.
- Se pueden ejecutar en máquinas con características poco especializadas.
- Fácil implementación en entornos de desarrollo para dispositivos móviles.
- Actualización de datos instantánea y características de tiempo real.

Desventajas

- Al ser un tipo de bases de datos relativamente nuevo poseen un soporte no tan amplio por parte de las empresas que ofrecen estos servicios.
- Son sistemas que se encuentran en constante evolución y no están completamente maduros. Esto provoca que empresas que no confían en estos sistemas por ser relativamente nuevos limiten los recursos que se proporcionan para la innovación de estos sistemas. Sin embargo, cabe mencionar que son empresas tecnológicas líderes como Facebook y Google las que se encargan de innovar en el desarrollo de estos sistemas.
- Al ser nuevos no cuentan con las herramientas suficientes para realizar análisis de minería de datos. Se necesita un amplio conocimiento de programación para poder realizar este tipo de análisis.
- No existe una comunidad tan grande de desarrolladores para este tipo de sistemas de bases de datos. Por lo cual es más complejo encontrar soluciones a problemas que puedan presentarse con estos sistemas.
- Falta de ciertos estándares que regularicen el uso de estas bases de datos. Falta una entidad que se encargue de proporcionar ciertos parámetros para incrementar la compatibilidad entre los sistemas de bases de datos NoSQL.

3.2.2 FIREBASE REALTIME DATABASE.

Firestore Realtime Database es un servicio de gestión de bases de datos NoSQL que está alojado en la nube, este servicio es proporcionado por la empresa Google. Los datos que utiliza este gestor se almacenan en un formato JSON, lo cual permite la estructuración de los datos en forma horizontal, generalmente basado en una estructura de tipo árbol. Esta plataforma brinda la característica de sincronización de datos en tiempo real, lo que significa que cada cliente que se encuentre conectado a la base de datos puede ver información actualizada de manera inmediata. El sistema en general brinda una alta compatibilidad con múltiples plataformas a través de sus SDK (Software Development Kit) orientados a iOS, Android y JavaScript.

A continuación, se muestra una tabla con las principales funcionalidades que ofrece Firestore Realtime Database.

Tabla 3.1 Funciones Firestore Realtime Database.

Función	Descripción
Tiempo real	A diferencia de los sistemas gestores de bases de datos típicos que utilizan solicitudes HTTP para realizar consultas y manejo de datos, Firestore Realtime Database utiliza una sincronización de datos, lo que significa que cada vez que un dato es modificado dentro de la base de datos, todos los dispositivos conectados a la base recibirán una actualización con los datos nuevos en cuestión de milisegundos. Esto permite crear experiencias envolventes y colaborativas.
Sin conexión	Las aplicaciones que implementan Firestore Realtime Database en su desarrollo seguirán reflejando la base de datos aun cuando no se posea una conexión a internet, esto debido a que el SDK proporcionado por Firestore hace que los datos persistan en un almacenamiento local del dispositivo para que cuando se cuente nuevamente con una conexión a internet, el dispositivo que ejecuta la aplicación sincronice los datos con la versión actualizada de la base de datos.
Accesible desde dispositivos cliente	Una de las ventajas de utilizar Firestore Realtime Database es que a diferencia de otras bases de datos esta no necesita de un servidor intermediario de aplicaciones por lo cual se puede acceder directamente a la información dentro de la base de datos desde un navegador o dispositivo móvil. A pesar de este hecho la

base de datos se mantendrá siempre segura ya que se cuenta con métodos de seguridad que se implementa a través de reglas dentro de la consola de Firebase, estas reglas se basan en expresiones que se llevan a cabo cada vez que se consultan o se registran datos en la base.

Las características ofrecidas por la base de datos de Firebase permiten el desarrollo de aplicaciones innovadoras dentro del área de las redes sociales ya que por la persistencia local de datos las aplicaciones que implementan la base de datos se podrán adaptar para otorgar al usuario una experiencia ampliamente satisfactoria. Además, no existen pérdida de información ya que la base de datos se encuentra en constante sincronización cuando hay y no hay conexión.

La plataforma de Firebase Realtime Database proporciona múltiples funcionalidades y varias optimizaciones en comparación con una base de datos tradicional de tipo relacional. En general la plataforma está enfocada a permitir solo el uso de operaciones de ejecución rápida, lo cual garantiza la optimización de sistemas y eficacia del servicio, generando de este modo una experiencia en tiempo real capaz de proporcionar respuestas a una cantidad masiva de usuarios.

A diferencia de una base de datos relacional en Firebase Realtime Database no existen tablas ni registros, ya que los datos son almacenados en estructuras JSON, cada vez que un nuevo dato es ingresado a la base de datos este se convierte en un nuevo nodo de la estructura. Por lo general para manejar los múltiples datos se utilizan claves asociadas que pueden ser proporcionadas por el desarrollador, aunque Firebase igual proporciona una función encargada de realizar esta tarea.

A pesar de que la base de datos utiliza una estructuración de árbol JSON, los datos que se almacenen en ella pueden corresponder a los tipos de datos nativos permitidos por el formato JSON de este modo el código que se desarrolle será más fácil de mantener.

3.3 DISPOSITIVOS MÓVILES INTELIGENTES.

Un dispositivo móvil es un aparato pequeño de computación móvil que suele denominarse también como una computadora de bolsillo. Estos dispositivos se caracterizan por tener capacidades de procesamiento altas en comparación de su tamaño diminuto, generalmente ofrecen conexión a Internet y capacidades de almacenamiento en memoria. Son diseñados para realizar funciones específicas, pero en su mayoría ofrecen la posibilidad de realizar múltiples funciones de un carácter más general.

Debido a su tamaño tan compacto estos dispositivos han tenido un gran auge en los tiempos actuales, ya que son fáciles de transportar y resultan ser muy atractivos para los usuarios en el sentido de que ofrecen soluciones y herramientas que facilitan la vida cotidiana.

Cada dispositivo móvil inteligente cuenta con un sistema operativo el cual permite aprovechar las características físicas del hardware del dispositivo móvil. Con el pasar del tiempo, la tecnología móvil inteligente ha crecido a pasos acelerados con una tendencia innovadora significativa, cada vez son más las empresas involucradas en el desarrollo de tecnologías para este tipo de dispositivos.

En el mercado actual existen muchas opciones para adquirir un dispositivo móvil, las cuales van desde aparatos muy sencillos, hasta dispositivos que ofrecen los últimos avances tecnológicos. Esto presenta una variable en precios que puede resultar abrumadora, sin embargo y a pesar de esto los usuarios siempre encuentran una opción que se adapte a sus necesidades y posibilidades económicas.

3.3.1 APLICACIONES MÓVILES.

Una aplicación móvil o también conocidas como “app” por el acortamiento de la palabra “application” en inglés es un programa informático que permite realizar funciones específicas dentro del dispositivo móvil que la esté ejecutando. Generalmente estas aplicaciones desempeñan tareas específicas, aunque más recientemente se cuenta con desarrollos que permiten ejecutar multitareas, es decir que la aplicación realiza una tarea específica en primer plano mientras que en segundo plano puede realizar otra totalmente diferente.

Las aplicaciones surgen con el desarrollo de los dispositivos móviles y las primeras datan de los años noventa donde se implementaron como agendas electrónicas, juegos de arcades, calendarios entre muchos otros, estas aplicaciones resultaban ser muy básicas y consumían pocos recursos.

En la actualidad el desarrollo de aplicaciones móviles se enfoca en los conocidos dispositivos inteligentes que engloban a los teléfonos inteligentes, tabletas y relojes solo por mencionar algunos. Uno de los detonadores para la innovación en el campo de las aplicaciones fue el surgimiento de sistemas operativos que permiten el fácil desarrollo e implementación de aplicaciones en los sistemas que los ejecutan, siendo los principales exponentes los sistemas operativos de Android y IOS de la compañía Apple.

Muchas empresas se dedican actualmente al desarrollo de aplicaciones móviles y el mercado dirige muchos recursos en el desarrollo de nuevas tecnologías aplicativas. Mayoritariamente las empresas o desarrolladores independientes ofrecen sus aplicaciones a través de las plataformas oficiales de distribución que proporcionan las empresas que son dueñas de los diferentes sistemas operativos. Muchas de las aplicaciones que se encuentran actualmente en el mercado son de carácter gratuito, aunque existe una gran cantidad de aplicaciones de paga.

Los principales sistemas operativos que se encuentran vigentes en la actualidad son:

- Android
- iOS
- BlackBerry OS
- Windows Phone
- Samsung
- Nokia

La siguiente gráfica muestra la presencia en el mercado que tiene cada sistema operativo con relación a las ventas que realizaron desde abril del 2017 a abril del 2018, según la compañía especializada en estas estadísticas “StatCounter”.

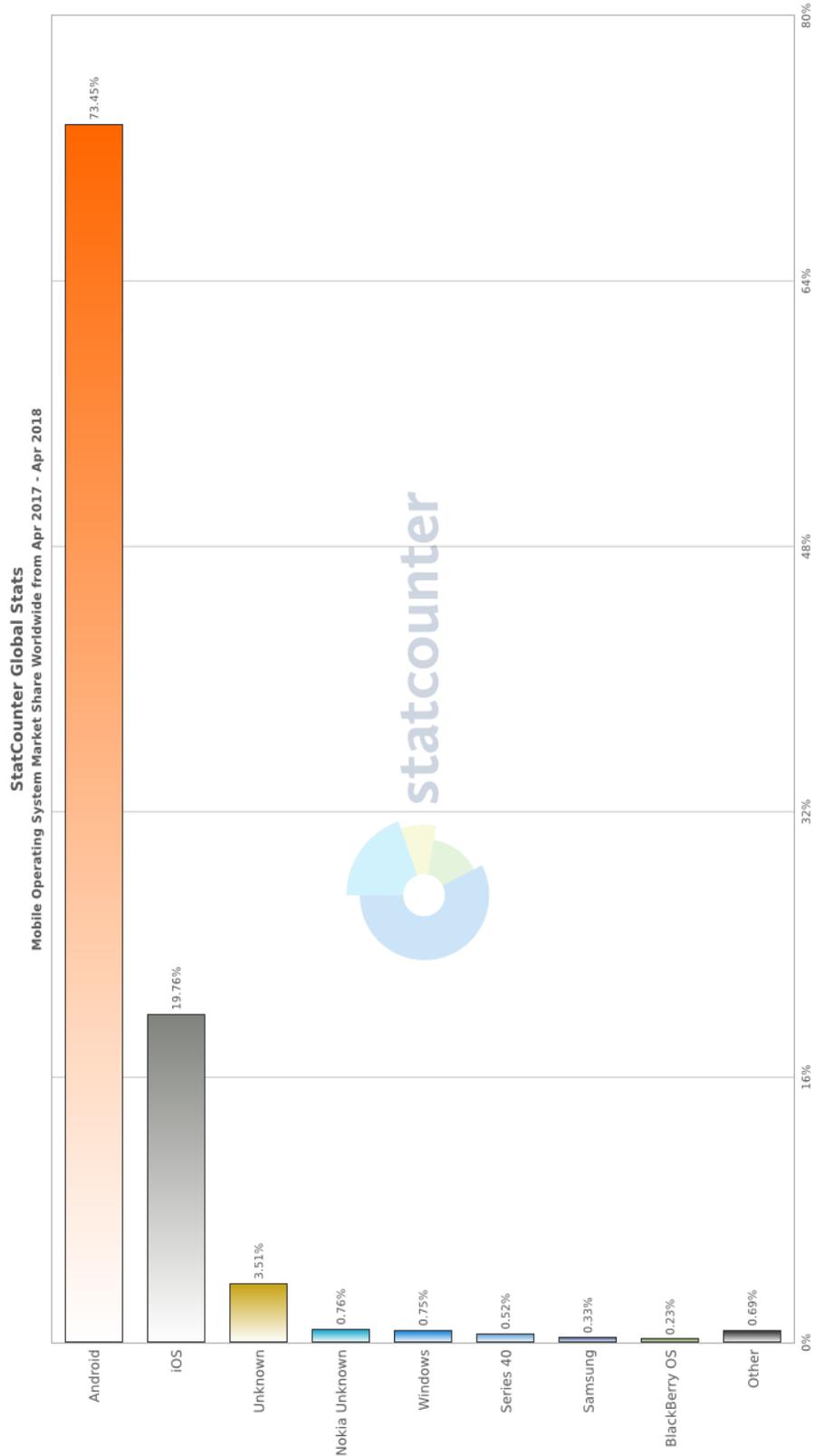


Figura 3.4 Cuota de mercado de los Sistemas Operativos M3viles a nivel mundial.

[3]

Para el desarrollo de una aplicación se debe considerar el dispositivo que estará ejecutándola, así como el sistema operativo mediante el cual se podrá implementar. Cada aplicación deberá estar escrita en un lenguaje de programación compilado que se aceptado por el entorno de desarrollo seleccionado para el desarrollo.

Por lo general las aplicaciones cuando se encuentran en la etapa de desarrollo se prueban mediante el uso de emuladores que virtualizan un sistema operativo o las características de un dispositivo móvil físico, posterior a las pruebas las aplicaciones se ponen en el mercado garantizando su funcionamiento en múltiples equipos.

3.3.2 ANDROID.

Android es un sistema operativo que se utiliza en diferentes dispositivos móviles, este sistema operativo se encuentra basado en Linux por lo cual sus desarrollos son de código abierto. Principalmente se encuentra enfocado en dispositivos móviles con características inteligentes, así como pantallas táctiles dentro de los cuales se pueden mencionar teléfonos inteligentes, tabletas o relojes.

El desarrollo de Android estuvo a cargo de Android Inc., la cual es una empresa desarrolladora de tecnología en software. Cabe destacar que durante el desarrollo fue la empresa Google la encargada de respaldar económicamente ese proyecto, para posteriormente adquirir la compañía completa de Android Inc., en el año 2005.

El primer dispositivo móvil que contó con el sistema operativo Android fue el HTC Dream el cual salió a la venta a finales del año 2008.

Este sistema operativo ha tenido un éxito grandioso gracias a la aceptación de los usuarios y su implementación en equipos desarrollados por diferentes marcas de telefonía celular.

El sistema operativo es capaz de correr aplicaciones desarrolladas mediante el Android SDK, siendo Java el lenguaje de programación más ampliamente utilizado para estos fines, sin embargo, más recientemente se han desarrollado herramientas que permiten el desarrollo de aplicaciones para sistemas operativos Android utilizando diferentes lenguajes de programación como lo son C++, o Kotlin. Adicionalmente existen en el mercado herramientas de software que permiten a usuarios sin conocimientos especializados de programación desarrollar aplicaciones sencillas para funcionar en Android.

Las aplicaciones que se desarrollan para Android se comprimen en un formato APK, que permite la fácil instalación de la aplicación en cualquier dispositivo que cuente con un explorador de archivos.

Android cuenta con su propio entorno de desarrollo integrado oficial el cuál es Android Studio. Este entorno presentó su primera versión estable en diciembre del año 2014. En general este entorno está basado en el software IntelliJ IDEA de JetBrains, además cuenta con una distribución gratuita, por lo que cualquier persona que quiera incursionar en el desarrollo de aplicaciones móviles para Android podrá hacerlo a través de esta herramienta. El software está disponible para plataformas Microsoft, Mac y Linux. La última versión estable es la 3.0, y fue lanzada en octubre de 2017.

3.4 TEORÍA DE BOTS.

Se puede definir a un Bot como un programa informático que se encarga de ejecutar tareas programadas que se repiten con alta frecuencia y que no podrían ser realizadas por una persona debido a su capacidad y al hecho de que son tareas muy tediosas. El área de acción de un Bot se encuentra mayormente enfocada en Internet, sin embargo, esto no excluye su uso en sistemas no conectados a la red. Se le denomina Bot haciendo referencia a la palabra Robot.

Los Bots pueden ser desarrollados en múltiples lenguajes de programación y pueden funcionar tanto del lado del servidor como del lado del cliente dentro de una arquitectura establecida. En ocasiones los Bots suelen ser denominados como sistemas expertos ya que se especializan en realizar algunas funciones específicas.

Los Bots suelen tener diferentes fines de uso, algunos ejemplos pueden ser las arañas web implementadas por ciertos motores de búsqueda en Internet, estos Bots se encargan de navegar a través de la web recorriendo todos los elementos de un sitio web de forma automática con el fin de recolectar información de manera eficaz y muy rápida, cosa que sería prácticamente imposible para una persona. Otro ejemplo serían aplicaciones web interactivas en las cuales puedes hablar con un Bot que simula ser un humano, esto se logra mediante la implementación inteligencia artificial dentro del Bot.

En general los Bots por sus características pueden ser utilizados de una buena manera o con fines maliciosos. Cuando un Bot es utilizado para realizar tareas que no afectan a terceros y que facilitan acciones que de otro modo sería casi imposible de realizar dentro de un marco legal, entonces el Bot tiene un uso adecuado. Por otro lado cuando un Bot se desarrolla con intenciones que puedan afectar a terceros como por ejemplo recopilar información de manera automática para campañas publicitarias que posteriormente terminan en Spam, para realizar copias masivas de información que circula en la red que cuenta con derechos de autor, o cuando se utiliza para realizar ataques informáticos de forma masiva a algún servicio en línea, es entonces que el Bot se vuelve una herramienta nociva y peligrosa.

Dentro de las redes sociales, los Bots pueden ser utilizados a través de servicios de mensajería instantánea que proporcionan las mismas plataformas, generalmente se usan con la finalidad de simular interacciones humanas y brinda servicios de asistencia de manera rápida eficaz. En este caso el Bot se programa para utilizar un lenguaje natural cuando interactúa con algún usuario, brindando también la posibilidad de añadir una base de inteligencia artificial para estos fines.

La complejidad de las tareas que puede realizar un Bot solo se ve limitada por la imaginación del que lo desarrolla, ya que estos pueden realizar actividades muy básicas como dar likes en una red social hasta funcionar como un asesor que brinda asistencia técnica al usuario.

En la actualidad los Bots representan un área con amplias oportunidades para la innovación, ya que empresas importantes han comenzado a incursionar en sus usos y aplicaciones. Los Bots son una tecnología que se encuentra en desarrollo y que presenta alternativas eficaces para solucionar ciertos problemas.

3.5 FACEBOOK MESSENGER.

Facebook Messenger es una aplicación que brinda servicio de mensajería instantánea para los usuarios de Facebook. Este servicio se implementó como una aplicación independiente a la aplicación de la red social Facebook en el año 2010, la aplicación se encuentra disponible para sistemas operativos iOS y Android. Del mismo modo Facebook Messenger cuenta con un sitio web dedicado para la plataforma.

El hecho de que Facebook separara su aplicación principal que contiene la red social de los servicios de mensajería obliga al usuario a descargar ambas aplicaciones de manera independiente si quiere contar con todos los beneficios que Facebook ofrece.

La plataforma de Facebook Messenger permite a los usuarios de la red social Facebook intercambiar mensajes entre sí a través de internet. Además de los servicios de texto plano, Messenger cuenta con los complementos de llamadas de voz y video llamadas, que se pueden efectuar individualmente o en grupo. Dentro de la aplicación para Android se cuenta con soporte integrado para mensajes de tipo SMS. La aplicación general presenta una navegación eficiente y de fácil uso, esta permite abrir varias cuentas de forma paralela, así como la minimización de ventanas de conversación que se ejecutan en segundo plano. En la parte de seguridad la plataforma ofrece opciones de cifrado de extremo a extremo.

La plataforma también ofrece opciones de entretenimiento como la posibilidad de ejecutar juegos especiales que han sido seleccionados por la empresa. También se ofrece la capacidad de enviar archivos adjuntos, videos, así como imágenes, o formatos del tipo GIF. Una de las características llamativas es la posibilidad de enviar emojis que en la actualidad son muy recurrentes para expresar sentimientos y emociones a través de los mensajes.

Facebook en tiempos comenzó a permitir que las empresas y los usuarios puedan utilizar funciones especiales a través de Messenger mediante las cuales se les puede dar seguimiento a una compra, realizar pagos, así como la asistencia técnica y atención al cliente de manera automatizada.

También Facebook permite que desarrolladores independientes puedan integrar sus desarrollos en Messenger, esto con la finalidad de que los usuarios puedan utilizar sus aplicaciones favoritas dentro de la aplicación de Messenger. Más recientemente en el año 2016 Facebook introdujo su API para desarrolladores la cual permite crear Bots dentro de la plataforma, los cuales deberán cumplir con las condiciones de uso de la red social para garantizar el correcto uso de esta tecnología y beneficiar de este modo a los usuarios. Adicionalmente Facebook ha incentivado el desarrollo de tecnologías con Bots implementando extensiones dentro de la plataforma para interactuar con estos, así también integrando un buscador específico para ver los Bots disponibles dentro de la plataforma.

En la actualidad Facebook Messenger cuenta con un total de 1.2 mil millones de usuarios, lo cual ha sido medido en el mes de abril de 2017.

CAPÍTULO 4

CONGRUENCIA METODOLÓGICA

CAPÍTULO 4. CONGRUENCIA METODOLÓGICA.

TEMA: Desarrollo de sistema móvil para brindar soluciones eficaces contra la depresión y sus consecuencias a nivel social, mediante la implementación de Bots y sus algoritmos de acción basados en la interacción con él usuario para obtener datos estadísticos generados por el usuario a través del uso de la propia plataforma, que se convertirán en información útil mediante su representación gráfica.

PROBLEMA: La depresión es un trastorno del estado de ánimo de una persona que es bastante frecuente en la actual sociedad en la que vivimos. En algunas ocasiones, la depresión y sus síntomas pueden llegar a desarrollarse crónica o recurrentemente, impidiendo el buen progreso de la vida de una persona, ya sea en el trabajo, el entorno social o familiar. Este problema afecta directamente el desarrollo de una sociedad general y sus índices de productividad.

A continuación, se presenta la matriz metodológica que aborda los objetivos generales del proyecto que se desarrolla para dar solución al problema planteado, de igual forma se ven plasmados los objetivos específicos que se enfocan más al aspecto técnico y de desarrollo de este caso de estudio.

Para cada objetivo se consideran las hipótesis que son las que dan origen al caso de estudio y cada segmentación para deducir los objetivos tanto generales como específicos, del mismo modo se exponen las variables que intervienen en la problemática en cuestión, así como los indicadores que se tendrán como referencia para poder realizar el análisis correcto de cada objetivo planteado.

4.1 MATRIZ METODOLÓGICA.

La siguiente tabla representa la matriz metodológica asociada al desarrollo del sistema, especificando tanto los objetivos generales, así como los específicos tomando en cuenta las hipótesis correspondientes para cada caso, se mencionan también las variables que se encuentran involucradas y los indicadores que sirven como análisis.

Tabla 4.1 Matriz metodológica.

OBJETIVOS	HIPOTESIS	VARIABLES	INDICADORES
<u>GENERAL:</u> Determinar los factores causales de la depresión en la actualidad.	El desarrollo y continuidad de la depresión se produce por la combinación del estrés y de emociones negativas persistentes en la vida de una persona.	Estado de ánimo. Factores bioquímicos. La personalidad.	Cantidad de actividades realizadas en el día. Promedio de conversaciones con amistades de la persona objeto de estudio.
<u>GENERAL:</u> Detectar síntomas de depresión y sus variantes.	Sensaciones como: menor energía para llevar a cabo las labores rutinarias, pérdida de interés en el sexo, sensación de inquietud y nerviosismos, despertares nocturnos o excesivo sueño, molestias en la garganta, aumento del ritmo cardíaco, etc.	Estado de ánimo. Factores personales (despido, pérdidas personales).	Edad. Desarrollo laboral. Familia. Alimentación.
<u>GENERAL:</u> Soluciones viables para combatir los síntomas de depresión.	En vista del deterioro neurocognitivo y neurofisiológico que ocasiona la depresión y en vista de la alteración del funcionamiento diario y de las funciones psicosociales, ambos asociados con la función cognitiva, es aconsejable buscar medios para preservar dicha función cognitiva, con la incentivación de crear nuevas relaciones	Círculo social de la persona. Factores físicos (clima, alimentación).	Estadística de ánimo generada por la plataforma.

	interpersonales o fortalecer las que ya se tienen. Proponer rutinas cambiantes e interesantes para la persona a través de múltiples consejos.		
ESPECIFICO: Desarrollo de la aplicación móvil interactiva con el usuario.	El usuario necesita una aplicación con interfaz amigable y de fácil intuición para que su uso sea constante y no invasivo.	Plataforma de desarrollo para la aplicación (Web, Android Studio). Sistema de base de datos tipo NoSQL (Firebase).	Técnicas de desarrollo web. Lenguajes de programación.
ESPECIFICO: Desarrollo de un Bot para mensajería instantánea y recolección de datos.	El usuario debe interactuar con la aplicación mediante una interfaz que le resulte familiar como la mensajería instantánea (Facebook-Messenger).	Lenguaje de programación para Bot (JavaScript). Diferentes IDE's para el óptimo desarrollo del Bot.	Nivel de automatización del Bot. Programación estructurada con flujo lógico para la interacción del Bot con el usuario. Almacenamiento de datos en la nube.

Finalmente se realizará un periodo de pruebas con la aplicación en producción para garantizar la eficacia y funcionalidad de todas las implementaciones específicas y generales.

4.2 MODELO DE DESARROLLO.

El desarrollo de la aplicación se realiza mediante la implementación de una metodología ágil que nos proporciona un marco de trabajo que nos ayuda a estructurar, planificar, probar y controlar el desarrollo de la aplicación durante todas sus fases desde el comienzo y hasta el despliegue de la aplicación final.

En este caso se utiliza el Desarrollo Rápido de Aplicaciones (RAD: Rapid Application Development por sus siglas en inglés) que se adecua de manera fácil al nivel de desarrollo con el que se trabaja ya que esta metodología permite el análisis y desarrollo aplicable iterativo durante todas las fases del desarrollo, agilizando de este modo procesos de programación y pruebas para mantener una alta calidad durante todo el periodo de desarrollo.

Debido a que la aplicación es de bajo presupuesto, ya que herramientas de software como API's especializadas para trabajar con Bots proporcionadas por empresas líderes tienen costos elevados, se enfocan los recursos en la rápida implementación dentro del ambiente de desarrollo gratuito que nos proporciona Android Studio, así mismo se utiliza como marco de trabajo Visual Studio y apoyo del Microsoft Bot Framework para la parte de modelado y prueba del Bot. A grandes rasgos el lenguaje de programación que se utiliza es Java, Javascript dentro del servidor con la implementación de Node.js para los diferentes módulos que presenta la aplicación.

Finalmente, para la parte de almacenamiento y administración de datos se utiliza una base de datos de tipo NoSQL que nos ofrece FireBase de Google, por su alta escalabilidad horizontal para trabajar con múltiples usuarios y su fácil implementación dentro de las aplicaciones móviles para Android, así como para aprovechar las cualidades de data en tiempo real. Cabe mencionar que los datos que se almacenaran dentro de la base representan en su mayoría datos estadísticos e información personal de dominio público ya que son recolectados de Facebook con base en sus políticas de uso; debido a esto la información puede ser consultada por cualquier entidad que lo requiera. La seguridad de la propia base de datos de implementará con base a las reglas de seguridad ofrecidas por Firebase. Todos estos aspectos se explicarán y detallarán a fondo posteriormente.

Con el fin de producir un sistema de calidad con un desarrollo rápido se realizará iteración de prototipos con la participación de un usuario simulado, para de este modo poder obtener constantes mejoras, tanto en la lógica de programación para el Bot como en la interfaz de interacción con el usuario. Principalmente deben establecerse las prioridades dentro del desarrollo para avanzar de forma ordenada y sistemática.

A continuación, se muestra de manera gráfica la metodología a seguir para el desarrollo de la aplicación exponiendo así los procesos a realizar durante todas las etapas del desarrollo y hasta la implementación.

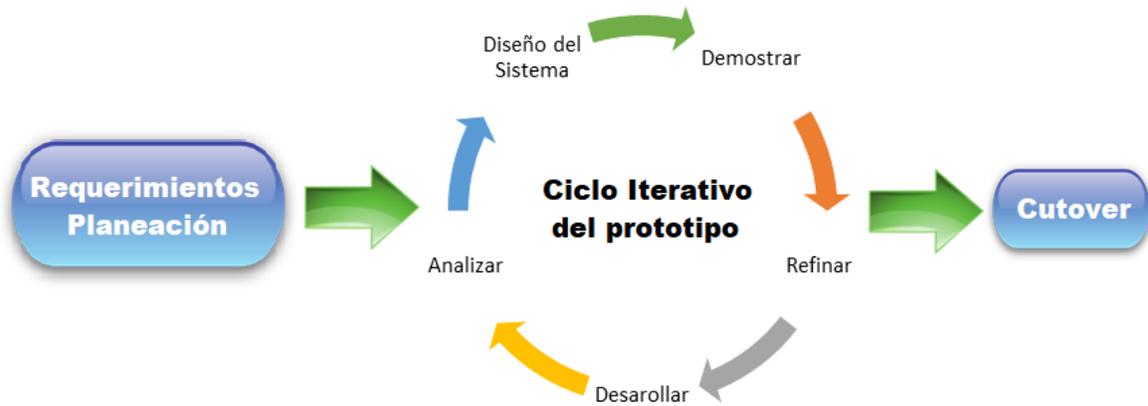


Figura 4.1 Modelo ágil tipo Desarrollo Rápido de Aplicaciones a utilizar.

4.3 ETAPAS DEL MODELO RAD.

Como se explicó anteriormente el modelo a utilizar es el que nos brinda la capacidad de implementar la metodología RAD (Rapid Application Development) que a su vez proporciona un marco de trabajo fácilmente adaptable al desarrollo de la aplicación que se ha propuesto. Los beneficios de utilizar esta metodología son amplios y convenientes para este caso particular, ya que nos permite el desarrollo interactivo de la aplicación con el fin de realizar mejoras de calidad constantemente de una manera rápida y eficiente.

A continuación, se explicará a detalle cada una de las etapas del modelo propuesto para el desarrollo de la aplicación, englobando así ciertas especificaciones que son necesarias para el correcto desempeño de la aplicación, así como el eficaz desarrollo del proyecto considerando la interacción con el usuario.

4.3.1 REQUERIMIENTOS.

En esta sección se realizará un desglose general de todos los requerimientos con los que debe contar la aplicación para que cumpla con los objetivos planteados, de este modo se definirán las funciones y procesos que realiza la aplicación a nivel general involucrando los diferentes módulos que la componen como son, la base de datos en tiempo real (Firebase), módulo y estructura del Bot, así como la interfaz de la aplicación móvil para dispositivos Android.

REQUERIMIENTOS GENERALES DE LA APLICACIÓN

A continuación, se muestra la tabla referente a los requerimientos generales de la aplicación con el nombre del requerimiento y su respectiva descripción.

Tabla 4.2 Requerimientos generales de la aplicación.

Número	Requerimiento General	Descripción
1	Aplicación móvil para dispositivos Android.	Se desarrollará una aplicación móvil que funcionará principalmente en dispositivos con sistema operativo Android, esto debido a su amplia presencia en el mercado de los dispositivos móviles, y ya que su distribución puede facilitarse a través de la tienda oficial de PlayStore.

2	Chat-bot interactivo.	<p>Se desarrollará un bot que podrá interactuar con los usuarios a través de la aplicación de mensajería instantánea de Facebook Messenger, el bot se encargará de guiar al usuario a través de todas las características que ofrece el sistema, así como también será el encargado de realizar inserciones y consultas a la base de datos.</p> <p>Este Bot debe interactuar con el usuario de una forma poco invasiva y de fácil manejo, para que el usuario hable con el intuitivamente.</p>
3	Administración de usuarios	<p>Cada usuario contará con un Identificador Único que servirá para acceder a la información estadística que se genere a través de la interacción con el Bot, así mismo el Bot utilizará este identificador para reconocer a cada usuario de manera única y de este modo poder recolectar sus datos para inserción a la base de datos para su posterior manejo y análisis.</p>
4	Recolección de información personal.	<p>El Bot tendrá la capacidad de recolectar los datos del usuario que interactúe con él a través del servicio de mensajería escogido (Messenger), con la implementación de formularios interactivos a través de una secuencia lógica de mensajes poco invasivos para el usuario.</p>
5	Consulta de información personal importante.	<p>La aplicación móvil permitirá al usuario que por medio de su identificador único pueda consultar su información relevante personal de manera eficaz, la cual podrá ser su índice de felicidad, momentos felices, momentos tristes, entre otros.</p>
6	Consulta de información global importante.	<p>La aplicación móvil permitirá a cada uno de los usuarios consultar la información global y estadísticas producidas en base a la información general</p>

		proporcionada por toda la comunidad de usuarios. Esto con el fin de que cada usuario pueda percibir el panorama general de cuál es su situación personal en comparación con una media global.
7	Base de datos en tiempo real.	Tanto la aplicación móvil como el módulo del Bot estarán conectados a una base de datos de tipo no relacional (Firebase) que nos brinda la opción de tener acceso a los datos en tiempo real mediante funciones permanentes de escucha a la base. Esto nos permitirá actualizar gráficas, y datos simultáneamente dentro de la interfaz de la aplicación móvil, así como en el Bot cuando este esté activo.
8	Ranking de usuarios.	La aplicación permitirá a través de un sistema de calificación, desplegar en la interfaz un ranking de los diez usuarios que presentan el mejor estado de ánimo, esto con el fin de servir como motivación e incentivar a los demás usuarios a repetir sus pasos, o ponerse en contacto con ellos.
9	Recepción de consejos.	El Bot a través de su interacción con el usuario por medio de los servicios de mensajería instantánea será capaz de recibir consejos de todos los usuarios para su almacenamiento en la base de datos con el fin de que estos consejos sirvan de ayuda para combatir los efectos de la depresión en las personas que los lean.
10	Proporcionar consejos de manera aleatoria.	El Bot que tendrá acceso a la base de datos donde previamente se han almacenado los consejos proporcionados por los usuarios, proporcionará a los usuarios que interactúen con él un consejo de manera aleatoria para que este sea calificado. De igual manera estos consejos se actualizarán y

		sincronizarán con la base de datos en tiempo real con base a las interacciones y recepciones que el Bot mantenga con toda la comunidad de usuarios.
11	Ranking de consejos.	Los usuarios a través de la interacción con el Bot podrán calificar los consejos que previamente han sido proporcionados por los usuarios, ya sea negativa o positivamente, esto con la finalidad de que en la aplicación móvil se pueda consultar un top diez de consejos que se posicionarán dependiendo de las calificaciones de los usuarios en tiempo real.
12	Regulación de consejos.	El sistema será capaz de determinar cuáles consejos resultan poco eficientes, o son de un carácter negativo en base a las calificaciones de usuarios para que sean eliminados de la base de datos y no afecten directamente a los usuarios.
13	Gráficas en tiempo real.	La aplicación móvil a través de la consulta de los usuarios deberá mostrar los datos de manera atractiva mediante la implementación de graficas que resulten de fácil interpretación para el usuario, por lo cual se usarán de gráficas circulares. Así mismo estas gráficas deberán de reflejar todos los cambios que se realicen en los datos en tiempo real, para que se haga evidente como la interacción con todo el sistema ocasiona una fluctuación directa de la información.
14	Vinculación de usuarios.	Mediante la interacción con el Bot, esté podrá sugerirle al usuario la interacción con otro usuario de la aplicación. Con este fin, el Bot proporcionará la información necesaria de un usuario a otro para que estos puedan entrar en contacto y comunicarse, con el fin de compartir experiencias, o sentimientos creando un vínculo entre los usuarios

		de la aplicación que persiguen un objetivo único, el cual es el bienestar emocional común.
15	Servidor en la nube.	Se montará el módulo del Bot en un servidor en la nube lo cual garantizará el funcionamiento constante del Bot para su interacción con el usuario a cualquier hora del día, en cualquier día.

REQUERIMIENTOS NO FUNCIONALES.

A continuación, se muestra una tabla que engloba todos los requerimientos no funcionales que tendrá la aplicación para su correcto funcionamiento y desempeño. Estos requerimientos deberán ser tomados en cuenta por todas las personas que interactúen con la aplicación, ya que podrían resolver algunas dificultades de uso que pudieran presentarse.

Tabla 4.3 Requerimientos generales de la aplicación.

Número	Requerimiento no funcional	Descripción
1	Conexión a Internet.	El usuario que haga uso de la aplicación deberá estar conectado a alguna red que le proporcione acceso a Internet, ya que para interactuar con el Bot a través de Messenger como para acceder a los datos y funciones de la aplicación móvil esto es necesario. Esto es debido a que el Bot y toda su programación se encontrarán alojadas en un servidor en la nube, donde estará ejecutándose permanentemente para su funcionamiento. Por otro lado, la aplicación accede a los datos generados por los usuarios a través de internet, ya que la base de datos se encuentra en la nube de Firebase que nos permite su acceso en cualquier momento.
2	Página de Facebook para la aplicación.	La aplicación necesita contar con una página de Facebook, para que esta actúe como la entidad intermediaria con la cual el usuario podrá

		comunicarse. El Bot es el encargado de gestionar los mensajes recibidos a través de la página y contestar con respecto a su programación. Es justamente aquí y por este medio que el Bot interactuará con los usuarios.
3	Cuenta de Desarrollador Facebook.	Para poder realizar el desarrollo del Bot con todas sus características, es necesario contar con una cuenta de Desarrollador Facebook, la cual podremos obtener en el sitio oficial de Facebook. Esto es necesario para poder hacer uso de las API's proporcionadas para trabajar con Facebook Messenger, así como para firma de acuerdo con las condiciones de uso de la aplicación.
4	Webhook.	Se requiere la configuración de un Webhook para la aplicación y que este se aloje en un servidor público que soporte HTTPS, un certificado valido de SSL y un puerto para soporta peticiones múltiples al servidor.
5	Perfil de Facebook.	Por parte del usuario es necesario que éste cuente con un Perfil de Facebook, para poder interactuar con la aplicación y el Bot, ya que éste último se encargará de recabar los datos en base a la información que esté presente en el perfil de Facebook del usuario con el que interactúe.
6	Facebook Messenger.	A pesar de que el Bot será capaz de realizar todas sus funciones a través de una interfaz web, se recomienda fuertemente que el usuario cuente también con la aplicación oficial de Facebook Messenger ya que de este modo el Bot podrá implementar todas sus características sin contrariedades. Esto debido a que Facebook presenta ciertas irregularidades con su servicio de mensajería en la interfaz web.

7	Seguridad.	Al comenzar la interacción con el Bot se dará por entendido que el usuario admite y autoriza el uso de su información personal y datos generados con fines de análisis para el desarrollo de la aplicación. No habrá sistema de login ya que se pretende que la aplicación sea de libre acceso a información con la finalidad de fomentar la interacción de usuarios.
8	Plataforma de desarrollo.	<p>Para el desarrollo de la aplicación se utilizará una computadora con las siguientes características:</p> <ul style="list-style-type: none"> • Procesador Intel Core i5 • Memoria RAM de 8GB • Tarjeta gráfica NVIDIA GEFORCE GTX 50 • Sistema Operativo Windows 10 de 64 bits. <p>Esto con la finalidad de que el desarrollo se realice sin contrariedades al correr la herramienta básica de desarrollo Android Studio y todas sus características de diseño.</p>
9	Plataforma de ejecución.	El desarrollo de la aplicación se realizará enfocada en dispositivos móviles que cuenten con sistema operativo Android. Debido a que las pruebas se realizarán en un dispositivo que cuenta con Android 6.0 Marshmallow, se recomienda que para que la aplicación cuente con un rendimiento óptimo el usuario utilice versiones del Sistema Operativo iguales o superiores.
10	Rendimiento.	El sistema será capaz soportar el manejo e interacción con múltiples usuarios, así como la transferencia de datos e información al mismo tiempo.

11	Usabilidad.	El sistema deberá ser lo suficientemente explícito e intuitivo para que el usuario lo pueda usar fácilmente.
----	-------------	--

4.3.2 CICLO ITERATIVO DEL PROTOTIPO.

En el ciclo iterativo del prototipo se busca realizar el desarrollo de la aplicación planificada en diversos bloques que englobarán la programación completa del Bot, el diseño de la interfaz para la aplicación móvil, la estructuración e implementación de la base de datos, y la programación de todas las funciones de la aplicación.

Dentro de este ciclo se realizarán diversas acciones establecidas por bloque que de realizarse de manera consecutiva irán mejorando el funcionamiento general de la aplicación, tras la terminación del primer ciclo se volverá a iterar con base en los resultados obtenidos. De este modo se irán obteniendo prototipos a lo largo de todo el proceso, que se irán optimizando y adaptando a los requerimientos dependiendo de las necesidades o dificultades que se presenten o vayan surgiendo a lo largo de todo el desarrollo.

En la siguiente tabla, se desarrollarán los diferentes bloques que componen el ciclo iterativo, se explicarán de manera general todas las acciones a realizar en cada etapa que se relacionan directamente con el desarrollo general de la aplicación.

Tabla 4.4 Etapas del ciclo iterativo.

Etapa del ciclo iterativo	Desarrollo
Analizar	<p>Dentro de esta etapa del proyecto se busca que el desarrollador comprenda completamente cuales son los requerimientos generales y no funcionales de la aplicación que se han planteado con anterioridad.</p> <p>De la misma manera se analizará cuáles son las funciones y programas que se implementarán para el desarrollo completo de la aplicación, esto para que cumplan de una manera óptima con su función, maximizando el rendimiento y adaptándose a todos los requerimientos.</p> <p>Una vez realizada la primera iteración de todo el ciclo, al volver a esta etapa se volverá a analizar si los resultados obtenidos cumplen satisfactoriamente con todos los requerimientos planteados, de no ser así se propondrán nuevas soluciones que busquen enfocarse a la adaptación del sistema para que este se adecue a lo que se ha solicitado. Posteriormente se pasará a realizar las siguientes etapas para validar el análisis hecho y optimizar de este modo el prototipo de la aplicación con que se está trabajando.</p>
Diseño del Sistema	<p>En la parte del diseño, se plasmará de manera concreta cual será la estructura que tendrá la</p>

	<p>aplicación en general. En esta etapa se englobarán el tema de estructuración de la base de datos, para que esta funcione de una manera eficaz y óptima.</p> <p>También se establecerá la arquitectura global de la aplicación que involucra la comunicación bidireccional entre el Bot y la base de datos, así como las consultas a la base que realiza la aplicación móvil. Por otro lado, se diseñarán los algoritmos y funciones que se vayan a implementar para el correcto funcionamiento de la aplicación.</p> <p>Finalmente se elaborará el diseño de la interfaz gráfica con la que interactuará el usuario, esto para la parte de la aplicación móvil, ya que la interfaz del Bot se encuentra previamente definida por los estándares establecidos por Facebook Messenger.</p> <p>Si tras realizar una iteración no se han cumplido con los objetivos de la aplicación de manera exitosa, se optará cambiar el diseño previo del prototipo del sistema cambiando las mínimas características posibles para que se adapte a las circunstancias que hayan evitado que se concretara con el proyecto. En un caso extremo y de ser absolutamente necesario, se rediseñará todo el sistema. Todo con el fin de obtener un diseño que cumpla perfectamente con los requerimientos establecidos.</p>
Demostrar	<p>Esta etapa se base en los resultados obtenidos dentro de la etapa de diseño para comprobar que estos resultados satisfacen de manera adecuada los principales escenarios que pueden ocurrir. Se busca demostrar que el diseño propuesto se adecua a los principales escenarios que se podrían presentar y que todas las funciones se adaptan a los requerimientos que se plantearon principalmente.</p>
Refinar	<p>El propósito principal de esta etapa es refinar todos elementos que se han propuesto hasta este momento, con el fin de que estos elementos puedan coincidir de la mejor manera posible con los requerimientos. Para lograr esto, se evaluarán cada uno de los elementos en función de los eventos factibles que puedan ocurrir durante el uso de la aplicación.</p>

	<p>Al final se obtendrá el esquema final de la aplicación completa, y se podrá implementar en la parte de desarrollo.</p>
Desarrollar	<p>Durante esta etapa se inicia con el proceso de traducir todos los diseños obtenidos previamente al lenguaje máquina, esto se logrará mediante la programación del ambiente virtual que tendrá la aplicación, del mismo modo se programarán aquí todas las funciones que tendrá la aplicación utilizando el lenguaje definido con anterioridad.</p> <p>También se desarrollará el módulo del Bot y toda su programación.</p>

4.3.3 CUTOVER.

Esta es la fase final del proyecto, en la que tras realizar el ciclo iterativo la cantidad de veces necesarias para que el proyecto funcione en su totalidad abarcando todos los requerimientos y cumpliendo con los objetivos planteados inicialmente, vamos a obtener el estado final del prototipo de la aplicación.

El objetivo principal de esta fase será migrar de un estado de desarrollo en el que se cuenta con un ambiente controlado para el producto, a un estado de producción real en el que la aplicación opere dentro de un ambiente real interactuando con múltiples usuarios y estados que puedan influir en su funcionamiento.

Para poder realizar este cambio es necesario verificar ciertos aspectos importantes como constatar que todos los requerimientos se han incorporado al diseño de la aplicación y que se han implementado todos los cambios necesarios en caso de que estos se hayan hecho presentes. Se han realizado pruebas específicas para verificar el correcto funcionamiento de todas las características del sistema, así como para corroborar la entera satisfacción con respecto a la calidad del proyecto. También se deberán evaluar los resultados que la aplicación genere dentro de un ambiente simulado, ya que esto podrá darnos una idea de cuál será su desempeño al enfrentarse a un ambiente de producción real.

Por otro lado, debemos tener en consideración que los usuarios requerirán un manual o instrucciones de uso para utilizar la aplicación de la forma más correcta. Sin embargo, es de destacar que la aplicación proporcionará una interfaz simple y amigable, por lo cual no requerirá un documento de instrucciones, todos los pasos a seguir necesarios para interactuar a través de la aplicación serán desplegados por esta misma de manera sencilla y contará con una interfaz intuitiva que ayudará a la navegación del usuario a través de la aplicación para su óptimo funcionamiento.

En esta parte se realizará la documentación necesaria, para explicar los procesos y esquemas que se utilizaron al programar la aplicación, esto con el fin de que en un tiempo posterior se tenga la capacidad de escalar la aplicación para adaptarla a los nuevos cambios tecnológicos, y principalmente para realizar algún plan de mantenimiento en caso de que esto así se requiera.

Para este caso particular, la aplicación hace uso de ciertos componentes como API's proporcionadas por Facebook y servicios de almacenamiento en la nube intrínsecos de la implementación de la base de datos de Firebase que por ser de etapa de desarrollo se obtienen gratuitamente. Sin embargo, para poder llevar la aplicación a producción real, se necesita pagar por los servicios y pasar filtros de verificación establecidos por las empresas que proporcionan estos servicios que garantizan que la aplicación cumple con sus condiciones de uso.

CAPÍTULO 5

DISEÑO E IMPLEMENTACIÓN

CAPÍTULO 5. DISEÑO E IMPLEMENTACIÓN.

5.1 PLANTEAMIENTO.

Como se ha visto en el desarrollo del planteamiento del problema, los trastornos depresivos en su conjunto son un caso de estudio sumamente amplio que tiene diferentes puntos de análisis y en especial cada caso en particular de esta afección puede repercutir en la salud tanto física como mental a nivel cognitivo en la persona que la padezca de un modo diferente. Cada persona que sufre de depresión puede presentar un cuadro con características diferentes, sin embargo, se tiene la constante de un estado de ánimo de decaimiento, así como falta de entusiasmo y una reducción considerable en la eficiencia con la que la persona afectada lleva a cabo sus actividades diarias.

Es por eso por lo que al ser tan variado el caso de estudio y con múltiples ramificaciones en el tipo de cuadros sintomáticos que se pueden presentar, que se busca atacar el problema desde una perspectiva global y un modo amplio para que de esta manera la solución que se alcance logre abarcar la mayoría de los cuadros depresivos que puedan presentarse, para satisfacer de este modo las necesidades del proyecto para dar una solución adecuada al problema que se está atacando.

Para el desarrollo de la aplicación se toman en cuenta múltiples requerimientos que involucran las necesidades del usuario final, haciendo de este modo que la aplicación posea una calidad alta, así como poder dotar a la aplicación de una fácil escalabilidad y mantenimiento; esta aplicación al ser una aplicación móvil se podrá distribuir a través de la tienda oficial de PlayStore para su fácil distribución y con fines de realizar pruebas de calidad, así como análisis estadísticos para mejoras en base a comentarios de los usuarios.

Se busca que la interfaz con el usuario sea simple para que este pueda interactuar dentro de la aplicación de una forma eficaz y ágil, ayudando a que su uso sea sencillo sin convertirse en algo tedioso para incentivar el continuo y constante uso de la aplicación para optimizar los resultados que esta ofrece. Para la parte del Bot, este deberá ofrecer una interacción amigable con el usuario para ser lo menos invasivo posible, se comunicará con este a través de la plataforma de mensajería instantánea de Facebook Messenger para establecer una relación usuario-maquina eficiente, explotando también de este modo todas las posibilidades que este nos brinda como recabar información de manera automatizada, capacidad para trabajar con múltiples usuarios de manera independiente, y realizar tareas repetitivas en periodos cortos de tiempo y a largo plazo.

Adicionalmente se tendrán que considerar con base a un análisis realizado de los datos recolectados por el Bot cuál será la mejor solución a ofrecer para el usuario que padece de depresión, siendo estas soluciones propuestas de actividades a

realizar y enlazar usuarios para que se comuniquen e interactúen con el fin de promover las relaciones personales con propósitos de mejorar el estado de ánimo del usuario. Las soluciones propuestas serán variables y adaptables a cada caso particular con la opción de poderse actualizar en base a ciertos requerimientos o nuevas soluciones que se consideren más optimas, así mismo se brindará la opción de incrementar el número de soluciones factibles para que la aplicación tenga una escalabilidad adecuada y pueda crecer de manera periódica.

5.2 CASOS DE USO.

En este apartado se definen cuáles son los diferentes casos de uso que se pueden desarrollar dentro de la aplicación, se pretende utilizar un lenguaje sencillo para describir las diferentes acciones que podrá desempeñar cada actor dentro del sistema o la aplicación en general.

Esta ayudará a comprender de manera más sencilla cómo es que interactúan y se relacionan los diferentes actores a través de la aplicación, que en este caso son el usuario y el Bot. Se describirán algunas características de la aplicación, en específico las que representan una función que no está oculta al usuario, lo que significa que no se detallarán procesos de programación ni cómo es que estos están programados a fin de que los esquemas generados sirvan como una herramienta que simplifique el entendimiento de la aplicación en general.

A continuación, se muestra el diagrama de casos de uso que representa las actividades que realizan los actores dentro del sistema.

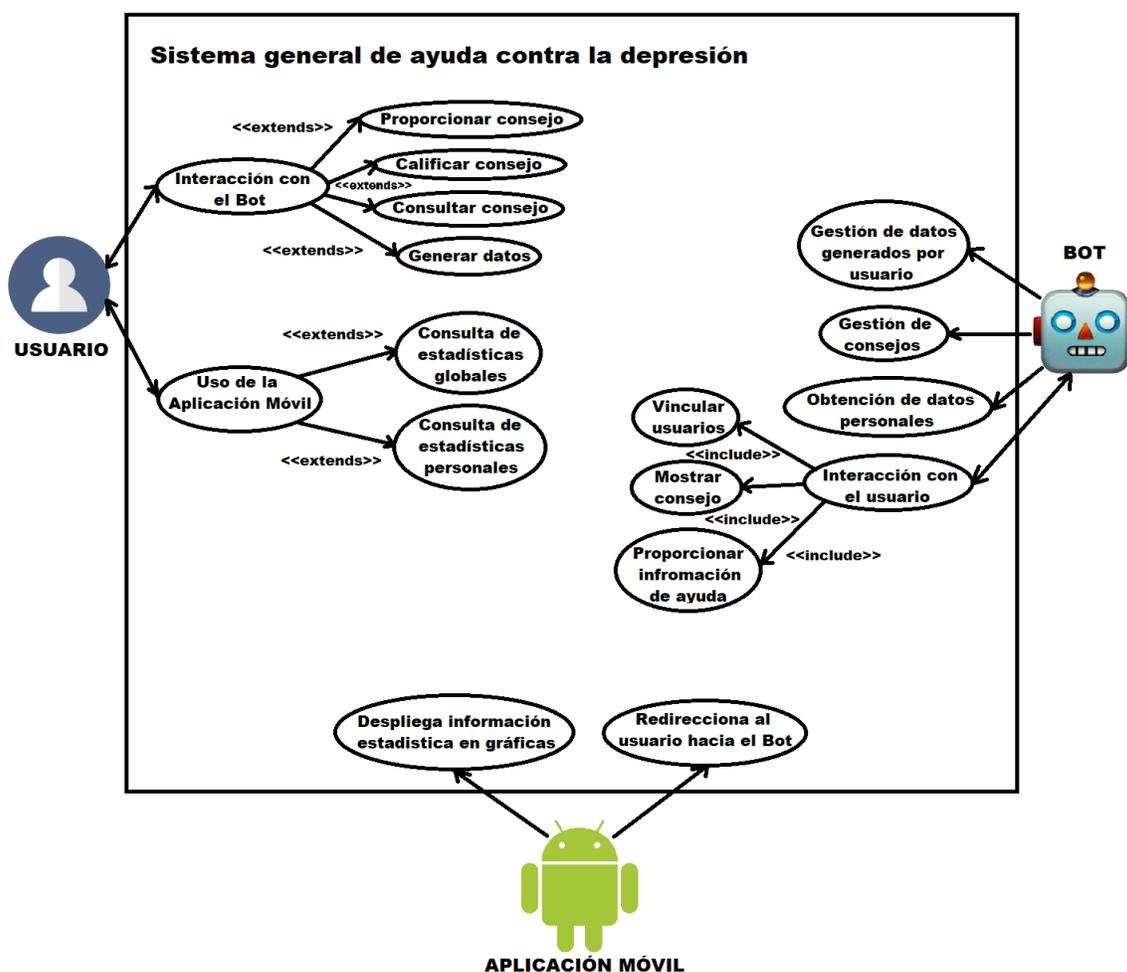


Figura 5.1 Diagrama general de casos de uso.

Ya que se tiene establecido el diagrama de casos de uso que involucra todas las acciones que pueden realizar los múltiples actores dentro del sistema, los cuales son el usuario, el Bot y la aplicación móvil. Se procede a especificar cada una de estas actividades que están involucradas dentro del sistema, para esto se detallará específicamente el diagrama de casos de uso en las siguientes tablas.

5.2.1 CASOS DE USO DEL USUARIO.

Las siguientes tablas definen de manera concreta cuales son las actividades involucradas para los casos de uso que tienen como actor principal al usuario. De la misma manera definen el flujo básico que describe el caso de uso y cómo es que este debe desarrollarse, por otro lado, en caso de que así exista se define el flujo alterno que puede tomar el caso de uso. Adicionalmente se toman en consideración cuales son las precondiciones que deben existir para que el caso de uso se desarrolle de manera correcta y si es que existe una postcondición también será mencionada y descrita de forma concreta.

Tabla 5.1 Caso de uso (usuario): Interacción con el Bot.

Nombre del caso de uso	Interacción con el Bot
Actor	Usuario
Descripción	<p>En este caso de uso el usuario será capaz de interactuar con el Bot a través del servicio de mensajería instantánea de la aplicación oficial de Facebook Messenger.</p> <p>Para la parte de la interacción, esta podrá llevarse a cabo de a través de mensajes de texto, o interactuando con el Bot a través de las plantillas que este vaya proporcionando al usuario.</p> <p>Dentro de este caso de uso se proporcionarán los datos que posteriormente serán utilizados para generar estadísticas y gráficas relevantes para el análisis del estado de ánimo tanto individual por usuario, como a nivel global de la comunidad de usuarios.</p> <p>Dentro de esta interacción con el Bot, el usuario podrá además realizar opciones de consulta, en el caso de que quiera leer algún consejo, de igual manera podrá calificar el consejo que el Bot le haya proporcionado, con la finalidad de ir midiendo la eficacia de los consejos.</p> <p>También podrá brindar algún consejo, que será almacenado en la base de datos y podrá ser</p>

	mostrado a la comunidad de usuarios para su evaluación.
Flujo básico	<ol style="list-style-type: none"> 1. El usuario abre una conversación en Facebook Messenger a través de la cual podrá comunicarse con el Bot por medio de la página de Facebook que funge como entidad del Bot. 2. El Bot despliega una bienvenida en la ventana de conversación que tiene con el usuario en la cual se indican los pasos a seguir para poder interactuar correctamente con el Bot. El primer paso del usuario es presionar un botón de “Empezar”, con lo cual se toma por entendido que el usuario acepta la interacción con el Bot y permite las acciones que este pueda realizar, por lo tanto, se registra al usuario en el sistema. 3. El usuario verá en la ventana de conversación la información relevante personal como el Identificador Único que servirá para que este consulte posteriormente sus estadísticas personales de estado de ánimo directamente en la aplicación móvil. 4. Se mostrará dentro de la ventana de conversación una plantilla con botones donde el usuario deberá seleccionar su estado de ánimo actual. 5. Después de haber seleccionado su estado de ánimo se mostrará en la ventana de conversación una platilla con botones donde el usuario podrá seleccionar una de las actividades a realizar dependiendo del estado de ánimo que presenta actualmente.
Flujo alternativo	<ul style="list-style-type: none"> • El usuario inicia una conversación sin presionar el botón de “Empezar”. Se entiende entonces que el usuario no acepta los términos de interacción con el Bot y por lo tanto no se registra en el sistema.

	<ul style="list-style-type: none"> • El usuario manda un mensaje de texto, un archivo adjunto (imagen, enlace, etc.) sin haber aceptado la interacción con el Bot, entonces el usuario recibirá un mensaje por parte del Bot invitándolo a que acepte la interacción para que pueda proporcionarle todos los servicios del sistema y se pueda interactuar de forma correcta. • El usuario ya está registrado y con anterioridad ha aceptado la interacción con el Bot. Al abrir una nueva conversación gozará de los servicios y funciones que proporciona el Bot a través de la interacción.
Pre-condición	<ul style="list-style-type: none"> • El usuario debe contar con una conexión a internet para que pueda interactuar con el Bot. Esto es algo necesario ya que adicionalmente los servicios de mensajería instantánea como Facebook Messenger requieren una conexión para poder ser utilizados. • El usuario debe tener un perfil de Facebook que le servirá para poder hacer uso de la aplicación de Facebook Messenger a través de la cual interactuará con el Bot.
Post-condición	<ul style="list-style-type: none"> • Los datos generados por el usuario serán registrados en la base de datos.

Tabla 5.2 Caso de uso (usuario): Proporcionar consejo.

Nombre del caso de uso	Proporcionar consejo
Actor	Usuario
Descripción	<p>El usuario podrá a través de la interacción con el Bot en una conversación de Facebook Messenger proporcionar algún consejo. Este consejo deberá realizarse con fines de ayuda a los demás usuarios para mejorar su estado de ánimo.</p> <p>Una vez que el usuario proporciona algún consejo, este se almacenará en la base de datos para que</p>

	<p>posteriormente pueda ser mostrado de forma aleatoria a diferentes y múltiples usuarios que se encuentren interactuando con el Bot.</p> <p>Estos consejos se someterán a un proceso de evaluación por parte de los usuarios en el cuál podrán ser calificados tanto positiva como negativamente, esto para medir que tan efectivo resulta ser el consejo, además de que servirá para generar un top de los mejores consejos y del mismo modo poder eliminar los consejos que no sean útiles o resulten perjudiciales para la comunidad de usuarios.</p>
Flujo básico	<ol style="list-style-type: none"> 1. El usuario llega a la opción de proporcionar consejo dentro de la ventana de conversación de Facebook Messenger a través de la previa interacción con el Bot. Para llegar a este punto, anteriormente tuvo que seleccionar cuál era su estado de ánimo actual, con el fin de evitar que los consejos proporcionados por los usuarios sean de un carácter negativo, solo se le permitirá proporcionar consejos a los usuarios que presenten un estado de ánimo positivo o neutral. Ya que se considera que los usuarios con un estado de ánimo negativo no se encuentran en condiciones óptimas para proporcionar un comentario eficaz. 2. El usuario presiona el botón con la opción de proporcionar un consejo. El Bot despliega en la ventana de conversación las instrucciones para que el usuario proporcione efectivamente su consejo. 3. El usuario manda su consejo siguiendo las instrucciones proporcionadas por el Bot a través de un mensaje de texto dentro de la conversación.
Flujo alterno	<ul style="list-style-type: none"> • El usuario ingresa un consejo que no es válido por no ser de tipo texto (como un archivo adjunto, enlace, etc.). Entonces el Bot no registra el consejo y salta a otra parte de la interacción.

Pre-condición	<ul style="list-style-type: none"> El usuario debe haber interactuado previamente con el Bot con lo que se aceptan la interacción y todas las acciones que pueda realizar el Bot.
Post-condición	<ul style="list-style-type: none"> Los consejos proporcionados por el usuario serán almacenados en la base de datos.

Tabla 5.3 Caso de uso (usuario): Calificar consejo.

Nombre del caso de uso	Calificar consejo
Actor	Usuario
Descripción	<p>Tras la interacción con el Bot a través de la aplicación de mensajería instantánea de Facebook Messenger, el usuario podrá calificar los consejos que le muestre el Bot después de que los haya solicitado. Estos consejos se adquieren tras una consulta realizada a la base de datos por parte del Bot y se muestran al usuario como un mensaje de texto en la ventana de conversación.</p> <p>La finalidad de calificar el consejo es evaluar la eficacia de este para poder mostrar los mejores consejos a la comunidad de usuarios a través de la aplicación móvil. Por otro lado, en base a las calificaciones se determinará si un consejo resulta ser perjudicial para los usuarios y se eliminará definitivamente de la Base de Datos.</p>
Flujo básico	<ol style="list-style-type: none"> 1. El usuario podrá tener la opción de calificar un consejo dentro de la ventana de conversación de Facebook Messenger a través de la previa interacción con el Bot en la que se solicite la consulta de un consejo. Tras haber visualizado el consejo que el Bot despliega en la ventana de conversación, el usuario podrá calificar positiva o negativamente mediante una plantilla con botones desplegada por el Bot. 2. El usuario presiona una vez la calificación que otorga al consejo y continúa a otra parte de la interacción con el Bot.

Flujo alterno	<ul style="list-style-type: none"> • El usuario puede omitir dar una calificación a cierto consejo, simplemente realizando alguna de las otras actividades con las que puede interactuar con el Bot. • El usuario solo puede calificar una vez el consejo que el Bot le ha proporcionado. Por lo tanto, al querer volver a calificar un consejo, el Bot le dará al usuario la advertencia de que ya se ha calificado ese consejo y solamente se tiene permitido calificarlo una vez.
Pre-condición	<ul style="list-style-type: none"> • El usuario debe solicitar la consulta de un consejo a través de la interacción con el Bot, para poderlo calificar.
Post-condición	<ul style="list-style-type: none"> • La calificación del consejo queda registrada en la base de datos. • Los consejos se posicionan en el top de consejos dentro de la aplicación móvil dependiendo de las calificaciones obtenidas. Y los consejos negativos son eliminados definitivamente de la base de datos.

Tabla 5.4 Caso de uso (usuario): Consultar consejo.

Nombre del caso de uso	Consultar consejo
Actor	Usuario
Descripción	Después de una previa interacción con el Bot a través de la aplicación de mensajería instantánea de Facebook Messenger el usuario podrá consultar consejos que le serán de utilidad para mejorar su estado de ánimo. Estos consejos son previamente ingresados al sistema por la misma comunidad de usuarios. Y son mostrados por el Bot mediante un mensaje de texto en la conversación con el usuario, este consejo que se muestra es obtenido de manera aleatoria de la base de datos con la finalidad de que todos los consejos tengan la misma oportunidad de ser leídos por la comunidad de usuarios y de este

	<p>modo un usuario tenga múltiples opciones de consejos a recibir.</p> <p>Todos los usuarios tendrán la opción de recibir un consejo sin importar su estado de ánimo, ya que esto ayudará a mejorar su ánimo actual.</p>
Flujo básico	<ol style="list-style-type: none"> 1. El usuario tendrá en pantalla una plantilla con botones en la que se incluye la opción de consultar algún consejo tras la previa interacción con el Bot. 2. El usuario presiona la opción que le permite consultar un consejo y recibe el consejo por parte del Bot como un mensaje de texto.
Flujo alternativo	Sin flujo alternativo.
Pre-condición	<ul style="list-style-type: none"> • El usuario debe realizar la interacción previa con el Bot para poder consultar el consejo. • Deben existir consejos en la Base de Datos para poder ser proporcionados al usuario.
Post-condición	<ul style="list-style-type: none"> • El usuario podrá calificar el consejo que se le proporcionó a través del Bot.

Tabla 5.5 Caso de uso (usuario): Generar datos.

Nombre del caso de uso	Generar datos
Actor	Usuario
Descripción	<p>Mediante la interacción con el Bot a través de la aplicación de mensajería instantánea de Facebook Messenger, el usuario podrá generar datos personales que se almacenarán en la base de datos para su posterior uso por parte de la aplicación móvil y el mismo Bot.</p> <p>El usuario tras pulsar la opción “Empezar” y haber aceptado de este modo todas las funciones que puede realizar el Bot por medio de la interacción con el usuario, comenzará a generar datos que se asociarán a una entidad propia dentro de la base de datos.</p>

	Estos datos se generarán por medio de la interacción con plantillas que el Bot desplegará en la ventana de conversación.
Flujo básico	<ol style="list-style-type: none"> 1. El usuario pulsa el botón “Empezar”, y el Bot comienza a recolectar los datos básicos del usuario a través de su perfil de Facebook y de igual modo le asignará un Identificador Único que servirá para identificar a cada usuario dentro de la aplicación. Por lo tanto, el usuario de este modo genera los datos de nombre de perfil de usuario y su foto. 2. El usuario interactuará con el Bot el cuál le proporcionará una plantilla con botones a través de la ventana de conversación donde se le cuestionará sobre su estado de ánimo. Tras pulsar alguna de las opciones de la platilla, se generarán los datos sobre su estado de ánimo que se almacenarán directamente en la base de datos, para su posterior uso. Con el uso repetitivo de esta función el usuario irá modificando sus datos actuales dentro de la base de datos, conforme los vaya generando con la interacción del Bot.
Flujo alterno	<ul style="list-style-type: none"> • El usuario inicia una conversación sin presionar el botón de “Empezar”. Se entiende entonces que el usuario no acepta los términos de interacción con el Bot y por lo tanto no se genera ninguna clase de datos. Entonces el Bot recomienda al usuario aceptar las interacciones al presionar “Empezar” para poder generar datos.
Pre-condición	<ul style="list-style-type: none"> • El usuario debe haber interactuado previamente con el Bot de modo que se acepten la interacción y todas las acciones que pueda realizar el Bot.
Post-condición	<ul style="list-style-type: none"> • Todos los tipos de datos que se generen son almacenados en la base de datos para cada usuario correspondiente.

Tabla 5.6 Caso de uso (usuario): Uso de la aplicación móvil.

Nombre del caso de uso	Uso de la aplicación móvil
Actor	Usuario
Descripción	El usuario podrá hacer pleno uso de la aplicación móvil que corre bajo un sistema operativo Android. Esta aplicación tendrá varios menús para navegar a través de ella, y estará conectada directamente a la base de datos. El usuario podrá navegar a través de todas las opciones de la aplicación conforme mejor le parezca.
Flujo básico	<ol style="list-style-type: none"> 1. El usuario abre la aplicación desde un dispositivo móvil, y se encuentra con la interfaz de usuario que le proporcionará las instrucciones necesarias para su uso. 2. El usuario navega a través de las diferentes actividades que presenta la aplicación.
Flujo alternativo	No hay flujo alternativo.
Pre-condición	<ul style="list-style-type: none"> • El usuario deberá haber instalado previamente la aplicación en su dispositivo móvil. • El dispositivo móvil del usuario deberá contar con sistema operativo Android (preferentemente Android 6.0 Marshmallow, esto con la finalidad de que tenga la mayor compatibilidad posible con la programación y funciones de la aplicación).
Post-condición	No hay post-condición.

Tabla 5.7 Caso de uso (usuario): Consulta de estadísticas globales.

Nombre del caso de uso	Consulta de estadísticas globales
Actor	Usuario

Descripción	El usuario por medio de la aplicación móvil podrá consultar información relevante, en forma de gráficas y listas. Esta información consultada es de un carácter global, lo que significa que corresponde a las estadísticas generadas por toda la comunidad de usuarios de la aplicación y que interactúan con el Bot.
Flujo básico	<ol style="list-style-type: none"> 1. El usuario entra al menú de estadísticas de la aplicación móvil y selecciona la opción de estadísticas globales. 2. El usuario está en el menú de estadísticas globales y puede seleccionar entre las tres opciones diferentes que se encuentran disponibles, las cuales son tops de usuarios, top de consejos y gráficas de ánimo globales. El top de usuarios despliega en pantalla un top 10 de los usuarios que conforme a los datos que han generado tras la interacción con el Bot, presentan un estado de ánimo altamente positivo. El top de consejos despliega en pantalla un top 10 de los consejos mejores calificados por los usuarios. En las gráficas globales se podrá analizar mediante gráficos de tipo circular y otros datos, el nivel de ánimo que presenta la comunidad de usuario en tiempo real. 3. El usuario selecciona cualquiera de las opciones y visualiza todas las características que presenta la opción que seleccionó como se describió anteriormente.
Flujo alterno	No hay flujo alterno.
Pre-condición	<ul style="list-style-type: none"> • Para poder desplegar tanto el top de usuarios como el top de consejos, deberá haber registrados en la base de datos como mínimo diez consejos, y diez usuarios. En otro caso la visualización de estos datos no será posible. • El usuario debe tener instalada la aplicación en su dispositivo móvil.

Post-condición	No hay post-condición. (Se recomienda que el usuario interprete la información que se le proporciona de la manera que mejor le convenga para mejorar su estado de ánimo).
-----------------------	--

Tabla 5.8 Caso de uso (usuario): Consulta de estadísticas personales.

Nombre del caso de uso	Consulta de estadísticas personales
Actor	Usuario
Descripción	El usuario por medio de la aplicación móvil podrá consultar la información relevante personal que ha generado a través de la interacción con el Bot, en forma de gráficas y datos numéricos que representan índices. Esta información consultada es de un carácter personal y para acceder a ella es necesario proporcionar el identificador único generado por el Bot, esto no limita que los usuarios puedan compartir su Identificador Único con otros usuarios con la finalidad de incentivar las relaciones interpersonales, y que se promueva un proceso empático entre aquellas personas que consulten datos ajenos, promoviendo de este modo un mejor entendimiento del estado de ánimo de las personas que utilizan la aplicación.
Flujo básico	<ol style="list-style-type: none"> 1. El usuario entra al menú de estadísticas de la aplicación móvil y selecciona la opción de estadísticas personales. 2. El usuario entra a una interfaz donde se solicita el Identificar Único para acceder las estadísticas personales del usuario asociado a ese Identificador Único. 3. Tras haber proporcionado un identificador único, la aplicación muestra al usuario en forma de gráfico circular los datos relevantes sobre el usuario del cual se proporcionó el Identificador Único, así mismo se mostrarán datos como el estado de ánimo actual, y el índice de ánimo que el usuario tiene.

Flujo alterno	<ul style="list-style-type: none"> • El usuario proporciona un Identificador Único que no es el suyo y por lo tanto consulta estadísticas ajenas. • El usuario proporciona un Identificador Único equivocado, por lo tanto, la aplicación no despliega información e indica el error.
Pre-condición	<ul style="list-style-type: none"> • El usuario deberá contar con un Identificador Único para poder consultar este tipo de información. • El usuario deberá haber interactuado previamente con el Bot para poder generar los datos que serán representados estadísticamente por medio de gráficas.
Post-condición	<p>No hay post-condición. (Se recomienda que el usuario interprete la información que se le proporciona de la manera que mejor le convenga para mejorar su estado de ánimo o ayudar a algún usuario ajeno).</p>

5.2.2 CASOS DE USO DEL BOT.

Las siguientes tablas definen de manera concreta cuales son las actividades involucradas para los casos de uso que tienen como actor principal al Bot. De igual manera que se realizó con las tablas para los casos de uso del usuario, también se definen el flujo básico, flujo alternativo, precondiciones y post condiciones, para el respectivo caso de uso.

Tabla 5.9 Caso de uso (Bot): Gestión de datos generados por usuario.

Nombre del caso de uso	Gestión de datos generados por usuario
Actor	Bot
Descripción	El Bot es el encargado de gestionar todos los datos que se generen durante la interacción con el usuario a través de la aplicación de mensajería instantánea de Facebook Messenger. Principalmente se encargará de almacenar los datos que el usuario genera en la base de datos en tiempo real, de manera que el cambio en estos datos se vea instantáneamente reflejado en la base de datos y que todas las interacciones que se tengan con estos datos se actualicen en tiempo real con la finalidad de poder visualizar y analizar fluctuaciones en el estado de ánimo tanto global como a nivel personal.
Flujo básico	<ol style="list-style-type: none"> 1. El Bot despliega en pantalla las diferentes plantillas que mediante la interacción con el usuario generarán datos tanto globales, como personales. 2. Tras la interacción con el usuario, el Bot ingresará a la base de datos y actualizará todos los datos que se estén gestionando en ese momento.
Flujo alternativo	<ul style="list-style-type: none"> • El Bot no ingresa datos a la base si el usuario no es alguien que aceptó los términos de interacción mediante el botón “Empezar”
Pre-condición	<ul style="list-style-type: none"> • El Bot deberá estar alojado en un servidor que se encuentre activo las 24 horas del día para que este cuente con una alta disponibilidad para interactuar con los

	usuarios. Para esto se pretende utilizar un servicio de plataforma en la nube.
Post-condición	<ul style="list-style-type: none"> • Todos los datos quedarán registrados y actualizados en la base de datos, tanto los de carácter global como los de carácter personal.

Tabla 5.10 Caso de uso (Bot): Gestión de consejos.

Nombre del caso de uso	Gestión de consejos
Actor	Bot
Descripción	<p>El Bot se encargará de gestionar todos los consejos que se proporcionan durante la interacción con el usuario a través de la aplicación de mensajería instantánea de Facebook Messenger. Principalmente se encargará de recibir y almacenar los consejos que el usuario proporciona en la base de datos en tiempo real, de manera que el cambio en estos datos se vea instantáneamente reflejado en la base de datos y que todas las interacciones que se tengan con estos datos se actualicen en tiempo real. Por otro lado, el Bot es también el encargado de eliminar aquellos consejos que en base a las calificaciones resulten poco efectivos o contraproducentes para la comunidad de usuarios.</p>
Flujo básico	<ol style="list-style-type: none"> 1. Tras la solicitud del usuario el Bot proporciona a través de un mensaje de texto en la ventana de conversación las instrucciones para que el usuario pueda ingresar su consejo. Al mismo tiempo el Bot entra en estado de escucha para poder almacenar el consejo que se proporcione. 2. El Bot recibe satisfactoriamente el consejo, e inmediatamente lo ingresa a la Base de datos para que este pueda ser consultado en tiempo real. 3. Tras gestionar el consejo proporcionado, el Bot vuelve otra vez a su estado normal para

	que no se puedan ingresar más consejos sin que el usuario lo solicite.
Flujo alterno	<ul style="list-style-type: none"> El usuario solicita ingresar un consejo, pero ingresa un consejo en un formato no valido (archivo adjunto, enlace, etc.), por lo tanto, el Bot no lo ingresa a la base de datos y salta a otra interacción con el usuario, volviendo al mismo tiempo a su estado normal de no recepción de consejos
Pre-condición	<ul style="list-style-type: none"> El usuario debe haber interactuado previamente con el Bot y aceptado las interacciones y funciones mediante el botón “Empezar”.
Post-condición	<ul style="list-style-type: none"> Los consejos proporcionados se verán almacenados en la base de datos, para su inmediato manejo.

Tabla 5.11 Caso de uso (Bot): Obtención de datos personales.

Nombre del caso de uso	Obtención de datos personales
Actor	Bot
Descripción	El Bot se encargará de recolectar los datos personales básicos de los usuarios que interactúen con él a través de la aplicación de mensajería instantánea de Facebook Messenger. Principalmente se encargará de obtener datos básicos como el nombre del perfil de Facebook del usuario, su foto de perfil y su identificador único de usuario de manera automática mediante la implementación de ciertas funciones de Facebook. Estas acciones se realizan tras la previa aprobación del usuario al oprimir el botón “Empezar”. Estos datos serán almacenados en la base de datos y asociados a cada usuario de manera independiente.
Flujo básico	<ol style="list-style-type: none"> Tras la previa aceptación del usuario para las interacciones con el Bot mediante el botón “Empezar”, el Bot reconoce que se han aceptado las interacciones y procede a

	<p>recolectar los datos básicos del usuario de manera automática.</p> <p>2. El Bot registra al usuario con todos los datos recolectados dentro de la base de datos asociando al usuario a un Identificador Único.</p>
Flujo alternativo	<ul style="list-style-type: none"> El usuario no acepta las interacciones con el Bot ya que no presiona el botón “Empezar”, por lo tanto, el Bot no recolecta ninguna información personal del usuario y no lo registra a la base de datos.
Pre-condición	<ul style="list-style-type: none"> El usuario debe iniciar una conversación con la página de Facebook que funge como entidad del Bot y presionar el botón “Empezar”.
Post-condición	<ul style="list-style-type: none"> Los datos personales recolectados se verán almacenados en la base de datos, y se asociarán a cada usuario mediante un Identificador Único.

Tabla 5.12 Caso de uso (Bot): Interacción con el usuario.

Nombre del caso de uso	Interacción con el usuario
Actor	Bot
Descripción	El Bot se encargará de interactuar con el usuario a través de la aplicación de mensajería instantánea de Facebook de una forma poco invasiva y de manera intuitiva. Esto con la finalidad de que el Bot brinde un servicio eficiente al guiar al usuario a través de todas las características que este le puede proporcionar.
Flujo básico	<ol style="list-style-type: none"> El usuario inicia una conversación con la página de Facebook que representa la entidad del Bot. El Bot entra en acción para atender a todas las solicitudes y acciones que realice el usuario dentro de la ventana de conversación.

Flujo alternativo	No hay flujo alternativo.
Pre-condición	<ul style="list-style-type: none"> El usuario debe contar con un perfil propio de Facebook para poder comunicarse con el Bot a través del servicio de mensajería instantánea de Facebook Messenger.
Post-condición	No hay post-condición.

Tabla 5.13 Caso de uso (Bot): Vincular usuarios.

Nombre del caso de uso	Vincular usuarios
Actor	Bot
Descripción	Tras una previa interacción con el usuario en la que el usuario solicita el conocer a algún otro usuario de la aplicación, el Bot se encargará de proporcionar a ese usuario los datos necesarios de otro usuario de la aplicación para que pueda conocerlo; esto con la finalidad de fomentar el desarrollo de relaciones interpersonales que puedan ayudar a mejorar el estado de ánimo de la comunidad de usuarios.
Flujo básico	<ol style="list-style-type: none"> Tras la previa interacción correspondiente, el Bot recibe la solicitud por parte del usuario de conocer algún otro usuario de la aplicación, por lo tanto, el Bot ingresa a la base de datos y selecciona un usuario para vincularlo con el usuario que realiza la solicitud, de este modo el Bot proporciona los datos necesarios al usuario para que se ponga en contacto con el otro usuario.
Flujo alternativo	<ul style="list-style-type: none"> El usuario recibe los datos necesarios para ponerse en contacto con el otro usuario, sin embargo, por las condiciones de privacidad establecidas por el otro usuario dentro de Facebook no logra ponerse en contacto. En este caso se recomienda repetir el proceso para conocer a otro usuario.
Pre-condición	<ul style="list-style-type: none"> El usuario debe haber interactuado previamente con el Bot y aceptado las

	interacciones y funciones mediante el botón “Empezar”.
Post-condición	<ul style="list-style-type: none"> • Los usuarios de la aplicación interactuarán con otros usuarios de la comunidad.

Tabla 5.14 Caso de uso (Bot): Mostrar consejo.

Nombre del caso de uso	Mostrar consejo
Actor	Bot
Descripción	El Bot se encarga de desplegar en la ventana de conversación algún consejo que pueda ser de utilidad para el usuario que solicita consultar algún consejo.
Flujo básico	<ol style="list-style-type: none"> 1. Tras la previa interacción con el usuario a través de la aplicación de mensajería instantánea de Facebook Messenger, el usuario solicita la visualización de algún consejo. Entonces el Bot se dirige a la base de datos para obtener de manera aleatoria alguno de los consejos que se encuentran almacenados y que han sido proporcionados por la comunidad de usuarios. 2. El consejo es proporcionado por el Bot al usuario que lo solicita y lo muestra en la ventana de conversación a través de un mensaje de texto.
Flujo alternativo	No hay flujo alternativo
Pre-condición	<ul style="list-style-type: none"> • El usuario debe haber interactuado previamente con el Bot y aceptado las interacciones y funciones mediante el botón “Empezar”. • Deben existir consejos registrados en la base de datos para poder ser proporcionados de manera aleatoria. Como mínimo uno.
Post-condición	No hay post-condición.

	(Se recomienda que el usuario interprete de la manera que más le convenga el consejo que se le ha mostrado, con la finalidad de que este ayude a mejorar su estado de ánimo)
--	--

Tabla 5.15 Caso de uso (Bot): Proporcionar información de ayuda.

Nombre del caso de uso	Proporcionar información de ayuda
Actor	Bot
Descripción	El Bot se encarga de desplegar en la ventana de conversación datos de alguna entidad especializada en el control de la depresión y los trastornos psicológicos que puedan adjudicarse a esta situación, por ejemplo, el Centro de Servicios Psicológicos -Dr. Guillermo Dávila- de la Facultad de Psicología que proporciona servicio terapéutico en forma individual, grupal, de familia o pareja y diferentes talleres.
Flujo básico	<ol style="list-style-type: none"> 1. Tras la previa interacción con el usuario a través de la aplicación de mensajería instantánea de Facebook Messenger, el usuario solicita la obtención de información especializada para combatir la depresión. Entonces el Bot proporciona los datos de las asociaciones especializadas en estos casos, que se encuentren previamente definidos en la programación del Bot, a través de un mensaje de texto dentro de la ventana de conversación.
Flujo alternativo	No hay flujo alternativo
Pre-condición	<ul style="list-style-type: none"> • Los datos de las asociaciones especializadas deben estar previamente definidos en la programación del Bot.
Post-condición	No hay post-condición. (Se recomienda que el usuario utilice la información que se le proporciona de manera efectiva para resolver su situación de manera eficaz).

5.2.3 CASOS DE USO DE LA APLICACIÓN.

Al igual que en los casos anteriores en las siguientes tablas se definen de manera concreta cuales son las actividades involucradas para los casos de uso que tienen como actor principal en este caso a la aplicación móvil. Las tablas también definen el flujo básico, flujo alternativo, precondiciones y post condiciones.

Tabla 5.16 Caso de uso (App): Despliega información estadística en gráficas.

Nombre del caso de uso	Despliega información estadística en gráficas.
Actor	Aplicación Móvil
Descripción	La aplicación móvil será encargada de gestionar los datos generados por los múltiples usuarios para desplegar la información relevante de forma atractiva y entendible para el usuario, a través de la implementación de gráficos circulares y cálculo de índices. Cabe destacar que estos gráficos serán dinámicos y se actualizarán en tiempo real con respecto a los datos que se almacenen y actualicen en la base de datos, esto con la intención de que el usuario se capaz de visualizar las fluctuaciones en los datos y el comportamiento que presenta tanto un usuario de forma personal como la comunidad de usuarios como un ente global.
Flujo básico	<ol style="list-style-type: none"> 1. El usuario dentro de la aplicación móvil selecciona alguna de las opciones para visualizar estadísticas personales o globales. (También se incluyen las opciones de Top de usuarios y Top de consejos). 2. La aplicación móvil realiza consultas a la base de datos para obtener los datos precisos para generar las gráficas que se solicitan. De igual modo la aplicación será la encargada de realizar los cálculos para obtener índices y comparativas entre datos. 3. La aplicación despliega en pantalla las gráficas de manera atractiva que representan la información relevante para el usuario.

Flujo alternativo	<ul style="list-style-type: none"> El usuario solicita la consulta de alguna estadística, pero esta no se muestra por alguna inconsistencia en los datos, o por error de la solicitud del usuario. Entonces la aplicación informa sobre el error.
Pre-condición	<ul style="list-style-type: none"> Deben existir en la base de datos los datos necesarios, que se generaron a través de la interacción con el Bot, para poder generar las gráficas en base a esta información.
Post-condición	No hay post-condición. (Se recomienda que el usuario interprete la información que se le proporciona de la manera que más le convenga para mejorar su estado de ánimo)

Tabla 5.17 Caso de uso (App): Despliega información estadística en gráficas.

Nombre del caso de uso	Redirecciona al usuario hacia el Bot.
Actor	Aplicación Móvil
Descripción	La aplicación móvil se encargará de redireccionar a los usuarios hacia la interacción con el bot para que estos puedan generar sus datos personales.
Flujo básico	<ol style="list-style-type: none"> El usuario selecciona dentro de la aplicación móvil la opción que lo dirige a la conversación con el Bot. La aplicación se encargará de redireccionar al usuario a una conversación a través de Facebook con el Bot, en caso de que el usuario tenga instalada la aplicación oficial de Facebook Messenger se le preguntará al usuario si quiere usar esa aplicación o un navegador alternativo. (Se recomienda usar la aplicación oficial de Facebook Messenger para que la compatibilidad con todas las funciones del Bot se lleve a cabo de la manera más óptima).
Flujo alternativo	<ul style="list-style-type: none"> El usuario se redirecciona hacia la conversación con el Bot usando otro medio

	que no es la aplicación oficial de Facebook Messenger, como por ejemplo una interfaz proveedora de servicios web. Entonces no se garantiza la compatibilidad óptima con el Bot.
Pre-condición	<ul style="list-style-type: none">• El usuario debe contar con una conexión a internet para poder realizar el direccionamiento de forma exitosa.
Post-condición	<ul style="list-style-type: none">• Al ser redireccionado correctamente el usuario interactuará con el Bot para genera datos.

5.3 ELABORACIÓN.

Una vez que tenemos bien definidos los requerimientos funcionales y no funcionales que tendrá el sistema completo que involucra a la aplicación móvil y el Bot, así como también se han definido los casos de uso que tendrán los diferentes actores dentro del sistema, podemos proceder a la elaboración de la aplicación que tomará como base lo previamente descrito para enfocarse en la funcionalidad del sistema a base de analizar cómo es que todas las funciones y componentes se deben desarrollar de modo que en conjunto cumplan con los objetivos planteados de una manera óptima.

Dentro de este apartado se desarrollarán explícitamente cada uno de los componentes del sistema y cuáles son las herramientas necesarias para su desarrollo. Del mismo modo se explicará su funcionamiento dentro del sistema, haciendo énfasis en las relaciones que pueda tener con cada uno de los componentes que conforman el sistema general.

Más específicamente se describirá el proceso de elaboración de la base de datos en tiempo real, la programación y funcionamiento del Bot, así como el desarrollo de la aplicación móvil para Android. Estos componentes descritos en su conjunto conforman todo el sistema ya que su funcionamiento se encuentra condicionado por algunas de las funciones de cada uno de los componentes con los que se relaciona.

5.3.1 FLUJO LÓGICO DEL BOT.

A continuación, se describe de manera detallada la elaboración del Bot, así como de todas las funcionalidades que este requiere e implementa para su correcto desempeño dentro del sistema.

Antes de comenzar con el desarrollo y programación del Bot, se requiere definir un diagrama de procesos en el cual se especifique cual es el flujo lógico de acciones que realizará el Bot al momento de interactuar con el usuario. El siguiente diagrama representa el análisis sobre las actividades del Bot.

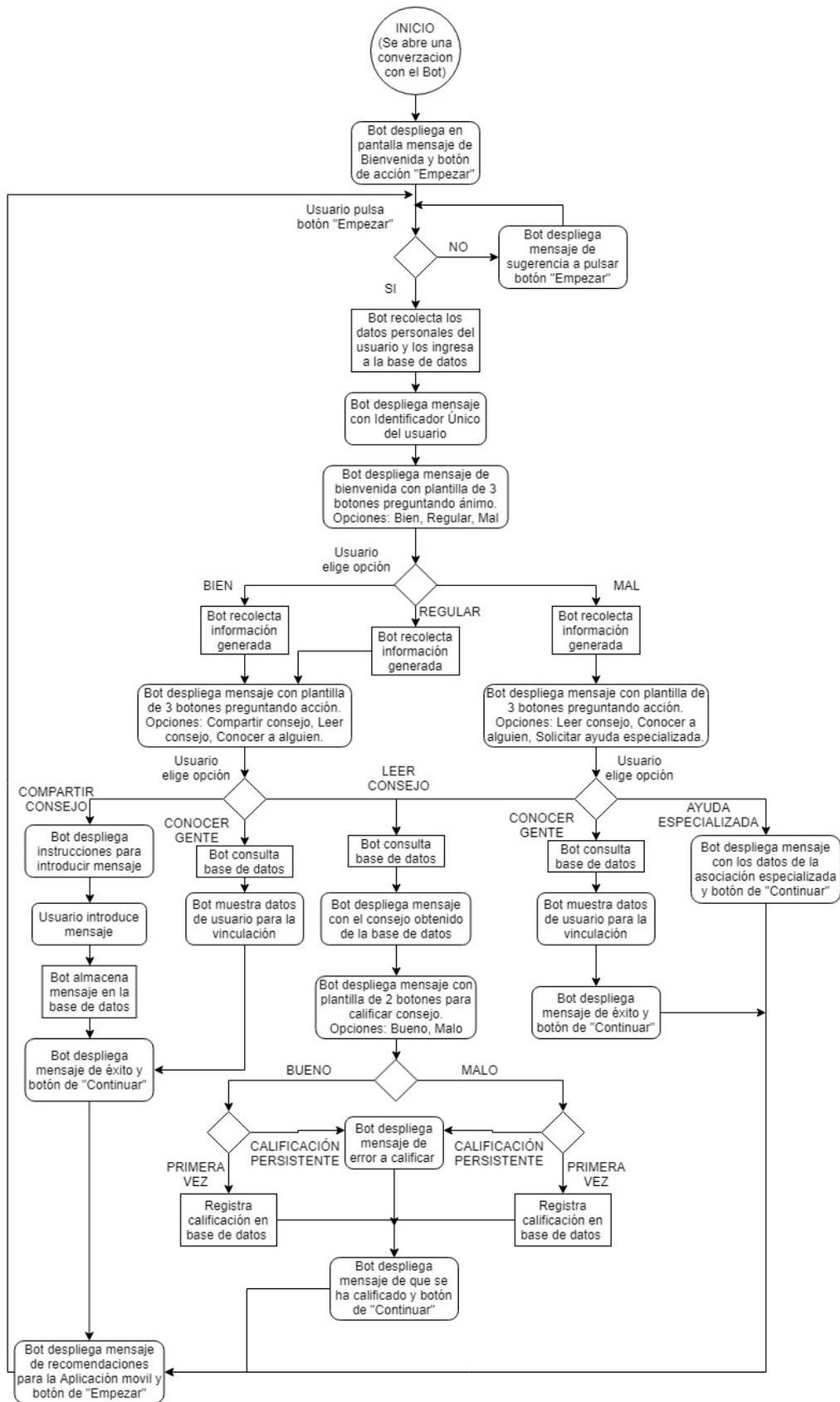


Figura 5.2 Diagrama de flujo para funcionamiento del Bot.

Como se puede apreciar en el diagrama de flujo previamente expuesto, las actividades que se desencadenan llevan una secuencia lógica que ayudará a a la fácil interacción del usuario con el Bot.

Dentro del diagrama de flujo se han tomado en consideración las acciones que son visibles para el usuario, como los mensajes que despliega o las plantillas con botones que permiten al usuario interactuar dentro del flujo lógico de la programación del Bot, así mismo se consideran de igual manera los procesos internos que debe realizar el Bot dentro de su programación como los accesos a base de datos para manipular información que se proporcione o se solicite.

También hay que destacar el hecho de que el Bot sigue un ciclo que es capaz de repetirse para la constante interacción con el usuario, sin embargo, el usuario se toma en consideración que el usuario puede desplazarse a lo largo de toda la ventana conversacional por lo cual podrá situarse en cualquier nodo del flujo lógico e iniciar la acción correspondiente con el Bot, esto permitirá que algunas de las funciones sean accedidas con mayor facilidad en caso de que así se lo requiera el usuario.

Una vez que se entienden por completo todos los procesos lógicos que seguirá el Bot para su funcionamiento e interacción con el Bot se puede proceder a su elaboración, a continuación, se detalla la estructuración del Bot y los componentes necesarios para su funcionamiento.

5.3.2 WEBHOOK.

El “webhook” se refiere a un método para la modificación en el funcionamiento de una aplicación que involucra la implementación de “callbacks” personalizados. Es por eso por lo que para el correcto funcionamiento del Bot se necesita la implementación de este método, ya que esto será la raíz del Bot donde se implementará el código de programación, así mismo es gracias a este método que se podrán gestionar los mensajes recibidos por parte del usuario y se podrán enviar tanto mensajes de respuesta como plantillas de interacción.

Para poder desarrollar la configuración del “webhook” es necesario contar con el entorno Node.js ya que este permite la implementación de la programación del Bot dentro de la capa de servidor, el código que se ejecutará en el servidor utilizará el lenguaje de JavaScript para su desarrollo.

Adicionalmente el alojamiento del código debe realizarse en un servidor HTTP público que debe contar con las siguientes características:

- Tener compatibilidad con el Protocolo seguro de transferencia de hipertexto (HTTPS).
- Poseer un certificado de Secure Sockets Layer (SSL) válido.
- Contar con un puerto abierto que acepte solicitudes GET y POST.

Para poder configurar el “webhook” de manera exitosa se utiliza Node.js para añadir las dependencias necesarias para la creación del “webhook”, las cuales se irán almacenando en un archivo ajeno de extensión “.JSON”.

El siguiente fragmento de código representa las dependencias necesarias para la configuración del “webhook”:

```
const
  express = require('express'),
  bodyParser = require('body-parser'),
  request = require('request'),
```

Una vez que se instalan correctamente las dependencias se crea el servidor HTTP que reciba las solicitudes a través de un puerto predeterminado. Para este caso específico y con la finalidad de realizar las pruebas se ha elegido el puerto 5000, el cual será el puerto a través del cual se podrá interactuar con el Bot utilizando la máquina de desarrollo como servidor. Para realizar las pruebas se utiliza “Ngrok”, el cual es un software que brinda la posibilidad de acceder al servidor local de la

máquina que lo ejecuta, este software genera una URL dinámica que cuenta con el protocolo HTTP y el certificado SSL, la cual se utilizará para comunicarse con los servicios de Facebook. Cabe destacar que esto solo se realiza con fines de prueba y dentro de la etapa de desarrollo ya que lo óptimo para el funcionamiento de la aplicación en producción es que los servicios del “webhook” se alojen en un servidor en la nube, para su alta disponibilidad.

Es necesario para la estructuración del “webhook” que este cuente con un extremo que acepte las solicitudes POST, verifique que los eventos recibidos sean compatibles para posteriormente poder analizar el mensaje recibido. Los mensajes que el “webhook” recibe son enviados directamente por la plataforma de Facebook en formato JSON.

En este caso particular el Bot necesita reconocer si el mensaje recibido es un texto plano o un evento “postback”, es por eso que dentro de la programación del “webhook” se realizará la gestión de los mensajes para poder realizar las acciones necesarias dependiendo del evento que se reciba.

La plataforma de Facebook asigna un identificador único a cada uno de sus usuarios al momento de realizar acciones de conversación con el Bot, este identificador único es conocido como PSID. Se explotará esta característica que ofrece Facebook para asociarlo como Identificador Único para la aplicación general, esto se puede realizar efectivamente ya que este identificador es exclusivo de la página de Facebook que se ha creado y no se puede utilizar fuera de la conversación que se ha creado, esto garantiza que solo los usuarios que han aceptado la interacción con el Bot puedan recibir mensajes de este.

El siguiente fragmento de código representa la configuración realizada para el proceso descrito anteriormente:

```
app = express().use(bodyParser.json()); // crea un servidor express http
// Selecciona el puerto del servidor y muestra un mensaje de exito
app.listen(process.env.PORT || 5000, () => console.log('webhook escuchando'));
// Crea el endpoint de nuestro webhook
app.post('/webhook', (req, res) => {
  let body = req.body;
  // Verifica si es un evento de una página suscrita
  if (body.object === 'page') {
    // Itera para cada entrada ya que puede haber multiples
    body.entry.forEach(function(entry) {
      // Obtiene el mensaje de entrada en forma de arreglo
      // como solamente se tiene un mensaje, se pone index 0
      let webhook_event = entry.messaging[0];
      console.log(webhook_event);
      // Obtiene el sender PSID
      let sender_psid = webhook_event.sender.id;
      // Verifica si el evento es un mensaje o postback
      // pasa el evento a la función necesaria para su manejo
      if (webhook_event.message) {
        handleMessage(sender_psid, webhook_event.message);
        console.log("EVENTO MENSAJE");
      } else if (webhook_event.postback) {
        handlePostback(sender_psid, webhook_event.postback);
      }
    });
    // Regresa un '200 OK' a todas las solicitudes
    res.status(200).send('EVENT_RECEIVED');
  } else {
    // Regresa un '404 Not Found' si el evento no es de una página suscrita
    res.sendStatus(404);
  }
});
```

Como se puede apreciar, este extremo del “webhook” itera sobre cada una de las entradas que se reciben, ya que pueden ser múltiples, de igual manera tras la gestión del mensaje recibido el extremo devuelve una respuesta 200, que informa a la plataforma de Facebook Messenger que el evento que envió se recibió de manera satisfactoria y por lo tanto ya no es necesario que vuelva a enviarlo.

Posteriormente es necesario configurar el soporte para solicitudes GET por parte de otro extremo del “webhook”, esto con la finalidad de que se pueda verificar el Bot con la plataforma oficial de Facebook Messenger ya que esta requiere de manera obligatoria que se garantice la compatibilidad con sus funciones y que el “webhook” es auténtico.

El siguiente código representa la configuración descrita para este extremo de verificación para el “webhook”.

```
// Añade soporte para solicitudes GET a nuestro webhook
app.get('/webhook', (req, res) => {
  // Token de verificación
  let VERIFY_TOKEN = "████████████████████";

  // Parametros de consulta
  let mode = req.query['hub.mode'];
  let token = req.query['hub.verify_token'];
  let challenge = req.query['hub.challenge'];

  // Verifica si el token y el modo estan en la cadena de consulta
  if (mode && token) {

    // Checa si el modo y el token enviados son correctos
    if (mode === 'subscribe' && token === VERIFY_TOKEN) {

      // Responde con el challenge del token
      console.log('WEBHOOK_VERIFICADO');
      res.status(200).send(challenge);

    } else {
      // Responde con '403 Forbidden' si el token no es correcto
      res.sendStatus(403);
    }
  }
});
```

Como se puede ver, el token de verificación es una cadena que nosotros especificamos (deberá ser una cadena secreta solo conocida por el programador) y que deberá ser idéntica a la que se registre dentro de la plataforma de Facebook Messenger, por otro lado, se consultan de la solicitud recibida por el “webhook” el modo y el token para compararlos, de este modo se garantiza que el “webhook” es auténtico y el correspondiente a nuestra página de Facebook. En caso de no lograrse la verificación la plataforma de Facebook Messenger no permitirá la interacción con el Bot hasta que se verifique el “webhook”.

Una vez que se ha verificado el correcto funcionamiento del “webhook” y la implementación de todas las funciones que este debe contener, se procede a suscribir el “webhook” a una aplicación de Facebook que se configurará a través de una cuenta de desarrollador de Facebook. La principal función de la aplicación de en Facebook es vincular la página de Facebook que funge como entidad del Bot con el “webhook” y la programación del Bot, adicionalmente a través de esta aplicación se podrán configurar diferentes parámetros, como lo son el tipo de evento que la plataforma enviará el “webhook”, generar los tokens de acceso y editar suscripción.

Como ya se mencionó previamente el sistema requiere del uso de una página de Facebook que servirá como entidad para el Bot, está la podemos crear directamente en el sitio oficial de Facebook mediante el uso de una cuenta de perfil de Facebook personal, la cual a su vez será asociada como administrador de la página. Cabe destacar que dentro de la fase de desarrollo solo los administradores pueden interactuar con el Bot, lo cual restringe su uso para usuarios ajenos que lo encuentren por casualidad.

La página de Facebook que se creó para usarse como entidad del Bot lleva el nombre de HappApp y se encuentra activa en Facebook dentro de la categoría de Educación. A continuación, se muestra una imagen de la visualización de la página dentro de la plataforma de Facebook.



Figura 5.3 Página de Facebook entidad del Bot.

Una vez que se cuenta con una cuenta de desarrollador de Facebook, podremos crear una aplicación dentro de la interfaz que se le proporciona a los

desarrolladores, esta aplicación será la que se vinculará con la página de Facebook que funge como entidad del Bot a través de la suscripción del “webhook”.

Para poder suscribir el “webhook” configurado a la aplicación que se crea dentro de la plataforma de Facebook para desarrolladores es necesario contar con la URL que apunte al servidor donde se encuentra alojado el “webhook”. Como se mencionó anteriormente, durante la fase de desarrollo este se utilizará “Ngrok” que será la encargada de proporcionarnos la URL necesaria para la suscripción del Bot.

La siguiente imagen representa el proceso de suscripción del “webhook” a la aplicación creada dentro de la plataforma oficial de Facebook para desarrolladores.

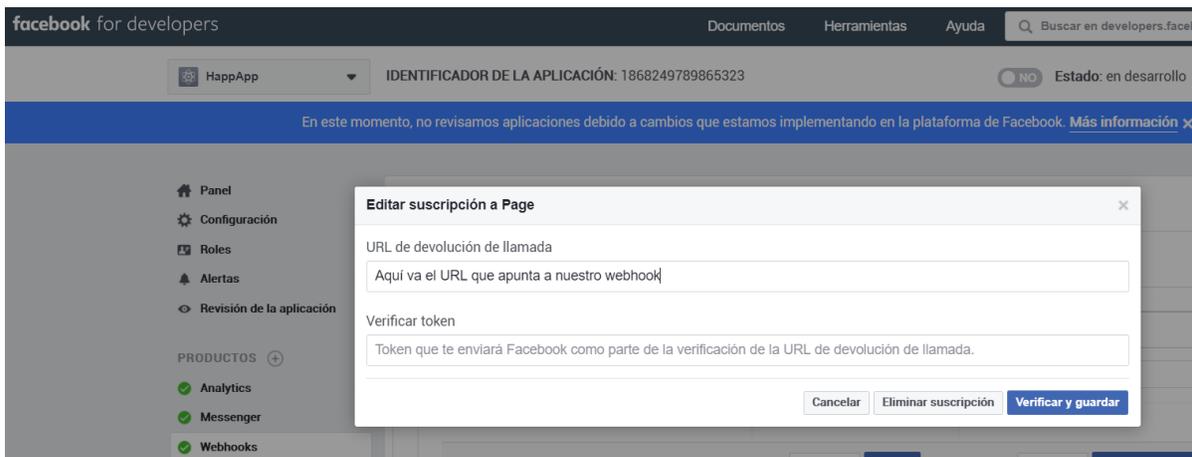


Figura 5.4 Suscripción del “webhook”.

Una vez que se ha suscrito el “webhook” de nuestro Bot satisfactoriamente a la aplicación que se creó dentro de la plataforma de Facebook para desarrolladores se procede a seleccionar cuales son los tipos de evento que se autoriza que se envíen al “webhook” por parte de la plataforma Messenger. Para el caso específico del Bot se seleccionan los eventos de “messages” y “messaging_postback”, ya que la gestión de eventos que realiza el Bot se enfoca principalmente en estos dos tipos de eventos.

Finalmente, para que podamos utilizar las funciones que nos proporcionan las API's de Messenger por medio del Bot, es necesario generar desde la plataforma de Facebook para desarrolladores un token de acceso a la página de nuestro Bot.

La siguiente imagen muestra la generación del token dentro de la plataforma de Facebook para desarrolladores.

Generación de token

Para empezar a usar las API es necesario el token de acceso de la página. Este token incluirá todos los permisos de Messenger aunque todavía no se haya aprobado su uso para tu aplicación. En este caso, solo podrás enviar mensajes a los administradores de la aplicación. También puedes generar tokens para páginas de las que no seas propietario mediante el inicio de sesión con Facebook.



Figura 5.5 Generación del token de acceso.

Con lo que se ha descrito anteriormente ya se tendrá configurado e manera exitosa todo lo necesario para la comunicación de la plataforma de Messenger con el Bot a través del “webhook”, los siguientes pasos a seguir consisten en la programación de todas las funciones que se describieron con anterioridad en el diagrama de flujo el Bot.

5.3.3 PROGRAMACIÓN DEL BOT.

Al igual que el “webhook”, la programación del Bot se ejecuta del lado del servidor, es por esta razón que el lenguaje seleccionado para su desarrollo es JavaScript a través del entorno que proporciona Node.js. Se utilizará una programación orientada a eventos para cumplir con el flujo lógico que se estableció previamente para el funcionamiento del Bot, así mismo dentro de la programación se detallarán componentes importantes como los mensajes que se desplegarán, así como las plantillas con botones que ayudarán a la interacción del usuario con el Bot.

Principalmente se tiene que comprender que, para el funcionamiento del Bot, el “webhook” es una pieza clave ya que es a través de este que se reciben los eventos y la información necesaria para identificar al usuario que se encuentra interactuando en el Bot y poder redirigir de este modo las respuestas generadas por el Bot hacia esa conversación. Para poder lograr este tipo de identificación, la plataforma proporciona a través del evento un identificador específico relacionado a la página del remitente que envía el mensaje que se conoce como PSID.

El siguiente fragmento de código muestra cómo se realiza la obtención del PSID para su posterior manejo.

```
// Obtiene el sender PSID
let sender_psid = webhook_event.sender.id;
```

Como se vio anteriormente es a través del “webhook” que se filtran los tipos de eventos que el Bot podrá manejar con su programación, para este caso específico el Bot solo trabajará con eventos de tipo “messages” y “messaging_postback”, los cuales se redireccionarán para su manejo a las funciones implementadas dentro de la programación del Bot.

El siguiente fragmento representa el filtrado de los eventos con los que podrá trabajar el Bot, y el llamado a las funciones que se encargarán de trabajar con esta información.

```
// Verifica si el evento es un mensaje o postback
// pasa el evento a la función necesaria para su manejo
if (webhook_event.message) {
  handleMessage(sender_psid, webhook_event.message);
  console.log("EVENTO MENSAJE");
} else if (webhook_event.postback) {
  handlePostback(sender_psid, webhook_event.postback);
}
```

Como se puede apreciar, durante el filtrado se hacen las llamadas a las funciones que serán las encargadas de procesar los datos proporcionados dependiendo del tipo de evento que se reciba, estas funciones recibirán como parámetros tanto el

tipo de evento que se está manejando como el PSID que servirá para identificar al usuario con el que se está interactuando.

Una vez que se filtra el tipo del evento que se ha recibido por parte de la plataforma Messenger, son las funciones que se definen dentro del Bot las encargadas de gestionar la información y realizar las actividades programadas para cumplir con el objetivo general del sistema. Se tienen definidas dos funciones principales, una llamada "handleMessage" encargada de manejar la información de los eventos de tipo mensaje y otra llamada "handlePostBack" que se encarga de manejar la información de los eventos de tipo "postback". Estas funciones reciben un parámetro en común el cual es el PSID para identificar al usuario con el que se mantiene la interacción.

Independientemente de la función que se haya seleccionado para manejar la información del evento, se necesita una función para poder responder al usuario con el que se esté interactuando a través de la ventana de conversación. Es por esto que se hace uso de la API de envío proporcionada por Facebook Messenger, para poder implementar esta herramienta se crea la función "callSendAPI" que recibirá como parámetros el PSID, y una respuesta válida, dentro de la función se realiza una solicitud que se dirige a la dirección de la API en la nube, estas solicitudes poseen dos propiedades básicas

- "recipient": Aquí se define el destinatario al cual se dirige la respuesta generada por el Bot, para este caso específico se utiliza el PSID para identificar al destinatario.
- "message": Aquí se define la estructura de la respuesta que será enviada como mensaje al destinatario que en este caso es el usuario con el cual se mantiene la interacción.

El siguiente fragmento de código representa la estructuración de la función “callSendAPI” utilizada para enviar las respuestas generadas por el Bot al usuario con el que se interactúa.

```
// Funcion que manda las respuestas a traves de la API
function callSendAPI(sender_psid, response) {
  // Estructura del cuerpo de la respuesta
  let request_body = {
    "recipient": {
      "id": sender_psid
    },
    "message": response
  }

  // Envio de la solicitud HTTP a la plataforma para la API
  request({
    "uri": "https://graph.facebook.com/v2.6/me/messages",
    "qs": { "access_token": PAGE_ACCESS_TOKEN },
    "method": "POST",
    "json": request_body
  }, (err, res, body) => {
    if (!err) {
      console.log('mensaje enviado!');
    } else {
      console.error("No se puede enviar el mensaje:" + err);
    }
  });
}
```

Como se puede apreciar, esta función define la estructura que tendrá el cuerpo de la respuesta que se enviará a la plataforma para su posterior envío al usuario. También se detalla la solicitud que se enviará a la plataforma de Facebook Messenger como un archivo de tipo JSON, la cual se encarga de desplegar la respuesta en la ventana de conversación correspondiente con el formato establecido por el Bot.

Cuando un usuario envía un mensaje a la conversación con el Bot a través de la plataforma de mensajería instantánea se produce un evento de tipo mensaje que será enviado a la dirección con la que se ha registrado el “webhook”, es entonces que este filtra el evento y lo pasa a la programación del Bot. Al ser un evento de tipo mensaje se llama a la función “handleMessage” para manejar la información proporcionada y contestar de manera adecuada al usuario. Esta función realiza las acciones necesarias para cumplir con el flujo lógico que se detalló anteriormente y hace uso de la función “callSendAPI” para enviar la respuesta que corresponda según el caso.

El siguiente fragmento de código representa la programación de la función “handleMessage” que maneja la información de los eventos de tipo mensaje para generar una respuesta según se requiera.

```
// Funcion que maneja eventos de tipo mensaje
var handleMessage = function(sender_psid, received_message) {

    let response;

    //verifica si el mensaje recibido fue escrito por el usuario
    if (received_message.is_echo){

    } else if (received_message.text && ConsFlag == 1) {
        InsertaConsejos(sender_psid, received_message.text);
        // Se crea la respuesta al evento
        response = ConsejoEscrito;
        ConsFlag = 0;
    } else {

        // Se crea la respuesta al evento
        response = EventoMensaje;

    }

    callSendAPI(sender_psid, response);

}
```

Como se puede apreciar la función recibe dos parámetros los cuales son el PSID y el evento de mensaje, como se definió anteriormente esta función se utiliza para responder a los mensajes que el usuario pueda mandar como texto, del mismo modo es la encargada de identificar cuando el usuario va a introducir un consejo conforme a lo que se ha establecido anteriormente.

Por otro lado, cuando se registra un evento de “postback” por parte del usuario a través de la plataforma de mensajería instantánea se hace la llamada a la función “handlePostBack” la cual es la encargada de gestionar estos eventos de tipo “postback” para producir la respuesta adecuada para el usuario conforme a lo establecido en el flujo lógico del Bot. Cabe mencionar que al igual que la función que se utiliza para manejar los eventos de mensaje, esta función hará uso de la función de apoyo “callSendAPI” para poder enviar la respuesta al usuario con el que se está interactuando a través de la plataforma de mensajería instantánea de Facebook Messenger.

El siguiente fragmento de código representa la programación de la función “handlePostBack” que es utilizada por el Bot para responder a las interacciones con el usuario, el Bot principalmente concentra sus funcionalidades a este tipo de

eventos ya que el permiten saber de manera exacta qué es lo que el usuario ha seleccionado para la interacción y en base a esto poder generar la respuesta adecuada.

```
var handlePostback = function(sender_psid, received_postback) {
  let response;

  // se obtiene el payload del evento postback
  let payload = received_postback.payload;

  // se elige la respuesta en base al postback recibido
  if (payload === 'BotonInicio') {

    callSendAPI(sender_psid, ReturnUID(sender_psid));
    response = BotonInicio;
    getUserDetails(sender_psid);

  } else if (payload === 'Fine') {

    IndiceFeliz(sender_psid);
    PromedioGlobal(1);
    response = Fine;

  }
}
```

Como se puede apreciar la función requiere de dos parámetros, tanto del PSID para poder identificar cuál es el usuario con el que se está interactuando, como el “postback” que se ha recibido, una parte muy importante de la función es la obtención de la propiedad “payload” del evento recibido ya que esto nos ayudará a poder definir la respuesta adecuada que proporcionará el Bot en base a la lógica que tiene programada. El fragmente de código que se ha expuesto anteriormente representa solo la parte principal de la función, ya que en la parte intermedia se programa la lógica que tendrá el Bot para responder a todas las solicitudes, eventos e interacciones del usuario por medio del análisis del evento “postback”.

A continuación, se muestra el código de la parte final de la función en la que se hace llamada a la función auxiliar “callSendAPI” para enviar la respuesta que se ha seleccionado a usuario con el que se interactúa a través de la ventana de conversación de la plataforma Facebook Messenger.

```
    } else if (payload === 'ConocerGente') {  
        ObtenerUsuarios(sender_psid);  
        response = ExitoContinuar;  
    } else if (payload === 'RecibirAyuda') {  
        response = Ayuda;  
        ExitoBucle(sender_psid, ExitoContinuar);  
    } else if (payload === 'RecomiendaApp') {  
        response = Recomendacion;  
    }  
  
    // llama a la funcion para enviar la respuesta  
    callSendAPI(sender_psid, response);  
}
```

Se han definido todos los componentes funcionales del Bot que permiten la correcta interacción con el usuario, que van desde la configuración del “webhook”, pasando por las suscripciones de acceso a la plataforma oficial de Facebook Messenger hasta las funciones auxiliares y API’s que se utilizan para manejar las interacciones del usuario con el Bot.

La programación lógica se realiza en base a condicionantes que permiten el análisis de las interacciones para responder con un comportamiento específico.

A continuación, se anexa un fragmento del código donde se realiza esta programación lógica.

```

if (payload === 'BotonInicio') {

    callSendAPI(sender_psid, ReturnUID(sender_psid));
    response = BotonInicio;
    getUserDetails(sender_psid);

} else if (payload === 'Fine') {

    IndiceFeliz(sender_psid);
    PromedioGlobal(1);
    response = Fine;

} else if (payload === 'Medium') {

    IndiceNormal(sender_psid);
    response = Medium;

} else if (payload === 'Bad') {

    IndiceMal(sender_psid);
    PromedioGlobal(0);
    response = Bad;

} else if (payload === 'CompartirConsejo') {

    ConsFlag = 1;
    response = EscribirConsejo;

} else if (payload === 'LeerConsejo') {

    ObtenerConsejos(sender_psid);
    console.log(CaliText);

} else if (payload === 'Bueno') {

    CalificacionBuena(CaliText, sender_psid);
    CaliText = null;
    response = RecienCalificado;

} else if (payload === 'Malo') {

    CalificacionMala(CaliText, sender_psid);
    CaliText = null;
    response = RecienCalificado;

} else if (payload === 'ConocerGente') {

    ObtenerUsuarios(sender_psid);
    response = ExitoContinuar;

} else if (payload === 'RecibirAyuda') {

    response = Ayuda;
    ExitoBucle(sender_psid, ExitoContinuar);

} else if (payload === 'RecomiendaApp') {

    response = Recomendacion;

}

```

Como se puede apreciar estas condicionantes se activan en base al análisis de la propiedad “payload” proporcionada por un evento de “postback”. Dentro de cada una de las acciones que realiza el Bot se incluyen los mensajes con los cuales responderá al usuario y las funciones que ejecutará para llevar a cabo tareas específicas como puede ser la recolección automatizada de datos o los accesos a la base de datos para consulta o inserción.

5.3.4 MENSAJES DEL BOT.

Al abrir una conversación con el Bot este dará un mensaje de bienvenida antes de que se realice cualquier interacción dentro de la ventana de conversación, para poder lograr esto se realizan solicitudes "Post" a la API de la plataforma oficial de Facebook Messenger que contendrán la estructura del mensaje de bienvenida que será desplegado, adicionalmente a ese mensaje se realizará la activación del botón "Empezar" que describe una propiedad "payload" en base a la interacción con este. Estas solicitudes se realizan mediante GitBash que en su entorno permite solicitudes de este tipo que finalmente quedarán registradas en la plataforma oficial de Facebook y se asociarán a la página que funge como entidad del Bot. El siguiente fragmente de código demuestra la implementación de las solicitudes anteriormente descritas.

```
curl -X POST -H "Content-Type: application/json" -d '{
  "get_started": {"payload": "BotonInicio"}
}' "https://graph.facebook.com/v2.6/me/messenger_profile?
access_token=EAAajKc0TnWsBALhhczl0R0o05XLeZCmcz5gbeWIyB4Iars
BPamZBZBjKh1LCnnrAZCIyZg4Glc5XF9twC4JAholQI"

curl -X POST -H "Content-Type: application/json" -d '{
  "greeting": [
    {
      "locale": "default",
      "text": "Hola {{user_first_name}} HappApp te ayudara! R
CLICK A EMPEZAR"
    }
  ]
}' "https://graph.facebook.com/v2.6/me/messenger_profile?
access_token=EAAajKc0TnWsBALhhczl0R0o05XLeZCmcz5gbeWIyB4Iars
BPamZBZBjKh1LCnnrAZCIyZg4Glc5XF9twC4JAholQI"
```

Los múltiples mensajes con los que puede responder el Bot a una interacción con el usuario son definidos mediante un formato estructurado tipo JSON, dentro de este caso específico se utilizan tanto mensajes planos como plantillas genéricas que se adaptan a la estructura establecida por la plataforma de Facebook Messenger con la finalidad de poder desplegar botones, imágenes y texto dentro de un solo mensaje de respuesta para el usuario.

A continuación, se muestran algunos de los mensajes estructurados que utiliza el Bot para responder a las interacciones con el usuario.

```
const EventoMensaje = {
  "attachment": {
    "type": "template",
    "payload": {
      "template_type": "button",
      "text": "Lo siento, para poder ayudarte tienes que dar click en EMPEZAR",
      "buttons": [
        {
          "type": "postback",
          "title": "EMPEZAR",
          "payload": "BotonInicio",
        }
      ]
    }
  }
};

const RecienCalificado = {
  "attachment": {
    "type": "template",
    "payload": {
      "template_type": "button",
      "text": "Gracias por calificar, esto ayuda a más personas :D",
      "buttons": [
        {
          "type": "postback",
          "title": "CONTINUAR",
          "payload": "RecomiendaApp",
        }
      ]
    }
  }
};
```

El código anterior representa la estructuración de mensajes que implementan una plantilla de respuesta para el usuario. Estas plantillas se definen en la propiedad “attachment” del mensaje que será enviado como respuesta a través de la API de la plataforma, esta además contiene dentro de su estructura las propiedades “type” y “payload”, que como se explicó anteriormente servirán para el análisis de las interacciones con el usuario. Estos ejemplos en particular representan la estructuración de una plantilla que involucran dentro del mensaje elementos de texto y botones. Para poder utilizar estas estructuras dentro de las funciones del Bot, estas son declaradas como constantes.

Las plantillas se encuentran limitadas al uso de tres botones dentro de su estructura ya que esto así lo delimita la plataforma de Facebook Messenger, a continuación, se muestra un ejemplo de mensaje que utiliza los tres botones.

```
const Medium = {
  "attachment": {
    "type": "template",
    "payload": {
      "template_type": "button",
      "text": "Vamos por buen camino! :D Elige una opcion =)",
      "buttons": [
        {
          "type": "postback",
          "title": "Compartir algún consejo para ayudar",
          "payload": "CompartirConsejo",
        },
        {
          "type": "postback",
          "title": "Leer algun consejo",
          "payload": "LeerConsejo",
        },
        {
          "type": "postback",
          "title": "Conocer a alguien",
          "payload": "ConocerGente",
        }
      ]
    }
  }
};
```

Cando la respuesta que dará el Bot no necesita de la implementación de botones dentro de su estructura no es necesario realizar la estructuración completa e una plantilla, simplemente se defina la propiedad "text" como se demuestra en el siguiente fragmento de código.

```
const EscribirConsejo = {
  "text": "A CONTINUACION ESCRIBE TU CONSEJO EN UN SOLO MENSAJE:"
};
```

Existe un caso en el que el Bot proporciona los datos de un usuario alterno al usuario con el que se encuentra interactuando, esto con el fin de vincular a estos usuarios, dentro de esa clase de mensaje también se involucra la imagen del perfil del usuario, es por eso que la plantilla utilizada involucra adicionalmente el parámetro "image_url" que sirve para desplegar una imagen dentro del mensaje por medio de un url que se proporciona. El siguiente fragmento de código demuestra la estructuración de este tipo de mensaje.

```
const UsuarioCon = {
  "attachment": {
    "type": "template",
    "payload": {
      "template_type": "generic",
      "elements": [
        {
          "title": "CONOCE A ESTA PERSONA",
          "image_url": Foto,
          "subtitle": "Buscala por su nombre en Facebook: "+Persona,
        }
      ]
    }
  }
};
```

Cabe destacar que por la extensión de las estructuras que presentan los mensajes utilizados por el Bot para responder al usuario con el que interactúa a través de la plataforma de Facebook Messenger, estas no serán expuestas en su totalidad, además de que se encuentran distribuidas a lo largo de toda la programación del Bot según su requerimiento.

A continuación, se anexa para mayor detalle el código de la estructura de los mensajes utilizados por el Bot como respuestas a las interacciones con el usuario.

```
const EventoMensaje ={
  "attachment": {
    "type": "template",
    "payload":{
      "template_type":"button",
      "text":"Lo siento, para poder ayudarte tienes que dar click en EMPEZAR",
      "buttons":[
        {
          "type": "postback",
          "title": "EMPEZAR",
          "payload": "BotonInicio",
        }
      ]
    }
  }
};

const RecienCalificado ={
  "attachment": {
    "type": "template",
    "payload":{
      "template_type":"button",
      "text":"Gracias por calificar, esto ayuda a más personas :D",
      "buttons":[
        {
          "type": "postback",
          "title": "CONTINUAR",
          "payload": "RecomiendaApp",
        }
      ]
    }
  }
};

const Recomendacion ={
  "attachment": {
    "type": "template",
    "payload":{
      "template_type":"button",
      "text":"Haz aportado mucho. Te recomendamos consultar datos relevantes en la apl
      tu Identificador Único que se te proporcionó al principio, si no lo recuerdas so
      "buttons":[
        {
          "type": "postback",
          "title": "EMPEZAR",
          "payload": "BotonInicio",
        }
      ]
    }
  }
};
```

```

const BotonInicio = {
  "attachment": {
    "type": "template",
    "payload":{
      "template_type":"button",
      "text":"Bienvenido a HappApp! :) ¿Cómo te sientes el día de hoy?",
      "buttons":[
        {
          "type": "postback",
          "title": "Muy bien! :D",
          "payload": "Fine",
        },
        {
          "type": "postback",
          "title": "Regular ",
          "payload": "Medium",
        },
        {
          "type": "postback",
          "title": "Mal :((",
          "payload": "Bad",
        }
      ]
    }
  }
};

const Fine = {
  "attachment": {
    "type": "template",
    "payload":{
      "template_type":"button",
      "text":"Me alegra que te sientas bien =) Dime que te gustaría hacer",
      "buttons":[
        {
          "type": "postback",
          "title": "Compartir algún consejo para ayudar",
          "payload": "CompartirConsejo",
        },
        {
          "type": "postback",
          "title": "Leer algun consejo",
          "payload": "LeerConsejo",
        },
        {
          "type": "postback",
          "title": "Conocer a alguien",
          "payload": "ConocerGente",
        }
      ]
    }
  }
};

```

```

const Medium = {
  "attachment": {
    "type": "template",
    "payload": {
      "template_type": "button",
      "text": "Vamos por buen camino! :D Elige una opcion =)",
      "buttons": [
        {
          "type": "postback",
          "title": "Compartir algún consejo para ayudar",
          "payload": "CompartirConsejo",
        },
        {
          "type": "postback",
          "title": "Leer algun consejo",
          "payload": "LeerConsejo",
        },
        {
          "type": "postback",
          "title": "Conocer a alguien",
          "payload": "ConocerGente",
        }
      ]
    }
  }
};

const Bad = {
  "attachment": {
    "type": "template",
    "payload": {
      "template_type": "button",
      "text": "No te preocupes! Lo podemos solucionar! :D elige una opcion",
      "buttons": [
        {
          "type": "postback",
          "title": "Leer algun consejo",
          "payload": "LeerConsejo",
        },
        {
          "type": "postback",
          "title": "Conocer a alguien",
          "payload": "ConocerGente",
        },
        {
          "type": "postback",
          "title": "Recibir ayuda especializada",
          "payload": "RecibirAyuda",
        }
      ]
    }
  }
};

```

```

const EscribirConsejo ={
  "text": "A CONTINUACION ESCRIBE TU CONSEJO EN UN SOLO MENSAJE:"
};

const Ayuda ={
  "text": "El Centro de Servicios Psicológicos -Dr. Guillermo Dávila-
  familia o pareja y diferentes talleres. Además, cuenta con programas
  Psicóloga Rebeca Sánchez. Edif. D , sótano, de la Facultad de Psicol
  Teléfono 5622-2309."
};

const ConsejoEscrito ={
  "attachment": {
    "type": "template",
    "payload":{
      "template_type":"button",|
      "text":"Recibimos tu consejo, gracias por aportar tu ayuda",
      "buttons":[
        {
          "type": "postback",
          "title": "CONTINUAR",
          "payload": "RecomiendaApp",
        }
      ]
    }
  }
};

const ExitoContinuar ={
  "attachment": {
    "type": "template",
    "payload":{
      "template_type":"button",
      "text":"Espero que estés contento con el servicio! :D, presiona co
      "buttons":[
        {
          "type": "postback",
          "title": "CONTINUAR",
          "payload": "RecomiendaApp",
        }
      ]
    }
  }
};

```

Estos mensajes al ser declarados constantes pueden ser utilizados por las funciones mediante la llamada a los nombres que identifican a cada una.

5.3.5 FUNCIONES ESPECIALES DEL BOT.

Dentro del diagrama de flujo lógico que se ha establecido para el Bot se hace referencia a diferentes actividades que tendrá que realizar fuera de la propia interfaz de la ventana de conversación proporcionada por la plataforma de Facebook Messenger, estas funciones principalmente se enfocan en el manejo de los datos generados u obtenidos a través del usuario que se almacenarán en la base de datos incluyendo así también las consultas que el Bot debe realiza a la base para mostrar ciertos mensajes específicos.

Para poder acceder a la base de datos que se utiliza en este caso específico la cual es Firebase Database es necesario instalar las dependencias necesarias al proyecto a través del entorno Node.js; tras realizar las configuraciones necesarias para el uso de las características de Firebase dentro de la aplicación se procede a programar las funciones especiales que serán usadas para el manejo de la información.

Como se detalló anteriormente, cuando el usuario realiza la primera interacción con el Bot a través del botón “Empezar” el Bot recolectará sus datos de manera automatizada haciendo uso de las propiedades de la plataforma de Facebook, para poder lograr esto se implementan tres funciones especiales que se encargarán de recolectar los datos y posteriormente ingresarlos a la base de datos de manera inmediata. El siguiente fragmento de código representa la programación de esas funciones.

```

const getUserDetails = (senderId) => {
  return new Promise((resolve, reject) => {
    getDetailsFromFacebook(senderId, (err, {statusCode}, body) => {
      if (err) {
        return reject(err);
      } else if (statusCode !== 200) {
        return reject({
          statusCode,
          message: 'Unable to fetch user data for user',
          senderId,
        });
      }
    });

    var usernamefull = body.first_name+" "+body.last_name;
    var FotoPerfil = body.profile_pic;

    writeFirstUserData(senderId, usernamefull, FotoPerfil);

    return resolve({
      name: body.first_name || body.last_name,
      profilePic: body.profile_pic,
      fbId: senderId,
    });
  });
});

const getDetailsFromFacebook = (userId, callback) => {
  request(
    {
      "method": "GET",
      "url": "https://graph.facebook.com/v2.6/"+userId+"?fields=first_name,last_name,profile_pic&access_token="
      "json": true,
      "qs": {
        "access_token": PAGE_ACCESS_TOKEN,
        "fields": 'first_name,last_name,profile_pic',
      },
    },
    callback
  );
};

```

Las dos funciones anteriores se utilizan para recolectar los datos del perfil de Facebook del usuario con el que se está interactuando, la primera función depende de la segunda ya que es en esta donde por medio de una solicitud Get hacia la API de Facebook se logra el acceso a la información básica del usuario comprobando mediante una autenticación que se está autorizado para realizar dicha acción. Por otro lado, una vez obtenidos los datos, la primera función se encarga de gestionar la información obtenida para extraer los datos que realmente interesan para los fines de la aplicación que en este caso son el primer nombre y apellido del usuario, así como su fotografía de perfil.

Posteriormente se utiliza una función que se encarga de ingresar los datos obtenidos por las funciones previas a la base de datos. El siguiente fragmento de código detalla la función descrita.

```
function writeFirstUserData(sender_psid, name, picture) {
  var ref = HappBase.ref('Usuarios/'+ sender_psid);
  var UID = sender_psid;
  var nombre = name;
  var foto = picture;

  //Inserta a Firebase los datos principales del usuario con el que interactua el bot
  ref.update({
    UsuarioID: UID,
    PerfilFB: nombre,
    FotoPerfil: foto
  });
}
```

Como se puede ver las escrituras a la base de datos se realizan mediante funciones que se obtienen a través de la instalación de la dependencia requerida por Firebase.

Otra de las actividades que debe realizar el Bot conforme a lo descrito en el flujo lógico es la gestión de datos que ayuden a generar estadísticas que serán representadas posteriormente de forma gráfica por la aplicación móvil. Para esto se elaboraron diferentes funciones que tienen como propósito generar datos numéricos que se registrarán a la base de datos, estos datos representan la información sobre el estado de ánimo de cada usuario. El siguiente código representa la programación de estas funciones.

```

function IndiceFeliz(sender_psid) {
  var ref = HappBase.ref('Usuarios/'+ sender_psid);
  //Incrementa el indice en 1
  ref.once('value').then(function(snapshot) {
    var Incrementa = (snapshot.val() && snapshot.val().PuntosAnimo) || 0;
    var Feliz = (snapshot.val() && snapshot.val().MomentosBien) || 0;
    var updates = {};
    updates['/PuntosAnimo/'] = Incrementa+1;
    updates['/AnimoActual/'] = 1;
    updates['/MomentosBien/'] = Feliz+1;

    ref.update(updates);
  });
}

function IndiceNormal(sender_psid) {
  var ref = HappBase.ref('Usuarios/'+ sender_psid);
  //Coloca el animo actual en 0
  ref.once('value').then(function(snapshot) {
    var updates = {};
    updates['/AnimoActual/'] = 0;

    ref.update(updates);
  });
}

function IndiceMal(sender_psid) {
  var ref = HappBase.ref('Usuarios/'+ sender_psid);
  //Decremeta el indice en 1
  ref.once('value').then(function(snapshot) {
    var Incrementa = (snapshot.val() && snapshot.val().PuntosAnimo) || 0;
    var Triste = (snapshot.val() && snapshot.val().MomentosMal) || 0;
    var updates = {};
    updates['/PuntosAnimo/'] = Incrementa-1;
    updates['/AnimoActual/'] = -1;
    updates['/MomentosMal/'] = Triste+1;

    ref.update(updates);
  });
}

```

Las funciones que se expusieron anteriormente representan el estado de ánimo del usuario con el que se interactúa, en base a las interacciones que se mantienen con

el usuario se determina cuál es la función que será llamada, sin embargo, estas funciones comparten la característica de que acceden a la base de datos para actualizar los datos que en esta se encuentran o crearlos si así se requiere. Posteriormente se detallará cual es la función de cada uno de estos datos.

Otro requerimiento de la aplicación involucra los datos generados de manera global por la comunidad de usuarios, es por eso que para gestionar la información general producida se hace uso de una función especial, que actualizará los datos de una entidad global para su posterior uso por la aplicación. El siguiente fragmento muestra la función descrita.

```
function PromedioGlobal(dato) {
  var ref = HappBase.ref('PromedioGlobal/');
  //Incrementa o decremmta el indice en 1
  ref.once('value').then(function(snapshot) {
    var Mal = (snapshot.val() && snapshot.val().AnimoMal) || 0;
    var Bien = (snapshot.val() && snapshot.val().AnimoBien) || 0;
    var updates = {};
    if(dato==0){
      updates['/AnimoMal/'] = Mal+1;
    }else if (dato==1){
      updates['/AnimoBien/'] = Bien+1;
    }

    ref.update(updates);

  });
}
```

Esta función hace referencia a una parte de la base de datos que contendrá los datos globales generados por la comunidad de usuarios, a través de esta función cuando es llamada se actualizan los datos numéricos especificados para después poder representar estadísticas de modo gráfico.

El Bot tendrá adicionalmente la característica de poder gestionar consejos proporcionados por los usuarios y los consultados en la base de datos para ser mostrados a los usuarios, todo esto para cumplir con lo establecido dentro del flujo lógico. Con la finalidad de conseguir esto se emplean las siguientes funciones especiales que permiten acceder a la base de datos para consultar los consejos y también recolectar el consejo escrito por el usuario para su posterior inserción a la base de datos.

```

function InsertaConsejos(senderID,consejo) {
  var ref = HappBase.ref('Consejos/');
  ConsNum =ConsNum+1;
  // obtiene una llave para la nueva entrada.
  var tip = consejo;
  var newPostKey = ref.push().key;

  // escribe el consejo y la caluficación en 0.
  var updates = {};
  updates[newPostKey+'/Texto/'] = tip;
  updates[newPostKey+'/Calificacion/'] = 0;

  if (HappBase.ref().child('Usuarios').equalTo(senderID)){
    ref.update(updates);
  }
}

```

La función mostrada anteriormente recibe como parámetro el consejo ingresado por el usuario, para posteriormente ingresarlo a la base de datos con una calificación igual a cero.

Otra de las funciones para trabajar con los consejos es la que permite obtener los consejos que ya se han ingresado a la base de datos para poder ser mostrados al usuario. A continuación, se muestra la función encargada de realizar esta acción.

```

function ObtenerConsejos(sender_psid) {

    var ref = HappBase.ref('Consejos/');

    ref.once("value", function(data) {

        var keys1 = data.val();
        var keys = Object.keys(keys1);
        var indi = randomIntFromInterval(0, keys.length-1);

        console.log(keys);
        console.log(keys.length);
        console.log(keys[indi]);

        var CalificaCons = keys[indi];
        obtenCali(CalificaCons);

        var refi = HappBase.ref('Consejos/'+CalificaCons);

        refi.once("value", function(data) {

            var Texto = data.val().Texto;

            const Consejote ={
                "text": Texto
            };

            const CalificaConsejo ={
                "attachment": {
                    "type": "template",
                    "payload":{
                        "template_type":"button",
                        "text":"Porfavor dinos ¿Cómo te resultó este consejo?",
                        "buttons":[
                            {
                                "type": "postback",
                                "title": "BUENO",
                                "payload": "Bueno",
                            },
                            {
                                "type": "postback",
                                "title": "MALO",
                                "payload": "Mal",
                            }
                        ]
                    }
                }
            };

            console.log(Texto);

            callSendAPI(sender_psid, Consejote);
            callSendAPI(sender_psid, CalificaConsejo);
        });
    });
}

```

La función anterior cuando es llamada realiza una consulta a la base de datos para a través de las llaves que identifican a cada consejo poder proporcionar un consejo de todos los registrados de forma aleatoria. De igual modo se encarga de desplegar en pantalla un mensaje de interacción para el usuario con dos botones para dar la opción de calificar el mensaje que se haya desplegado.

Una vez se ha desplegado un mensaje para ser visualizado por el usuario, este tendrá la opción de calificarlo positiva o negativamente, esto se logra mediante la implementación de dos funciones que se encargarán de realizar la tarea de modificar el dato de calificación que está asociado a cada consejo dentro de la base de datos. El siguiente código representa las funciones que se han descrito.

```

function CalificacionBuena(Calificacion, sender_psid)
{
  if(Calificacion == null){

    const unaVez ={
      "text": "Ya haz calificado. Solo puedes calificar una vez"
    };

    callSendAPI(sender_psid, unaVez);

  }else{
    var ref = HappBase.ref('Consejos/'+Calificacion);
    //Decrementa calificacion en 1
    ref.once('value').then(function(snapshot) {
      var Cal = (snapshot.val() && snapshot.val().Calificacion) || 0;
      var updates = {};
      updates['/Calificacion/'] = Cal-1;

      ref.update(updates);

    });
  }
}

function CalificacionMala(Calificacion, sender_psid)
{
  if(Calificacion == null){

    const unaVez ={
      "text": "Ya haz calificado. Solo puedes calificar una vez"
    };

    callSendAPI(sender_psid, unaVez);

  }else{
    var ref = HappBase.ref('Consejos/'+Calificacion);
    //Incrementa calificacion en 1
    ref.once('value').then(function(snapshot) {
      var Cal = (snapshot.val() && snapshot.val().Calificacion) || 0;
      var updates = {};
      updates['/Calificacion/'] = Cal+1;

      ref.update(updates);

    });
  }
}

```

Las funciones descritas anteriormente además de actualizar los datos de calificación para los consejos dentro de la base de datos permiten desplegar un

mensaje de advertencia informando al usuario si este ya ha calificado el consejo, ya que un usuario podrá calificar solo una vez el consejo que se le haya mostrado en turno según la interacción que se tenga con el Bot.

Para finalizar con las funciones especiales se necesita de una que acceda a la base de datos para obtener la información de algún usuario de manera aleatoria cuando el usuario que esté en la interacción con el Bot solicite que se le vincule con otro usuario. El siguiente fragmento de código muestra la función encargada de esta tarea.

```

function ObtenerUsuarios(sender_psid) {
    var ref = HappBase.ref('Usuarios/');
    .
    ref.once("value", function(data) {
        var keys1 = data.val();
        var keys = Object.keys(keys1);
        var indi = randomIntFromInterval(0, keys.length-1);

        console.log(keys);
        console.log(keys.length);
        console.log(keys[indi]);

        var GetUserario = keys[0];

        var refi = HappBase.ref('Usuarios/'+GetUserario);

        refi.once("value", function(data) {
            var Persona = data.val().PerfilFB;
            var Foto = data.val().FotoPerfil;

            const UsuarioCon ={
                "attachment": {
                    "type": "template",
                    "payload":{
                        "template_type":"generic",
                        "elements":[
                            {
                                "title":"CONOCE A ESTA PERSONA",
                                "image_url":Foto,
                                "subtitle":"Buscala por su nombre en Facebook: "+Persona,
                            }
                        ]
                    }
                }
            };

            callSendAPI(sender_psid, UsuarioCon);

        });

    });
}

```

Como se puede apreciar la función consulta la base de datos para obtener un usuario de manera aleatoria, asemejándose al mismo proceso que se realiza para

obtener los consejos, sin embargo, esta función también incluye la plantilla que será desplegada en la conversación con el usuario a manera de respuesta a la interacción solicitada, esta plantilla incluye la imagen, el nombre y los datos necesarios para poder vincular a los usuarios con otros usuarios de la aplicación.

5.3.6 BASE DE DATOS FIREBASE.

Para la implementación de la base de datos se trabaja con Firebase Database, la cual es una base de datos en tiempo real que se aloja en la nube, gracias a esta característica se puede contar con su alta disponibilidad, además de que la interfaz de trabajo que proporciona es relativamente sencilla de comprender y se adapta de manera eficaz a los propósitos de la aplicación.

La base de datos que se emplea al ser de tipo no relacional utiliza una estructuración diferente a las generalmente acostumbradas de tipo relacional, esta estructuración está basada principalmente en datos con formato tipo JSON. La principal característica que se busca explotar al utilizar esta base de datos es la sincronización de los datos con todos los usuarios en tiempo real, es decir que cuando un dato se modifique dentro de la base, todos los usuarios vean reflejados esos efectos de manera instantánea y simultánea. Adicionalmente la plataforma de Firebase permite una fácil implementación dentro del desarrollo de aplicaciones móviles y del lado del servidor que son los principales elementos que se utilizan en este caso específico para el manejo de datos.

Todos los datos se almacenan como un objeto de tipo JSON, es por esto por lo que la estructuración de la base puede verse reflejada como un árbol JSON. El siguiente árbol representa la estructuración de todos los datos con los que trabaja la aplicación, esto incluye información de usuarios, consejos e información estadística.

```

{
  "Usuarios": {
    "Indentificador Único": {
      "AnimoActual": ,
      "FotoPerfil": ,
      "MomentosBien": ,
      "MomentosMal": ,
      "PerfilFB": ,
      "PuntosAnimo": ,
      "UsuariID":
    }
  },
  "Consejos": {
    "Llave Push": {
      "Calificacion": ,
      "Texto": ,
    }
  },
  "PromedioGlobal": {
    "AnimoBien": ,
    "AnimoMal": ,
  }
}

```

Figura 5.6 Estructuración tipo JSON de los datos.

Como se puede ver, el árbol cuenta con tres nodos principales que son "Usuarios", "Consejos" y "PromedioGlobal". Estos nodos se utilizarán para almacenar los datos correspondientes con los cuales funcionará la aplicación.

El nodo "Usuarios" es dónde se almacenarán todos los datos referentes a cada usuario, cada que un usuario nuevo se registra a la base de datos se hace

asociándole un “Identificador Único” y se registra en el nodo de “Usuarios”. A su vez los principales datos que se almacenarán de cada usuario son:

- “AnimoActual”: Es un valor numérico.
- “FotoPerfil”: Es una cadena.
- “MomentosBien”: Es un valor numérico.
- “MomentosMal”: Es un valor numérico.
- “PerfilFB”: Es una cadena.
- “PuntosAnimo”: Es un valor numérico.
- “UsuarioID”: Es un valor numérico.

Como se puede ver estos datos almacenarán un tipo de dato específico, sin embargo, en este tipo de base de datos no se especifican dentro de la estructuración ya que pueden ser cualquiera de los tipos nativos que corresponden a los tipos de JSON disponibles. A pesar de esto, cabe mencionar que el tipo de datos que se almacenan dentro de cada nodo están determinados en las funciones implementadas por el Bot y que han sido descritas anteriormente, hay que tener en cuenta el tipo de dato con el que se está trabajando para poder acceder a este posteriormente sin generar inconsistencias.

Por otro lado, el nodo “Consejos” es dónde se almacenarán todos los consejos ingresados por los usuarios, cada vez que se ingresa un consejo la base de datos generará una llave para identificarlo a través del método “push()” el cual genera una llave única cada vez que se agrega un elemento secundario a la referencia especificada. Los datos que se almacenan para cada consejo son:

- “Calificacion”: Es un valor numérico.
- “Texto”: Es una cadena.

En el nodo “PromedioGlobal” se almacena información generada por la comunidad de usuarios de manera general, los principales datos que se almacenan son:

- “AnimoBien”: Es un valor numérico.
- “AnimoMal”: Es un valor numérico.

Para poder trabajar con la base de datos de Firebase es necesario realizar ciertas configuraciones dentro de nuestro sistema, por un lado, es necesario descargar las dependencias necesarias para poder implementar las funciones de la base del lado del Bot y por otro lado es necesario crear un vínculo con la aplicación móvil Android para hacer uso de las funciones correspondientes. Firebase permite el desarrollo y la implementación multiplataforma ya que se puede acceder a la base de datos desde un dispositivo móvil o una aplicación web como lo es en este caso el Bot. Adicionalmente cabe destacar que Firebase no necesita un servidor de aplicaciones para su implementación por lo que se puede gestionar la base directamente desde la aplicación móvil.

Posteriormente se descarga un archivo de configuración generado por Firebase que se añadirá al proyecto en Android. Ese archivo de configuración es de tipo JSON y contiene toda la información necesaria del proyecto para poder realizar la conexión con la aplicación móvil que se ha registrado por medio de la consola.

La base de datos tendrá libre escritura y consulta ya que se busca que cualquier usuario pueda generar datos a través del Bot sin tener que autenticarse, eso significa que no se crea ninguna restricción para acceder a los datos, solamente se necesita una conexión autorizada a la base la cual se realiza a través del Bot y la aplicación móvil. Esto se logra por medio de las reglas de la base de datos, el siguiente fragmento representa esta configuración de reglas dentro de la base de datos:

```
{  
  "rules": {  
    ".read": "true",  
    ".write": "true"  
  }  
}
```

5.3.7 PROGRAMACIÓN DE LA APLICACIÓN ANDROID.

El desarrollo y programación de la aplicación móvil se realizó a través de “Android Studio” el cual es un entorno de desarrollo integrado para sistemas operativos Android. En general la aplicación se compone de diferentes actividades individuales que proporcionan una sola pantalla en la cual se despliega la interfaz de usuario, mientras en segundo plano se realizan actividades o funciones programadas que en este caso son cálculos, y accesos a la base de datos.

La programación de las diferentes actividades se realiza a través de lenguaje Java, para hacer uso de este es necesario descargar el JDK oficial de java, el cuál es un kit de desarrollo que nos permite la implementación de lenguaje Java dentro del entorno de Android Studio.

El flujo lógico de las actividades programadas para la aplicación se describe a continuación en el siguiente diagrama.

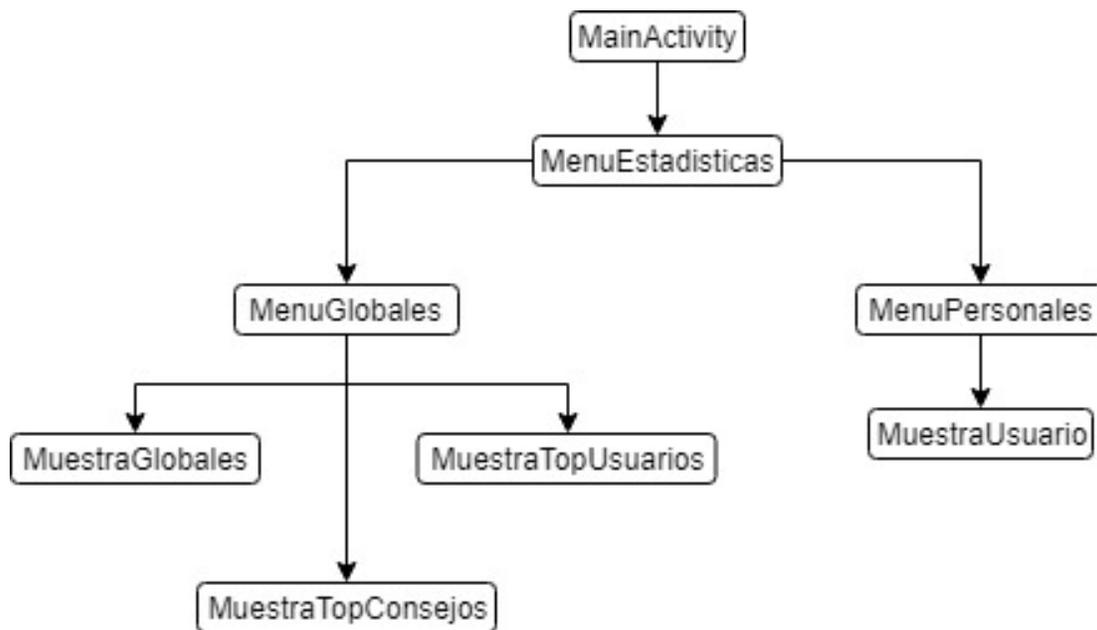


Figura 5.7 Diagrama de flujo para actividades Android

La primera actividad “MainActivity” representa un menú de inicio para el usuario en el cual se despliegan a través de dos botones las opciones para interactuar con el Bot y consultar información estadística. El siguiente fragmento de código representa la programación para la primera actividad.

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}

/** Llamada cuando el usuario toca el botón */
public void GoToMessenger(View view) {
    // Do something in response to button
    Intent IrBOT = new Intent(Intent.ACTION_VIEW, Uri.parse("https://m.me/HappAppUNAM"));
    startActivity(IrBOT);
}

/** Llamada cuando el usuario toca el botón */
public void GoToEstadisticas(View view) {
    Intent intent = new Intent( packageContext: this, MenuEstadisticas.class);
    startActivity(intent);
}

```

Como se puede ver el primer botón redirecciona al usuario a la conversación con el Bot a través de la plataforma de Facebook Messenger, el segundo botón inicia la segunda actividad dentro de la aplicación, cabe mencionar que la aplicación realiza las acciones descritas a través de intenciones programadas dentro de cada función.

Para la segunda actividad “MenuEstadisticas” de igual forma desplegará un menú presentando al usuario las opciones de estadísticas a consultar, que en este caso son las estadísticas por usuario y las estadísticas globales. El siguiente fragmento de código representa la programación de esta segunda actividad.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_menu_estadisticas);
}

public void GoToPersonales(View view) {
    Intent intent = new Intent( packageContext: this, MenuPersonales.class);
    startActivity(intent);
}

public void GoToGlobales(View view) {
    Intent intent = new Intent( packageContext: this, MenuGlobales.class);
    startActivity(intent);
}

```

Al igual que en la primera actividad, los botones dentro de esta segunda actividad nos dirigen a otras actividades por medio de las intenciones programadas.

Las actividades subsecuentes se activan dependiendo del botón que se seleccione durante la segunda actividad. En el caso de que se seleccione la consulta de estadísticas personales, se activará la actividad “MenuPersonales” en la cual se

solicitará al usuario que proporcione su identificador único para consultar estadísticas y un botón de envío, esto enviará el identificador a la actividad siguiente. El siguiente fragmento de código representa lo descrito anteriormente.

```
public static final String EXTRA_MESSAGE = "mx.unam.prueba.MESSAGE";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_menu_personales);
}

public void enviarUsuario(View view) {
    Intent intent = new Intent( packageContext: this, MuestraUsuario.class);
    EditText editText = (EditText) findViewById(R.id.editText);
    String message = editText.getText().toString();
    intent.putExtra(EXTRA_MESSAGE, message);
    startActivity(intent);
}
```

Como podemos ver el identificador único se almacenará en una variable de tipo cadena para su posterior análisis. La actividad siguiente a esta actividad recibe el identificador único y lo gestiona a través de una función para poder acceder a la base de datos.

La actividad “MuestraUsuario” recibirá el identificador único de la actividad anterior, con este identificador único se podrá acceder a la referencia específica dentro de la base de datos del usuario al que pertenece el identificador, de este modo se podrá obtener su información. Por medio de la información obtenida de la base de datos que corresponde al usuario, las funciones implementadas dentro de la actividad podrán realizar su programación utilizando los datos obtenidos. A grandes rasgos las funciones se encargan de gestionar los datos para desplegar en pantalla las gráficas que representan el estado de ánimo actual del usuario, así como su índice de ánimo que indica que tan feliz es ese usuario, también despliega una gráfica circular que compara los momentos felices contra los momentos tristes que ha tenido el usuario. En los siguientes fragmentos de código se representa la programación de esta actividad y todas sus funciones.

```

// Obtiene la intencion que inicio la actividad y extrae la cadena
Intent intent = getIntent();
String UserID = intent.getStringExtra(MenuPersonales.EXTRA_MESSAGE);

FirebaseDatabase database = FirebaseDatabase.getInstance();
final DatabaseReference myRef = database.getReference("Usuarios/"+UserID);

myRef.child("PerfilFB").addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        String message = dataSnapshot.getValue(String.class);

        if (message != null) {
            TextView textView = findViewById(R.id.textView);
            textView.setText("ESTADISTICAS DE:\n "+message);

            Caras(myRef);
            Bien(myRef);
        }else{
            message= "IDENTIFICADOR NO VÁLIDO, INTENTA DE NUEVO";
            TextView textViewerr = findViewById(R.id.textViewError);
            textViewerr.setText(message);
        }
    }
}
}

```

El código anterior se encarga de extraer la cadena proporcionada por la intención que inicio la actividad, después la asigna a una variable que es utilizada para acceder a la referencia que indica la base de datos. Si la referencia es accedida exitosamente se procede con las siguientes funciones, si la referencia es errónea, se indica que el identificador no es válido.

```

private void Bien(DatabaseReference Ref){

    Ref.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) { setData(dataSnapshot); }

        @Override
        public void onCancelled(DatabaseError databaseError) {
            String message = "Un error ha ocurrido al obtener los datos";

            TextView textView = findViewById(R.id.textViewError);
            textView.setText(message);
        }
    });
}
}

```

El código anterior obtiene los datos de la referencia a la cual se consultó y los pasa como parámetro a la función encargada de desplegar en pantalla las gráficas con los datos especificados.

```
private void setData(DataSnapshot dataSnapshot) {  
  
    int Happy = dataSnapshot.child("MomentosBien").getValue(Integer.class);  
    int Sad = dataSnapshot.child("MomentosMal").getValue(Integer.class);  
  
    pieChart = (PieChart) findViewById(R.id.chart1);  
  
    ArrayList<PieEntry> yVals = new ArrayList<>();  
    yVals.add(new PieEntry(Happy, label: "FELICES"));  
    yVals.add(new PieEntry(Sad, label: "TRISTES"));  
  
    PieDataSet set = new PieDataSet(yVals, label: "");  
    set.setColors(ColorTemplate.MATERIAL_COLORS);  
    set.setSliceSpace(2f);  
  
    Description description = new Description();  
    description.setText("MOMENTOS QUE SE HAN TENIDO ");  
    description.setTextSize(11);  
    pieChart.setDescription(description);  
  
    PieData data = new PieData(set);  
  
    data.setValueTextSize(20);  
    data.setValueTextColor(Color.WHITE);  
  
    pieChart.setData(data);  
    pieChart.invalidate();  
}
```

El código anterior corresponde a la función encargada de desplegar en pantalla las estadísticas en forma de gráfico circular que compara los momentos tristes contra los momentos felices, para lograr esto con éxito se utilizaron las dependencias de “MPAndroidChart”, la cual proporciona funciones auxiliares para generar gráficas dinámicas, esto se usa para poder reflejar los datos en tiempo real cuando los datos son modificados en la base de datos.

Adicionalmente existe una función que se encarga de desplegar en pantalla el estado actual del usuario representado por una imagen, así como de calcular el índice de felicidad de este mismo usuario para ser desplegado en pantalla. Se anexa a continuación el código de esa función que representa un anidamiento condicional

en el cual se selecciona la imagen a desplegar con base a los datos consultados, y se realizan los cálculos pertinentes para obtener el índice de felicidad del usuario.

```
private void Caras(DatabaseReference Ref){

    Ref.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {

            int AnimoActual =
dataSnapshot.child("AnimoActual").getValue(Integer.class);

            TextView textViewCara = findViewById(R.id.textViewCara);
            textViewCara.setText("Estado\nActual ");

            int AnimoMal =
dataSnapshot.child("MomentosMal").getValue(Integer.class);
            int AnimoBien =
dataSnapshot.child("MomentosBien").getValue(Integer.class);
            int IndiceAnimo = 0;

            if (AnimoBien > AnimoMal){
                IndiceAnimo = 10-((10*AnimoMal)/AnimoBien);

                String In = "";

                In = Integer.toString(IndiceAnimo);

                TextView textViewIndice = findViewById(R.id.textViewIndice);
                textViewIndice.setText("Indice de\nAnimo ");

                TextView textViewNumIndi = findViewById(R.id.textViewNumIndi);
                textViewNumIndi.setText(In);

            } else if(AnimoMal > AnimoBien){
                IndiceAnimo = -(10-((10*AnimoBien)/AnimoMal));

                String In = "";

                In = Integer.toString(IndiceAnimo);

                TextView textViewIndice = findViewById(R.id.textViewIndice);
                textViewIndice.setText("Indice de\nAnimo ");

                TextView textViewNumIndi = findViewById(R.id.textViewNumIndi);
                textViewNumIndi.setText(In);
            } else if(AnimoMal == AnimoBien){
                IndiceAnimo = 0;

                String In = "";

                In = Integer.toString(IndiceAnimo);

                TextView textViewIndice = findViewById(R.id.textViewIndice);
                textViewIndice.setText("Indice de\nAnimo ");

                TextView textViewNumIndi = findViewById(R.id.textViewNumIndi);
                textViewNumIndi.setText(In);

            }
        }
    });
}
```

```

        ImageView imageView = findViewById(R.id.Carita);
        if(AnimoActual == 1) {
            imageView.setImageResource(R.drawable.happy);
        } else if(AnimoActual == 2){
            imageView.setImageResource(R.drawable.neutral);
        } else if(AnimoActual == 3){
            imageView.setImageResource(R.drawable.sad);
        }
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

        String message = "Un error ha ocurrido al obtener los datos";

        TextView textView = findViewById(R.id.textViewError);
        textView.setText(message);
    }
});
}
}

```

La siguiente actividad a describir es aquella que se activa cuando el usuario selecciona consultar estadísticas globales, es entonces que la actividad “MenuGlobales” despliega en pantalla un menú que corresponde a todas las opciones de consulta de datos globales que se pueden obtener, que principalmente son top 10 de Consejos, top 10 de usuarios, y estadísticas globales.

Cuando se selecciona la opción de estadísticas globales se procede a ejecutar la actividad “MuestraGlobales” que guarda mucha similitud con la actividad que muestra las estadísticas de cada usuario. Esta actividad utiliza también las funciones de la dependencia “MPAndroidChart” para desplegar en pantalla una gráfica circular, del mismo modo se mostrará en pantalla el índice de felicidad global y los porcentajes de comparativos de felicidad y tristeza que presenta la comunidad de usuarios. A continuación, se anexa el código de esa actividad para demostrar cómo es que se logra realizar dichas funcionalidades.

```

public class MuestraGlobales extends AppCompatActivity {

    private PieChart pieChart;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_muestra_globales);

        FirebaseDatabase database = FirebaseDatabase.getInstance();
        final DatabaseReference myRef =
        database.getReference("PromedioGlobal");

        myRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                int AnimoMal =

```

```

dataSnapshot.child("AnimoMal").getValue(Integer.class);
    int AnimoBien =
dataSnapshot.child("AnimoBien").getValue(Integer.class);
    int IndiceAnimo = 0;

    if (AnimoBien > AnimoMal){
        IndiceAnimo = 10-((10*AnimoMal)/AnimoBien);

        String In = "";

        In = Integer.toString(IndiceAnimo);

        TextView textViewIndice =
findViewById(R.id.textViewIndice);
        textViewIndice.setText("Indice de\nAnimo ");

        TextView textViewRango = findViewById(R.id.textViewRango);
        textViewRango.setText("Rango del indice\nDe -10 a 10");

        TextView textViewNumIndi =
findViewById(R.id.textViewNumIndi);
        textViewNumIndi.setText(In);

    } else if(AnimoMal > AnimoBien){
        IndiceAnimo = -(10-((10*AnimoBien)/AnimoMal));

        String In = "";

        In = Integer.toString(IndiceAnimo);

        TextView textViewIndice =
findViewById(R.id.textViewIndice);
        textViewIndice.setText("Indice de\nAnimo ");

        TextView textViewNumIndi =
findViewById(R.id.textViewNumIndi);
        textViewNumIndi.setText(In);
    } else if(AnimoMal == AnimoBien){
        IndiceAnimo = 0;

        String In = "";

        In = Integer.toString(IndiceAnimo);

        TextView textViewIndice =
findViewById(R.id.textViewIndice);
        textViewIndice.setText("Indice de\nAnimo ");

        TextView textViewNumIndi =
findViewById(R.id.textViewNumIndi);
        textViewNumIndi.setText(In);
    }

    pieChart = (PieChart) findViewById(R.id.chart1);
    pieChart.setUsePercentValues(true);

    //pieChart.getDescription().setEnabled(false);

    ArrayList<PieEntry> yVals = new ArrayList<>();
    yVals.add(new PieEntry(AnimoBien, "PORCENTAJE\nFELICIDAD"));
    yVals.add(new PieEntry(AnimoMal, "PORCENTAJE\nTRISTEZA"));

```

```

PieDataSet set = new PieDataSet(yVals, "");
set.setColors(ColorTemplate.MATERIAL_COLORS);
set.setSliceSpace(2f);

Description description = new Description();
description.setText("Comparación de Felicidad y Tristeza a
nivel Global");
description.setTextSize(11);
pieChart.setDescription(description);

PieData data = new PieData(set);

data.setValueTextSize(20);
data.setValueTextColor(Color.WHITE);

pieChart.setData(data);
pieChart.invalidate();

ImageView imageView = findViewById(R.id.Carita);
if(IndiceAnimo >=2 && IndiceAnimo <=10) {
    imageView.setImageResource(R.drawable.happy);
} else if(IndiceAnimo < 2 && IndiceAnimo > -2){
    imageView.setImageResource(R.drawable.neutral);
} else if(IndiceAnimo >=-10 && IndiceAnimo <= -2){
    imageView.setImageResource(R.drawable.sad);
}

}

@Override
public void onCancelled(DatabaseError databaseError) {

    String message = "Ha ocurrido un error al obtener los datos";

    TextView textView = findViewById(R.id.textViewError);
    textView.setText(message);

}

});
}
}

```

Por otro lado, si el usuario selecciona ver algún el top 10 de usuarios, o el top 10 de consejos se desplegará en pantalla una lista con los 10 elementos según corresponda a lo que se ha seleccionado. Este orden se logra para el caso de los consejos a través de las calificaciones otorgadas por los usuarios que son datos que se almacenan en la base de datos, y para el orden de los usuarios el criterio utilizado es de acuerdo con los puntos de ánimo que la persona haya obtenido, estos se registran del mismo modo en la base de datos mediante la interacción con el Bot.

A continuación, se anexa el código de la programación que permite desplegar el top de consejos:

```

Query TopConsejos = myRef.orderByChild("Calificacion").limitToFirst(10);

TopConsejos.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        for (DataSnapshot userSnapshot: dataSnapshot.getChildren()) {

            String Consejo =
userSnapshot.child("Texto").getValue(String.class);
            Consejos.add(Consejo);
            int Calificacion = -
(userSnapshot.child("Calificacion").getValue(Integer.class));

            Normalizados.add(Calificacion);
        }

        try {

            int Diez = Normalizados.get(0);
            Calificaciones.add("10");
            for ( int Cali = 1; Cali <= 9; Cali ++ ) {
                int Rating = ((Normalizados.get(Cali))*10)/Diez;
                String In = "";
                In = Integer.toString(Rating);
                Calificaciones.add(In);
            }

            String datos [][] = {
                {Consejos.get(0), Calificaciones.get(0)},
                {Consejos.get(1), Calificaciones.get(1)},
                {Consejos.get(2), Calificaciones.get(2)},
                {Consejos.get(3), Calificaciones.get(3)},
                {Consejos.get(4), Calificaciones.get(4)},
                {Consejos.get(5), Calificaciones.get(5)},
                {Consejos.get(6), Calificaciones.get(6)},
                {Consejos.get(7), Calificaciones.get(7)},
                {Consejos.get(8), Calificaciones.get(8)},
                {Consejos.get(9), Calificaciones.get(9)}
            };

            Pinta2(datos);

        } catch (Exception excepcion) {
            TextView textView = findViewById(R.id.textViewError2);
            textView.setText("NO HAY SUFICIENTES CONSEJOS");
        }

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }

});
}

```

Del mismo modo se anexa el código de la programación que permite desplegar el top de usuarios:

```
Query TopUsuarios = myRef.orderByChild("PuntosAnimo").limitToFirst(10);

TopUsuarios.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        for (DataSnapshot userSnapshot: dataSnapshot.getChildren()) {

            String Usuario =
userSnapshot.child("PerfilFB").getValue(String.class);
            Usuarios.add(Usuario);
            int Indice = -
(userSnapshot.child("PuntosAnimo").getValue(Integer.class));
            String In = "";
            In = Integer.toString(Indice);
            Indices.add(In);
            String UID =
userSnapshot.child("UsuarioID").getValue(String.class);
            UIDS.add(UID);
            String Foto =
userSnapshot.child("FotoPerfil").getValue(String.class);
            Fotos.add(Foto);

        }

        try {
            String datos [][] = {
                {Usuarios.get(0),
Indices.get(0),UIDS.get(0),Fotos.get(0)},
                {Usuarios.get(1),
Indices.get(1),UIDS.get(1),Fotos.get(0)},
                {Usuarios.get(2),
Indices.get(2),UIDS.get(2),Fotos.get(0)},
                {Usuarios.get(3),
Indices.get(3),UIDS.get(3),Fotos.get(0)},
                {Usuarios.get(4),
Indices.get(4),UIDS.get(4),Fotos.get(0)},
                {Usuarios.get(5),
Indices.get(5),UIDS.get(5),Fotos.get(0)},
                {Usuarios.get(6),
Indices.get(6),UIDS.get(6),Fotos.get(0)},
                {Usuarios.get(7),
Indices.get(7),UIDS.get(7),Fotos.get(0)},
                {Usuarios.get(8),
Indices.get(8),UIDS.get(8),Fotos.get(0)},
                {Usuarios.get(9),
Indices.get(9),UIDS.get(9),Fotos.get(0)}
            };

            int[] datosImg = {R.drawable.happy , R.drawable.happy,
R.drawable.happy, R.drawable.happy, R.drawable.happy, R.drawable.happy,
R.drawable.happy, R.drawable.happy, R.drawable.happy, R.drawable.happy};

            Pinta(datos, datosImg);

        } catch (Exception excepcion) {
            TextView textView = findViewById(R.id.textViewError1);
            textView.setText("NO HAY SUFICIENTES USUARIOS");
        }
    }
});
```

```
    }  
  
    @Override  
    public void onCancelled(DatabaseError databaseError) {  
    }  
    }  
});  
}
```

Como podemos apreciar en los dos fragmentos de código anexados anteriormente, ambos implementan una consulta hacia la base de datos donde se filtran los datos para obtener los primeros diez elementos según se requiere. Los tops muestran adicionalmente una imagen para cada usuario, y un sistema visual de calificación para los consejos basados en estrellas.

Finalmente cabe mencionar que todas las actividades descritas anteriormente cuentan con un diseño implementado dentro de la interfaz gráfica que visualiza el usuario, estos diseños son adaptables a la mayoría de los dispositivos Android, en este caso específico serán absolutamente compatibles con versiones de sistema operativo superiores a Android 6.0 Marshmallow.

5.3.8 GENERACIÓN DE GRÁFICAS.

Como se describió anteriormente el usuario tiene acceso a la información estadística generada a través del uso de la plataforma mediante la interacción con el Bot. Para agregar valor a los datos y que estos sean de mayor interés para el usuario, se representan mediante la generación de gráficas que por su diseño sencillo y llamativo permitirán al usuario la fácil comprensión y análisis de las estadísticas con la finalidad de que pueda hacer uso de esta información en su beneficio para mejorar su estado de ánimo.

Para lograr la generación correcta dentro de la aplicación de Android se ha utilizado el software gratuito “MPAndroidChart” el cuál es una dependencia que se puede utilizar en los proyectos de desarrollo Android. Este software nos ofrece ciertas funciones preprogramadas para la representación de datos a través de graficas desplegables para dispositivos Android.

La dependencia de “MPAndroidChart” se instala en el proyecto Android al añadir a las dependencias del código el siguiente fragmento.

```
dependencies {  
    implementation 'com.github.PhilJay:MPAndroidChart:v3.0.3'  
}
```

Se hace uso de la última versión de la dependencia que es de libre acceso en github.

Para poder generar las gráficas se necesita consultar la información que será representada directamente a la base de datos. Esta acción se ve representada por el siguiente fragmento de código.

```
FirebaseDatabase database = FirebaseDatabase.getInstance();  
final DatabaseReference myRef = database.getReference(S: "PromedioGlobal");  
  
myRef.addValueEventListener(new ValueEventListener() {  
    @Override  
    public void onDataChange(DataSnapshot dataSnapshot) {  
        int AnimoMal = dataSnapshot.child("AnimoMal").getValue(Integer.class);  
        int AnimoBien = dataSnapshot.child("AnimoBien").getValue(Integer.class);  
    }  
});
```

El código anterior hace referencia a los datos utilizados para generar las estadísticas globales. Como se puede apreciar en el código, para poder acceder a los datos estadísticos que son relevantes para la representación gráfica se ha creado una referencia a la base de datos para acceder a los nodos específicos que contienen la información necesaria. Cabe mencionar que esta función es una función que escucha a cualquier cambio realizado dentro de la base de datos, lo cual permite dotar a la aplicación de Android con una generación de gráficas en tiempo real que

se modificará con base a los cambios que se efectúen directamente en la base de datos.

Para poder trabajar con las funciones correspondientes a “MPAndroidChart” es necesario declarar una variable del tipo “pieChart” la cual hace referencia a una gráfica de tipo circular la cual es proporcionada por la dependencia de “MPAndroidChart”.

```
private PieChart pieChart;
```

Para el caso específico de las estadísticas globales se realiza una representación gráfica de la información de manera porcentual, es decir que los datos estadísticos obtenidos de la base de datos son representados en porcentajes. Para lograr esto se hace uso del método “setUsePercentValues” que se asocia a la variable PieChart declarada con anterioridad.

```
pieChart = (PieChart) findViewById(R.id.chart1);  
pieChart.setUsePercentValues(true);
```

Los datos que se obtuvieron anteriormente desde la base de datos se almacenan en un objeto “ArrayList” que almacena objetos que contienen los datos, asignando también las etiquetas que cada dato tendrá dentro de la gráfica. Posteriormente se crea un objeto “PieDataSet” que permite insertar los datos que serán representados gráficamente, así como también brinda la oportunidad de establecer los colores que serán usados y el tamaño que tendrá la gráfica. El siguiente fragmento de código representa lo descrito anteriormente.

```
ArrayList<PieEntry> yVals = new ArrayList<>();  
yVals.add(new PieEntry(AnimoBien, label: "PORCENTAJE\nFELICIDAD"));  
yVals.add(new PieEntry(AnimoMal, label: "PORCENTAJE\nTRISTEZA"));  
  
PieDataSet set = new PieDataSet(yVals, label: "");  
set.setColors(ColorTemplate.MATERIAL_COLORS);  
set.setSliceSpace(2f);
```

Para la parte de las acotaciones de la gráfica se utiliza un objeto del tipo “Description” mediante el cual se pasan los parámetros específicos que se desplegarán en la gráfica que en este caso es una acotación textual que indica que clase de información es la que la gráfica está representando. Adicionalmente se agrega a la gráfica un diseño circular central que hace la gráfica más llamativa. Lo anterior mente descrito se representa con el siguiente fragmento de código.

```
Description description = new Description();
description.setText("Comparación de Felicidad y Tristeza a nivel Global");
description.setTextSize(11);
pieChart.setDescription(description);
pieChart.setHoleColor(Color.rgb( red: 49, green: 82, blue: 147));
```

Finalmente se pasan todos los datos que se obtienen a un objeto "PieData" y se le da el formato a los textos que acotan la información dentro de la gráfica, que en este caso son tanto el color negro y un tamaño de proporción 20. Por último se añade la información a la gráfica y se despliega. El siguiente fragmento de código representa lo descrito anteriormente.

```
PieData data = new PieData(set);

data.setValueTextSize(20);
data.setValueTextColor(Color.BLACK);

pieChart.setData(data);
pieChart.invalidate();
```

Para la otra gráfica que se encuentra disponible en la aplicación que representa los datos por usuario se siguieron pasos similares, únicamente cambiando la información que se iba a representar. Con la finalidad de que el diseño de las gráficas fuera consistente.

CAPÍTULO 6

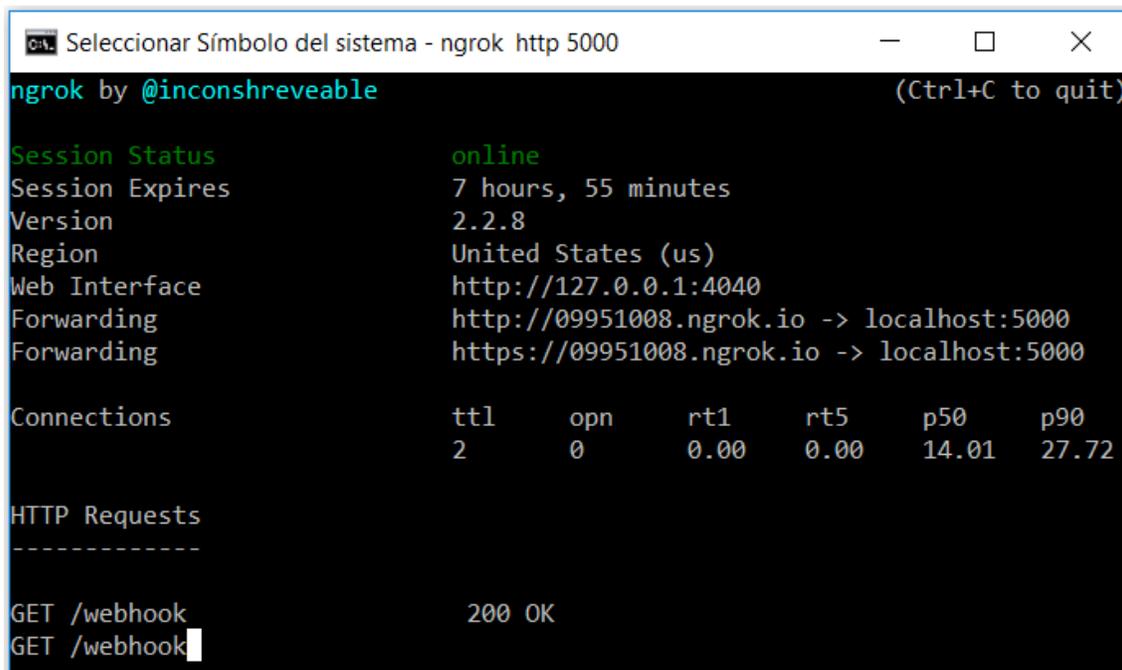
PRUEBAS

CAPÍTULO 6. PRUEBAS.

6.1 PRUEBAS CON EL BOT.

Tras concluir con el proceso de elaboración del Bot es necesario verificar que todas las funciones que se establecieron dentro de su programación se efectúan de manera correcta, para lograr esto el Bot se ejecutará en una computadora mediante Node.js así mismo la máquina tendrá en ejecución Ngrok, que a su vez habilitará el puerto 5000 de la máquina convirtiéndola en un servidor HTTP virtual para poder recibir las solicitudes y eventos que manda la plataforma de Facebook Messenger a la dirección que se le proporciona para la suscripción del “webhook”.

En las siguientes figuras se muestra el éxito de la conexión del Bot con la plataforma de Facebook Messenger a través del servidor implementado con Ngrok



```
Seleccionar Símbolo del sistema - ngrok http 5000
ngrok by @inconshreveable (Ctrl+C to quit)
Session Status      online
Session Expires    7 hours, 55 minutes
Version             2.2.8
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding          http://09951008.ngrok.io -> localhost:5000
Forwarding          https://09951008.ngrok.io -> localhost:5000

Connections
  ttl   opn   rt1   rt5   p50   p90
   2    0    0.00  0.00  14.01  27.72

HTTP Requests
-----
GET /webhook          200 OK
GET /webhook
```

Figura 6.1 Pantalla que muestra la ejecución de Ngrok.

En la captura de pantalla anterior se muestra la ejecución de Ngrok la cual genera una dirección con SSL válido y HTTPS los cuales son requeridos por la plataforma de Facebook para poder realizar la conexión del Bot, por otro lado, también se muestra que es el puerto 5000 de la máquina el que está en uso para la recepción y envío de datos.

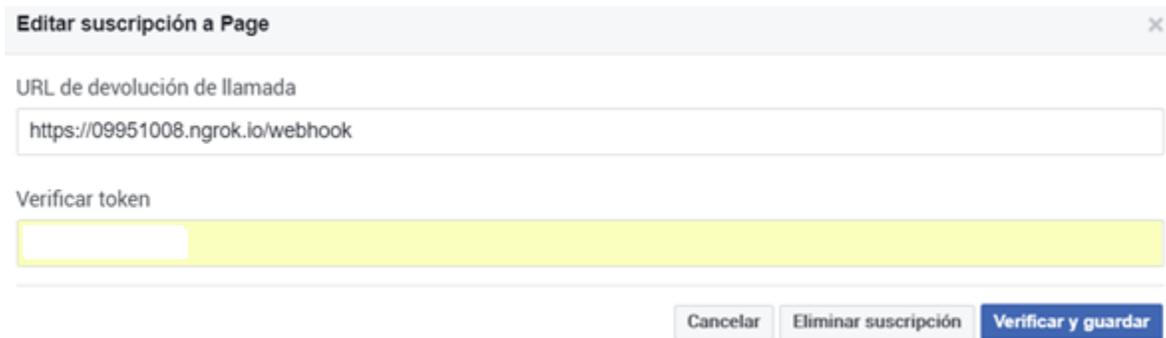


Figura 6.2 Pantalla de suscripción del “webhook”.

En la captura de pantalla anterior se muestra la suscripción que se realiza directamente en la plataforma de Facebook para desarrolladores utilizando la dirección que se genera a través de Ngrok.

Finalmente, la siguiente captura de pantalla demuestra la correcta compilación del Bot a través de Node.js, ya que no se produce ningún error al ejecutarlo, y del mismo nodo se aprecia la verificación del “webhook” lo que demuestra que la conexión con la plataforma de Facebook Messenger ha sido exitosa.

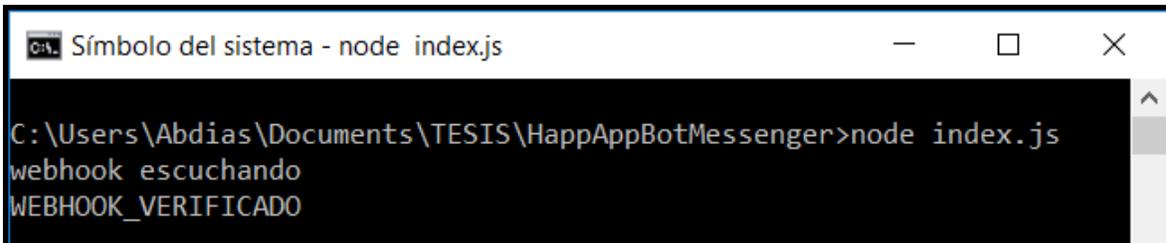


Figura 6.3 Pantalla de ejecución del Bot.

6.1.1 INTERACCIÓN DEL BOT CON EL USUARIO.

Una vez el Bot se encuentra ejecutándose en el servidor, el usuario podrá interactuar con él a través de una ventana conversacional. Para poder interactuar con el Bot el usuario debe iniciar una conversación con la página de Facebook que funge como entidad del Bot que en este caso es "HappApp".

A continuación, se muestra el proceso de interacción con el Bot mediante el cual se prueba el correcto funcionamiento de este.



Figura 6.4 Interacción con el Bot en Messenger.

Como se puede apreciar en la captura de pantalla anterior, el Bot cumple con su objetivo de desplegar un mensaje de bienvenida para el usuario, así como el botón "Empezar" con el que se aceptan las interacciones con el Bot.

Posteriormente se prueban las interacciones con el Bot de acuerdo con lo establecido en el diagrama de flujo lógico.



Figura 6.5 Interacción con el Bot en Messenger.

En la captura de pantalla anterior podemos apreciar que al pulsar el botón “Empezar”, con el que se aceptan las interacciones con el Bot, el Bot entra en acción y despliega el Identificador Único que se ha asignado al usuario, del mismo modo despliega la plantilla en la que se pregunta por el estado de ánimo del usuario y tres botones con diferentes opciones las cuales al ser pulsadas iniciarán una nueva interacción con el Bot.

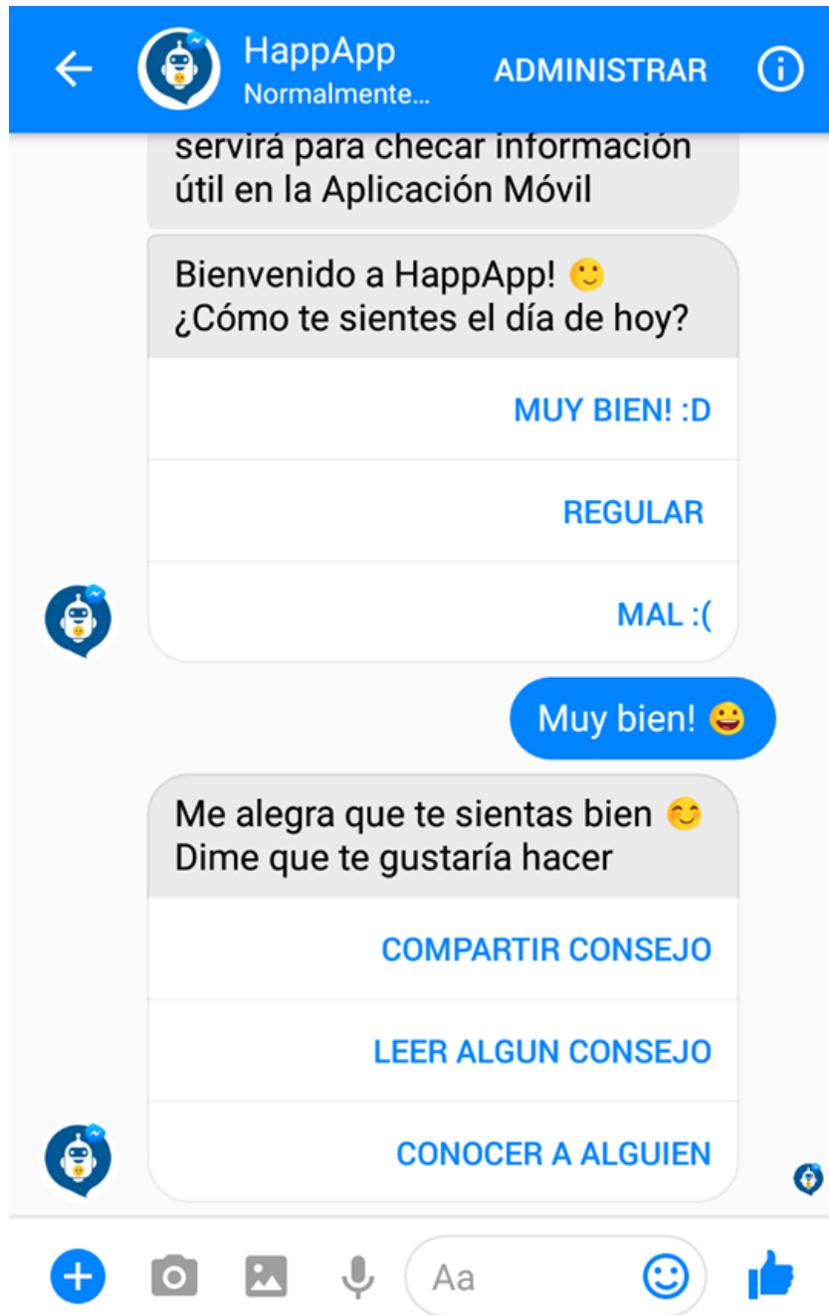


Figura 6.6 Interacción con el Bot en Messenger.

Al presionar el botón “Muy bien” se inicia otra interacción con el Bot que despliega las actividades que se pueden realizar en una plantilla con tres botones y las opciones para compartir consejo, leer consejo o conocer a alguien. Esto se demuestra en la captura de pantalla anterior.

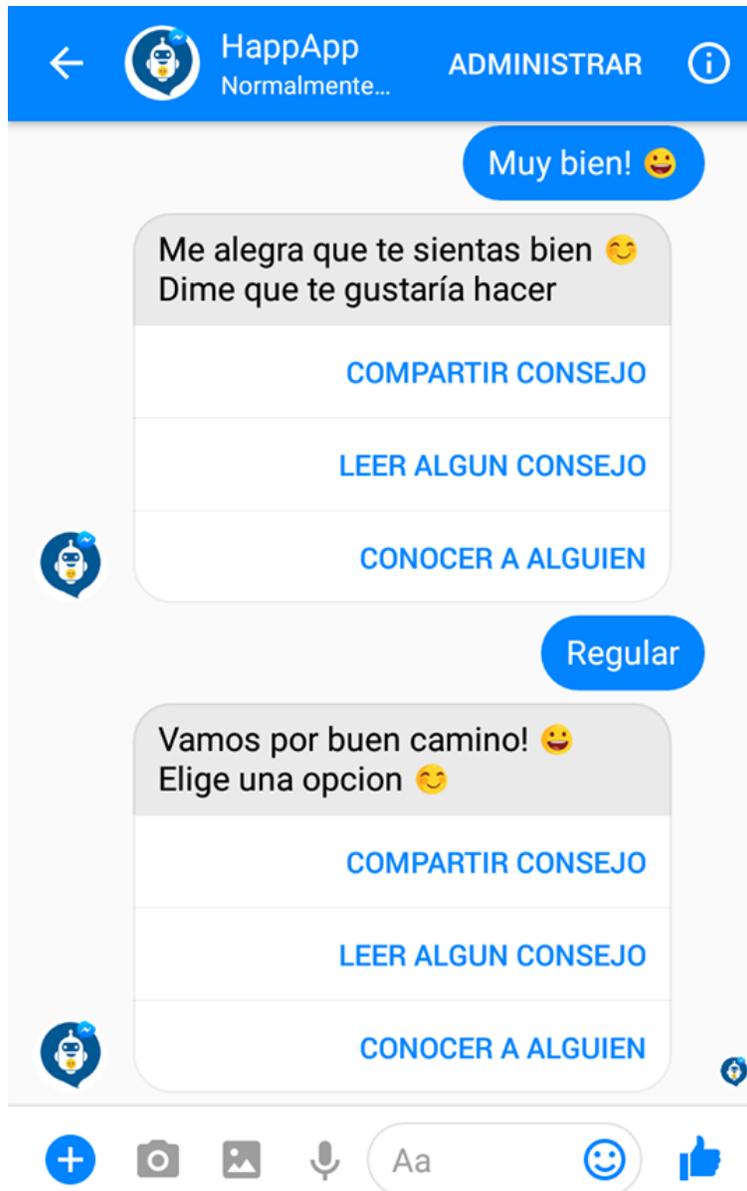


Figura 6.7 Interacción con el Bot en Messenger.

En la captura de pantalla anterior se demuestra el caso en el que el usuario presiona la opción “Regular”, como se puede ver el Bot responde a esta interacción desplegando una plantilla con un mensaje y tres botones con las opciones para compartir consejo, leer consejo o conocer a alguien. Cabe mencionar que también podemos apreciar que el usuario puede saltar a través de las interacciones con el Bot sin seguir una secuencia lógica, esto facilita la interacción del Bot con el usuario o corrección de errores en caso de que el usuario haya presionado por equivocación un botón el cual no quería presionar.



Figura 6.8 Interacción con el Bot en Messenger.

La captura de pantalla anterior muestra la respuesta del Bot cuando el usuario elige la opción “Mal”, como se puede observar la respuesta es una plantilla con mensaje y tres botones con opciones de leer consejo, conocer a alguien y consultar ayuda especializada.

Ahora se procede a probar las opciones especiales que ofrece el Bot tras la interacción con el usuario. Cuando el usuario pulsa la opción de “Leer algún consejo”, el Bot deberá desplegar en pantalla uno de los consejos que se encuentran almacenados en la base de datos el cual se selecciona de manera aleatoria, tras haberlo desplegado en pantalla el Bot también deberá proporcionar

una plantilla al usuario a través de la cual podrá calificar el consejo obtenido. La siguiente captura de pantalla demuestra que el Bot cumple con lo descrito anteriormente.

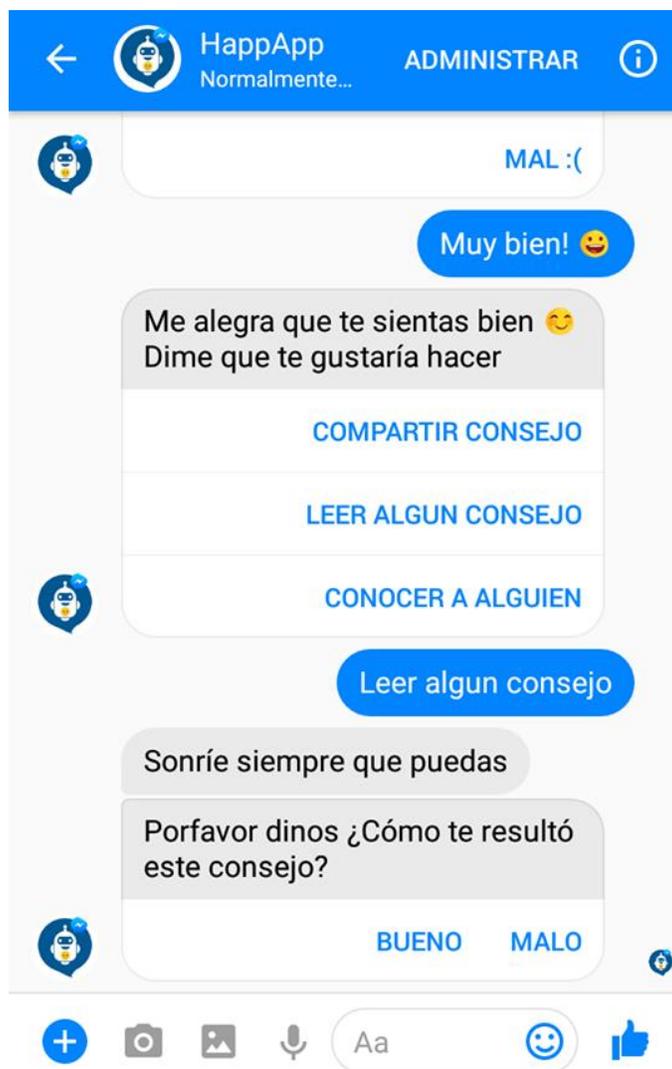


Figura 6.9 Interacción con el Bot en Messenger.

Después de recibir un consejo, el usuario podrá calificarlo a través de una plantilla proporcionada por el Bot que tiene dos botones con las opciones “Bueno” o “Malo”, al pulsar alguna de las opciones el Bot registrará la calificación y desplegará el mensaje correspondiente (Figura 6.9).

Cuando el usuario pulsa la opción de “Compartir Consejo”, esto hará que el Bot despliegue un mensaje con las instrucciones necesarias para que el usuario ingrese su consejo, tras haber recibido el consejo del usuario, el Bot desplegará los mensajes correspondientes según el flujo lógico que se estableció. Este proceso se muestra en la siguiente captura de pantalla.

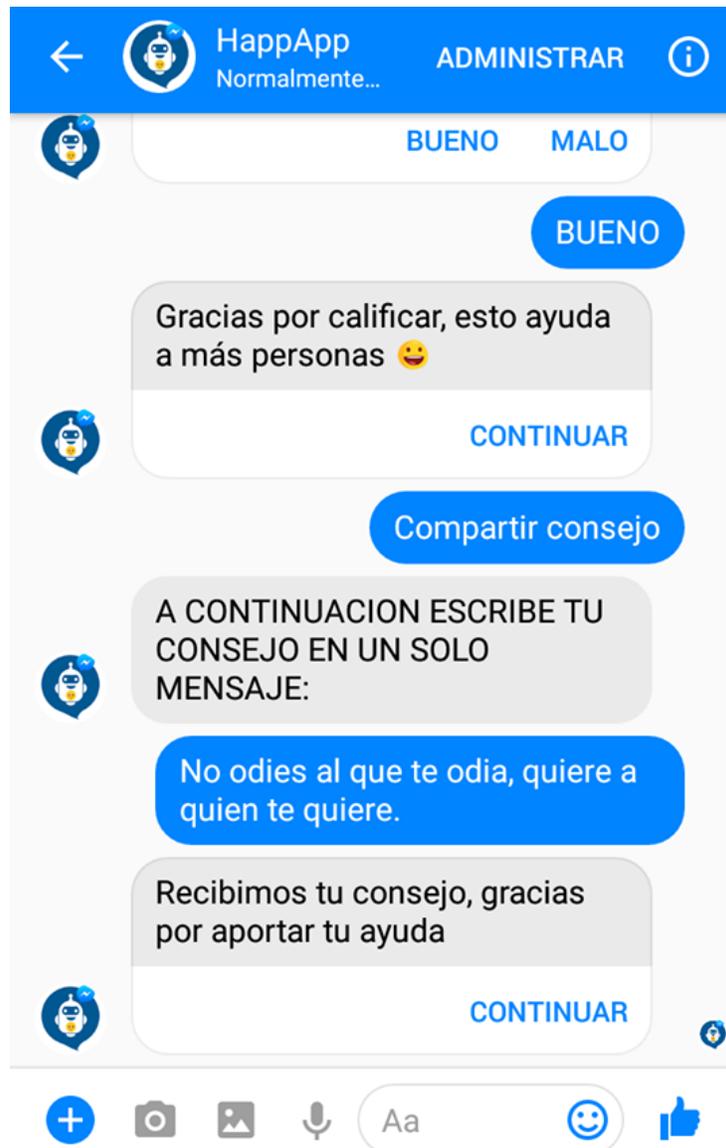


Figura 6.10 Interacción con el Bot en Messenger.

Cuando el usuario pulsa la opción de “Conocer a alguien”, el Bot despliega en pantalla la información necesaria de otro usuario para que se puedan contactar y un mensaje con el botón de “Continuar”.



Figura 6.11 Interacción con el Bot en Messenger.

Como se puede apreciar en la captura de pantalla anterior, el Bot despliega dentro de una plantilla la fotografía del perfil de Facebook de otro usuario, así como su nombre de perfil de Facebook, estos datos se recolectaron de los usuarios que interactúan con el Bot y se registraron a la base de datos.

Cuando el usuario pulsa la opción de “Ayuda especializada”, el Bot desplegará la información de un centro especializado para tratar cuadros de depresión y asuntos psicológicos. Esta información podrá cambiarse en cualquier momento por el desarrollador de la aplicación. En la siguiente captura de pantalla se demuestra el resultado de esta interacción.

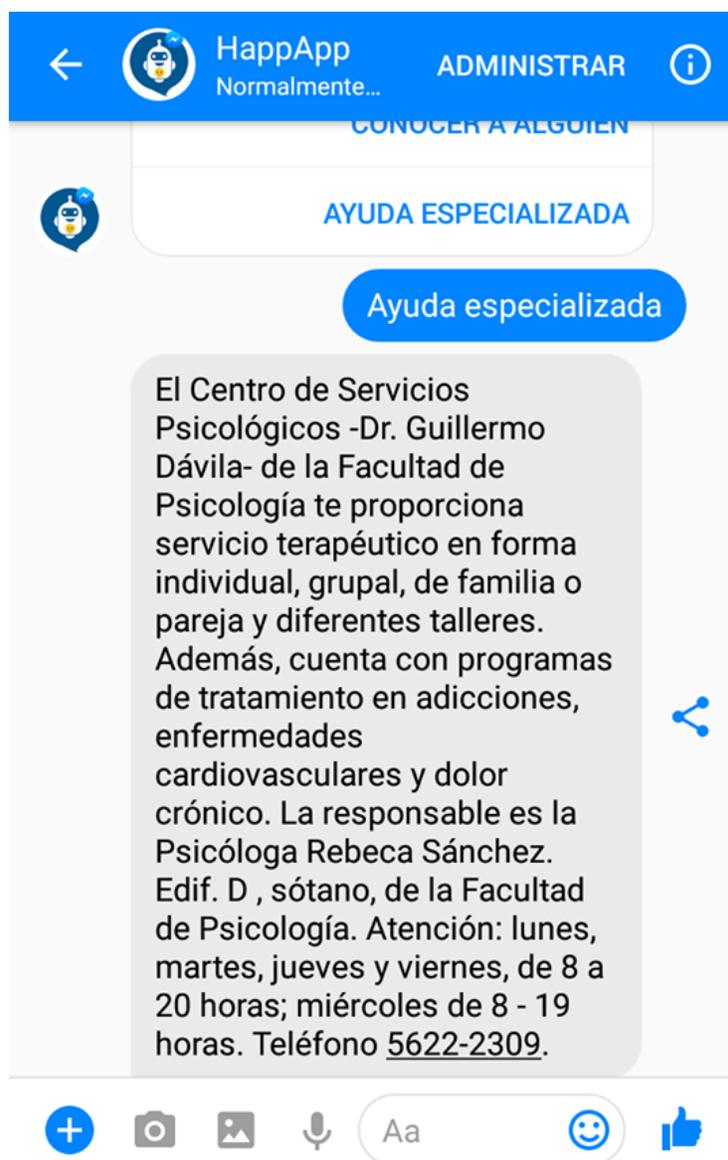


Figura 6.12 Interacción con el Bot en Messenger.

Cuando el Bot termina una interacción de las opciones especiales con éxito despliega un mensaje correspondiente y de la opción de “Continuar” mediante un botón, cuando el usuario presiona este botón el Bot despliega en pantalla un mensaje con recomendaciones para que el usuario utilice la aplicación móvil de Android, también junto con este mensaje se desplegará el botón “Empezar” que al

ser presionada volverá a repetir el ciclo para las interacciones, como se demostró al inicio de la conversación con él Bot. Esto se demuestra en la siguiente captura de pantalla.

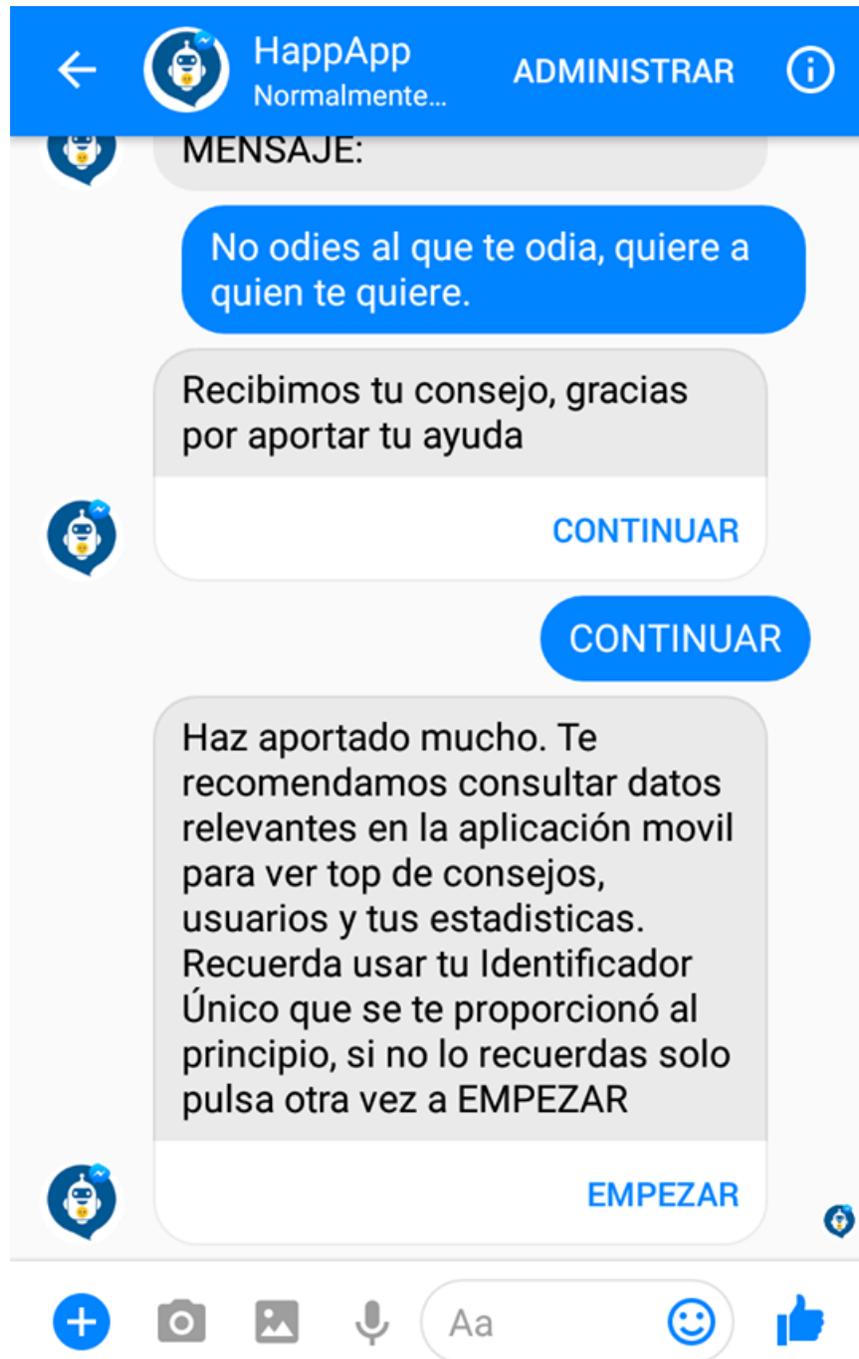


Figura 6.13 Interacción con el Bot en Messenger.

Otra de las interacciones posibles es que el usuario escriba mensajes hacia el Bot sin utilizar las interacciones establecidas, en este caso el Bot desplegará un mensaje invitando al usuario a interactuar con el Bot mediante las interacciones establecidas y el botón “Empezar”. La siguiente captura de pantalla demuestra esta interacción.

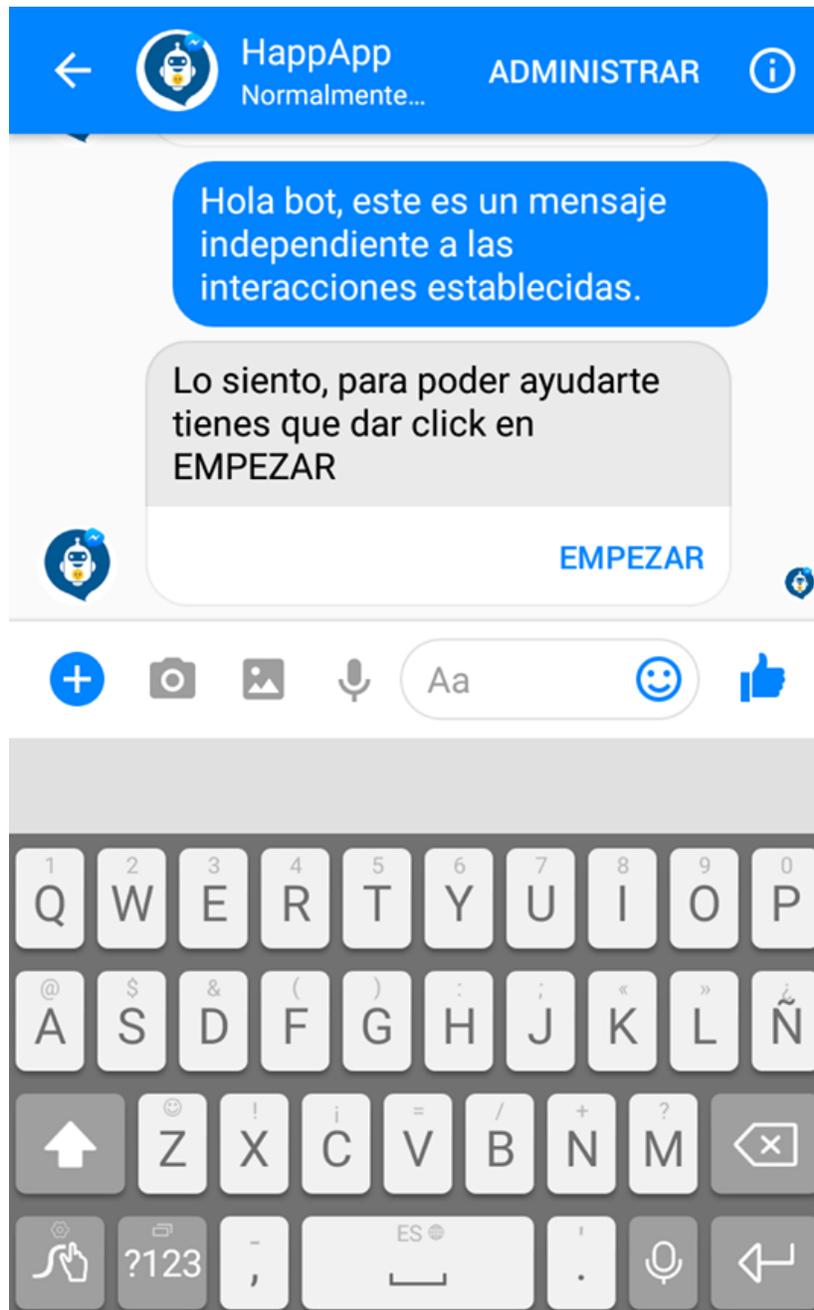


Figura 6.14 Interacción con el Bot en Messenger.

6.2 PRUEBAS CON LA BASE DE DATOS.

Para verificar el correcto funcionamiento de la base de datos, se comprueba que la base de datos se actualice tras realizar interacciones con el Bot que activen acciones de escritura a la base de datos. La base de datos se puede visualizar en la nube a través de la consola que proporciona Firebase.

Las siguientes capturas de pantalla demuestran la correcta implementación de la base de datos con el sistema.

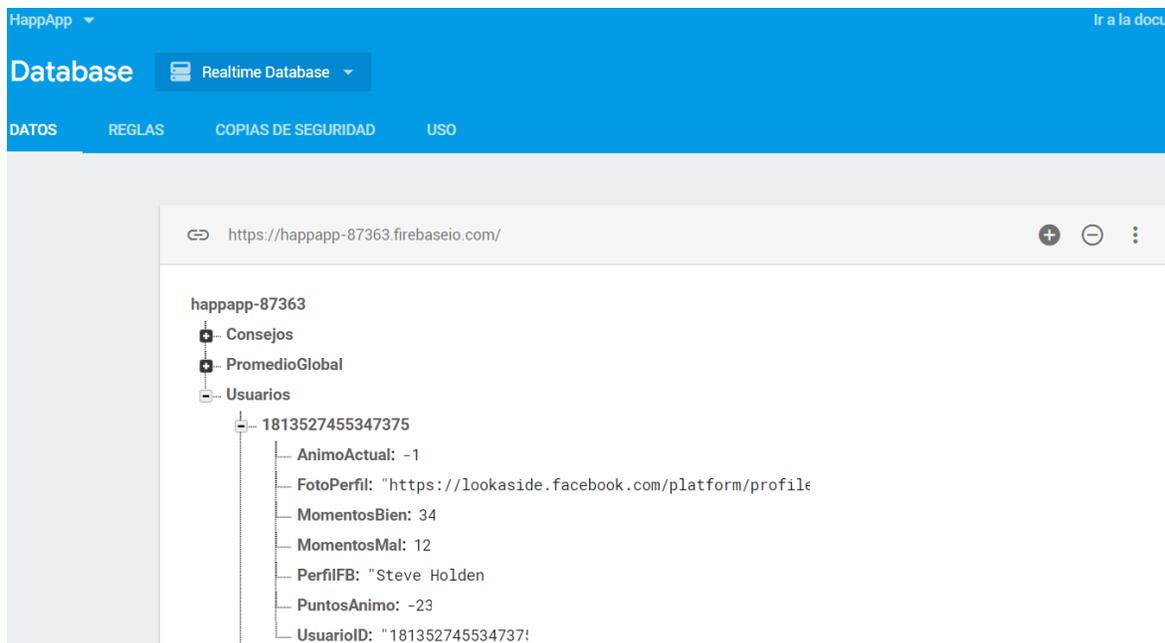


Figura 6.15 Inserción de usuarios.

La captura de pantalla anterior demuestra que los datos del usuario han sido correctamente insertados o actualizados a la base de datos. Se muestra que cada usuario es identificado por el Identificador Único que se le asignó, convirtiéndose de este modo en un nuevo nodo del árbol. También podemos ver que en la base de datos se conserva la estructuración definida para los usuarios, que se compone por datos de tipo cadena y numéricos. Los tipos de datos almacenados para cada usuario son:

- “AnimoActual”
- “FotoPerfil”
- “MomentosBien”
- “MomentosMal”
- “PerfilFB”
- “PuntosAnimo”
- “UsuarioID”

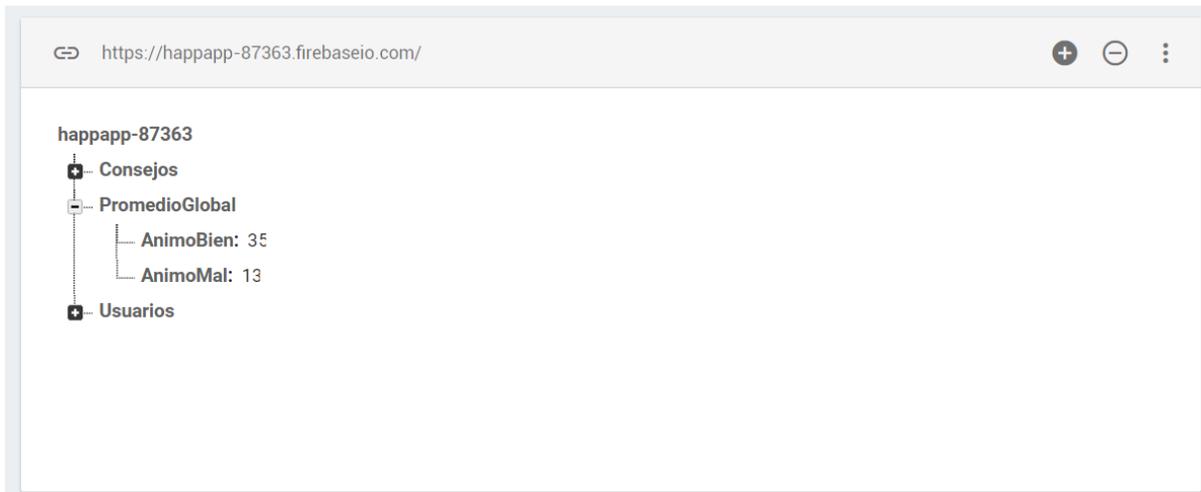


Figura 6.16 Inserción de datos globales.

La captura de pantalla anterior demuestra que las escrituras y actualizaciones que se realizan a la base de datos dentro del nodo de “PromedioGlobal” se realizan de forma satisfactoria e íntegra conservando así la estructura previamente descrita para este nodo. Los datos que se almacenan dentro de este nodo son de tipo numérico y corresponden a:

- “AnimoBien”
- “AnimoMal”

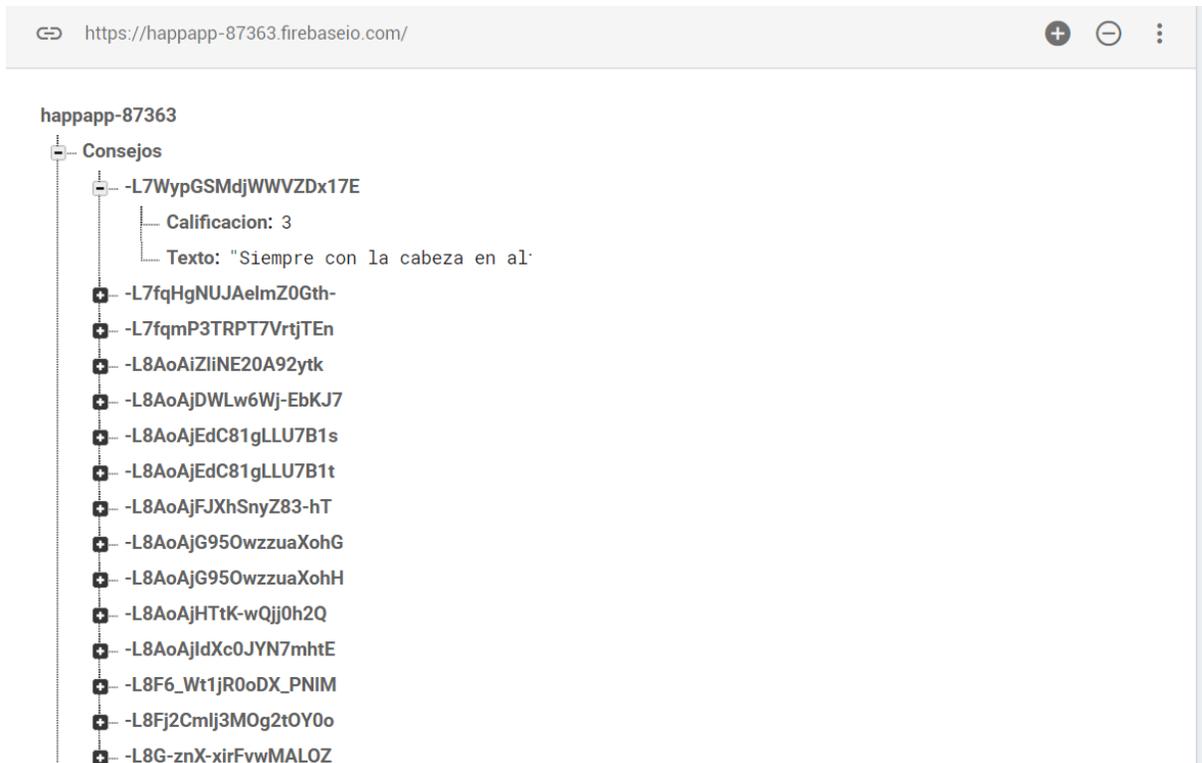


Figura 6.17 Inserción de consejos.

La captura de pantalla anterior demuestra que las inserciones y actualizaciones en el nodo de los consejos se realizan de manera adecuada, conservando la estructura definida para este nodo. Por otro lado, cabe mencionar que en este caso es una función de Firebase la encargada de escribir las llaves que identifican a cada consejo dentro del nodo "Consejos", estas son llaves aleatorias que se generan cada que un nuevo consejo es agregado y garantizan la no repetición de llaves dentro de la base de datos. Cada uno de los consejos cuenta con los siguientes datos dentro de su nodo:

- Calificación
- Texto

Adicionalmente se anexa una gráfica generada por Firebase que representa el uso de los datos de la base de datos. Esta cualidad de Firebase brinda la opción de consultar conexiones a la base que se encuentran activas, el almacenamiento utilizado y el disponible, el total de datos descargados, así como los cargados.



Figura 6.18 Gráficas Firebase.

6.3 PRUEBAS CON LA APLICACIÓN MÓVIL ANDROID.

Se debe garantizar el correcto funcionamiento de la aplicación móvil, para esto se realizan las pruebas pertinentes para verificar que la programación de la aplicación se ejecuta correctamente. Para realizar las pruebas se utilizó un teléfono inteligente de la marca Huawei del modelo GW el cual cuenta con las siguientes características:

Sistema operativo (SO)	Huawei GW
Sistema operativo	<u>Android 6.0.1 Marshmallow</u>
Hardware y memoria	Huawei GW
Arquitectura - Chipset	64bits - HiSilicon Kirin 620 (28nm)
CPU Procesador / Núcleos	1.2Ghz Octa-Core ARM Cortex-A53
GPU tarjeta gráfica	ARM Mali-450 MP4 533Mhz
Memoria RAM	2GB LPDDR3
Memoria interna	16GB (10GB accesible al usuario)
Pantalla	Huawei GW
Tipo de pantalla	TFT LCD IPS
Tamaño de la pantalla	5.5" pulgadas
Proporción de la pantalla	~70,1% (screen-to-body ratio)
Touchscreen	Capacitiva Multitouch
Resolución de la pantalla	720 x 1280 píxeles
Densidad (píxeles por pulgada)	267 PPI
Colores	16 millones

Figura 6.19 Características del dispositivo móvil de pruebas [4].

El dispositivo móvil cuenta con las características necesarias para ejecutar la aplicación de manera eficiente.

Para poder probar la aplicación, esta será instalada en el dispositivo a través de Android Studio el cual permite la compilación dentro del entorno de desarrollo, así como su instalación directa a un dispositivo conectado.

Cuando el proyecto de programación de la aplicación compila sin errores y satisfactoriamente, Android Studio genera el APK con todos los componentes necesarios para realizar la instalación directa en un dispositivo móvil con sistema operativo Android.

La siguiente captura de pantalla demuestra que la aplicación ha sido correctamente instalada dentro del dispositivo de pruebas.

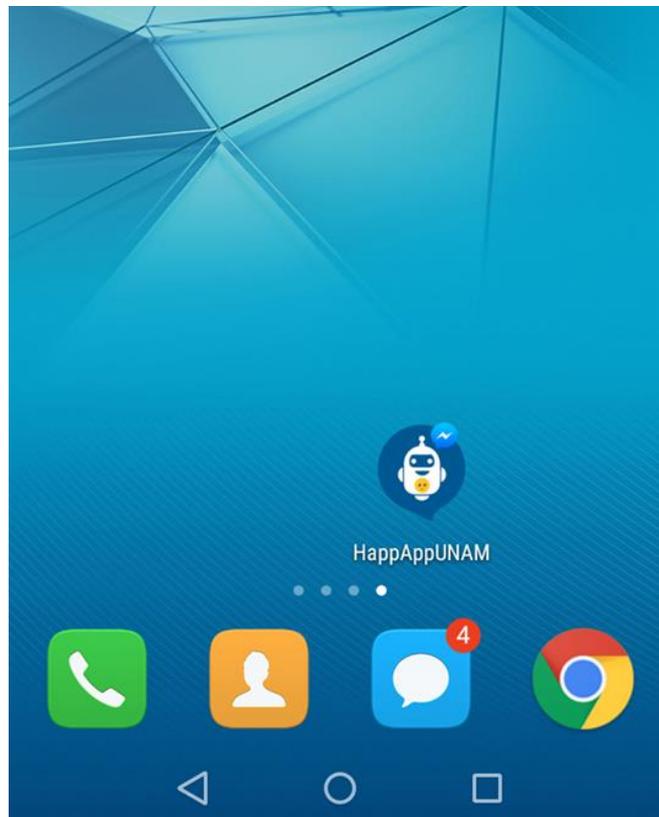


Figura 6.20 Icono de la aplicación móvil.

Como se puede apreciar en la captura de pantalla anterior, la aplicación se instaló correctamente ya que podemos visualizar el icono de la aplicación en la pantalla del dispositivo de prueba.

Posteriormente se prueba la navegación a través de la aplicación, para esto se pulsa el icono de la aplicación dentro del dispositivo de prueba para iniciar la ejecución de la aplicación.

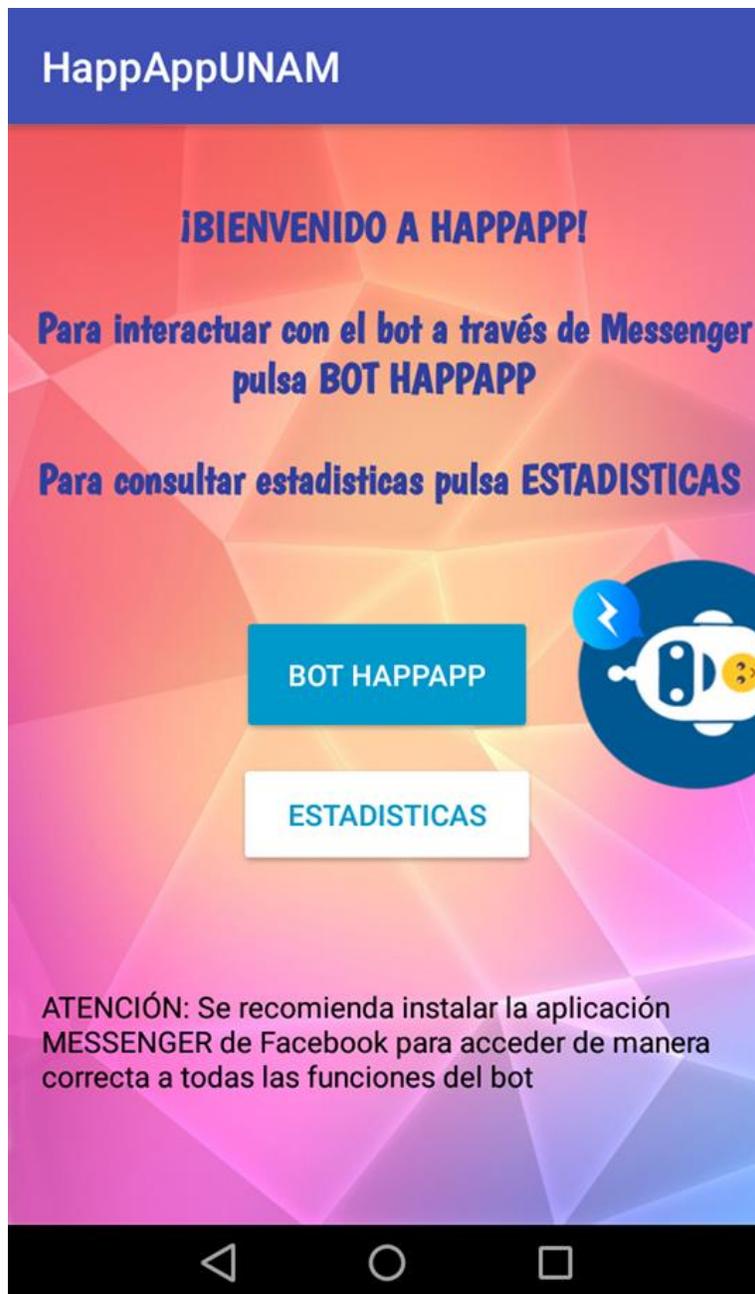


Figura 6.21 Aplicación Móvil Android.

La anterior captura de pantalla muestra la primera pantalla que se visualiza al iniciar la aplicación móvil dentro del dispositivo de pruebas. Como se puede apreciar la aplicación cumple con las características descritas dentro de la fase de elaboración. En pantalla se puede ver un menú con dos botones, así como las instrucciones necesarias para indicar la navegación a través de la aplicación.

Al presionar el botón “BOT HAPPAPP”, la aplicación redirecciona al usuario a la conversación con el Bot a través de la plataforma de Facebook Messenger. Cuando este botón se presiona por primera vez en pantalla se mostrará un aviso en el cual

se le pregunta al usuario con qué aplicación desea completar la acción solicitada, en este caso específico se recomienda utilizar la aplicación oficial de Facebook Messenger (Figura 6.22).

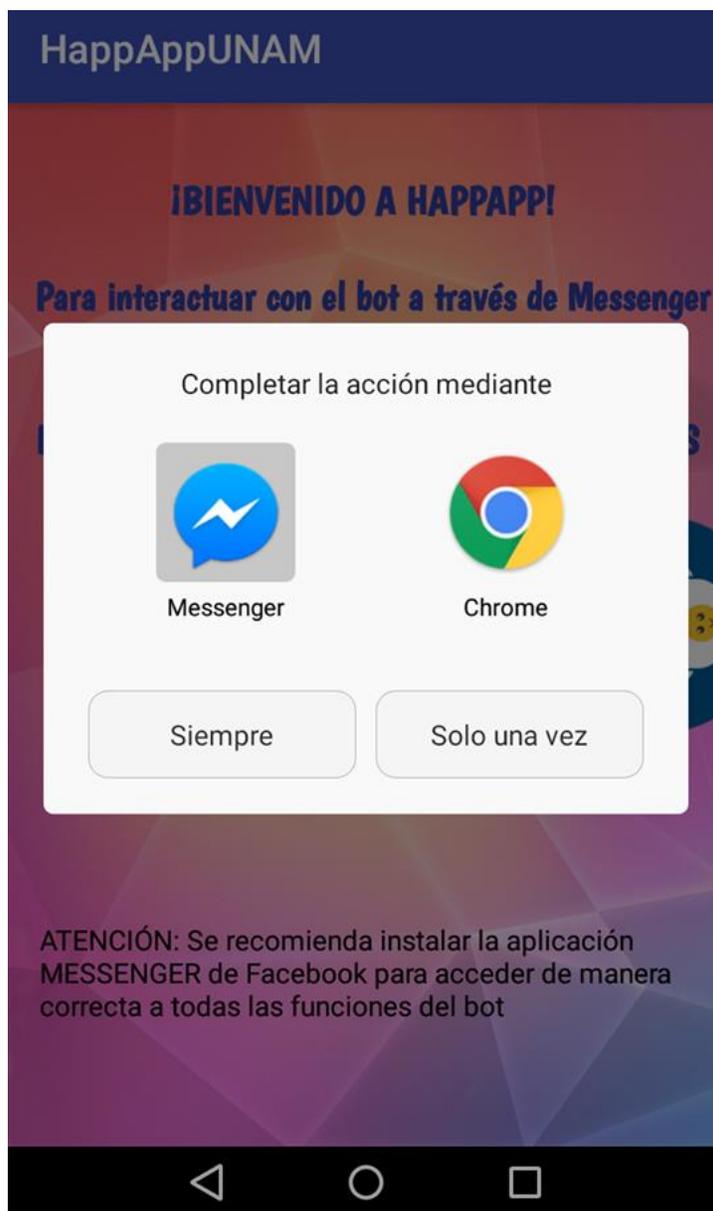


Figura 6.22 Aplicación Móvil Android.

Por otro lado, cuando el usuario presiona el botón “ESTADISTICAS”, se iniciará otra actividad dentro de la aplicación móvil la cual despliega en pantalla un menú con botones mediante el cual se podrá navegar a través de las funciones que ofrece la aplicación. La siguiente captura de pantalla muestra el menú descrito anteriormente.



Figura 6.23 Aplicación Móvil Android.

Como se puede apreciar el menú de estadísticas brinda dos opciones de consulta, la información global y las estadísticas por usuario, las cuales como ya se había descrito anteriormente dirigirán al usuario a las actividades correspondientes para cada caso.

Cuando el usuario presiona el botón “ESTADISTICAS POR USUARIO”, la aplicación procederá a desplegar en pantalla un mensaje con las instrucciones necesarias para consultar estadísticas personales, para lo cual se utiliza un identificador único el cual es colocado en una caja de texto para después ser enviado mediante la acción de un botón. La siguiente captura de pantalla demuestra lo descrito anteriormente.

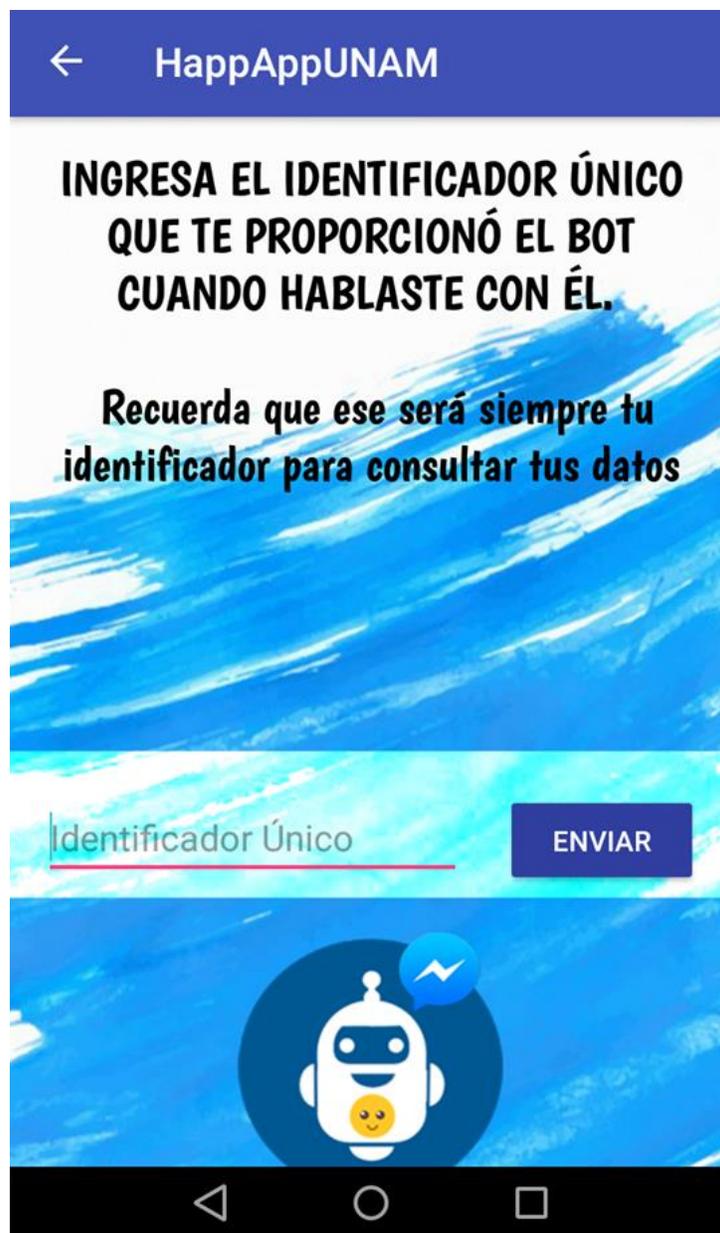


Figura 6.24 Aplicación Móvil Android.

Cuando el usuario ingresa un identificador válido la aplicación móvil deberá acceder a la base de datos para consultar los datos del usuario al que corresponde el identificador, si la consulta es exitosa se desplegará en pantalla la información útil para el usuario representada en forma gráfica circular, también se mostrarán las acotaciones necesarias, así como otros datos informativos de relevancia como el ánimo actual o el índice de ánimo. Para realizar esta prueba dentro de la aplicación móvil se ha insertado a la base de datos la información necesaria para que se ejecuten correctamente las funciones de la aplicación. La siguiente captura de pantalla demuestra el funcionamiento descrito anteriormente y que este se ha llevado a cabo correctamente.

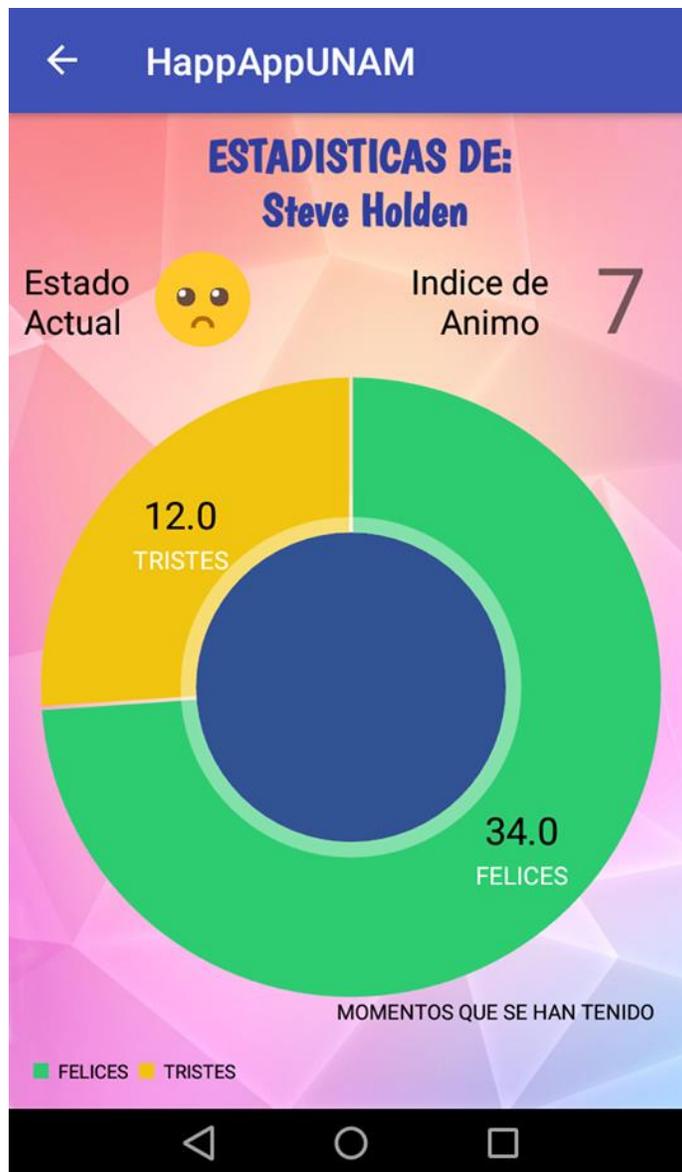


Figura 6.25 Aplicación Móvil Android.

Si el usuario decide consultar las opciones de información global presionando el botón "INFORMACIÓN GLOBAL" se desplegará en pantalla un menú con tres botones brindando al usuario la posibilidad de consultar estadísticas globales, top de usuarios y top de consejos. La siguiente captura de pantalla muestra el menú descrito anteriormente.



Figura 6.26 Aplicación Móvil Android.

Cuando el usuario selecciona la opción de consultar “ESTADÍSTICAS GLOBALES”, al igual que cuando se consultan las estadísticas personales se desplegará en pantalla la información útil para el usuario correspondiente a la información global registrada por toda la comunidad de usuarios representada en forma gráfica circular, también se mostrarán las acotaciones necesarias, así como otros datos informativos de relevancia como el ánimo actual o el índice de ánimo. Para realizar esta prueba se realizaron inserciones de usuarios ficticios a la base de datos con la finalidad de generar información suficiente para que las pruebas fueran más significativas. La siguiente captura de pantalla muestra lo descrito anteriormente.

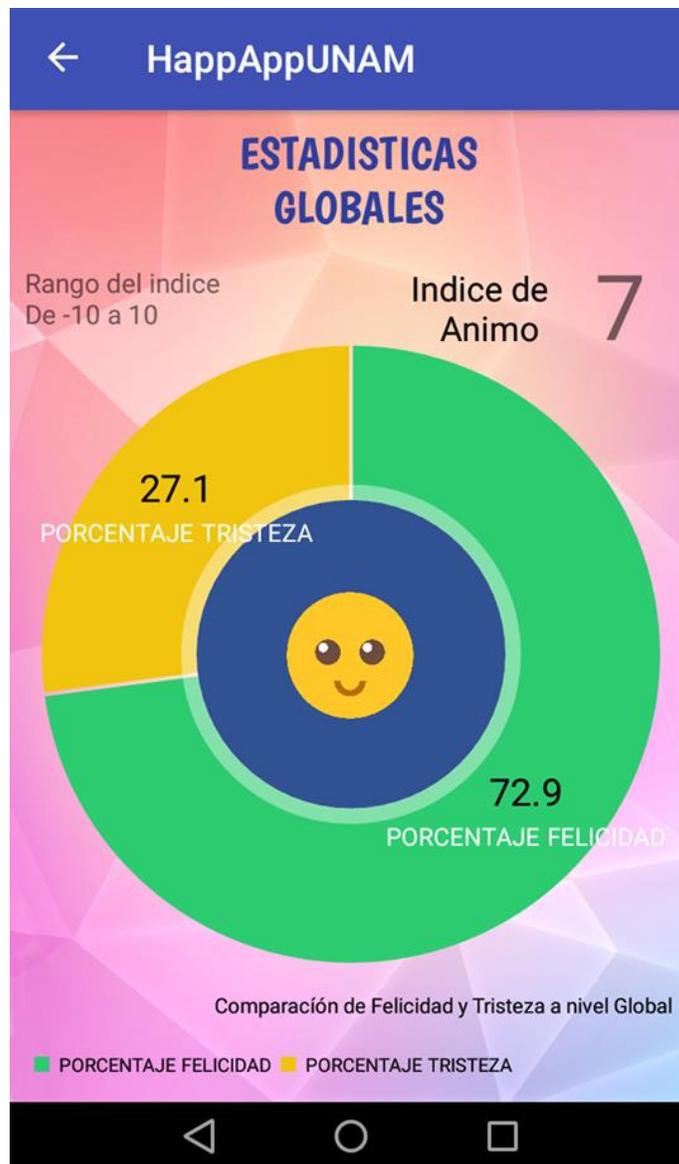


Figura 6.27 Aplicación Móvil Android.

Si el usuario selecciona consultar el “TOP 10 USUARIOS” se desplegará en pantalla una lista de diez usuarios que son seleccionados en base a los puntos de felicidad que se generan tras la interacción con el usuario. Esta lista muestra información de cada usuario como su nombre, su puntaje y su identificador único, esto para que otros usuarios puedan ver sus estadísticas y las interpreten de la manera que sea más conveniente. Para realizar esta prueba se ha ingresado a la base de datos la información necesaria de múltiples usuarios para poder probar el correcto funcionamiento de la aplicación. La siguiente captura de pantalla demuestra la correcta implementación del listado para el top diez de usuarios.



Figura 6.28 Aplicación Móvil Android.

Como se puede ver en la captura de pantalla anterior, el top de usuarios se encuentra en una lista en la cual para poder visualizar a todos los usuarios dentro de la lista permite el desplazamiento hacia abajo con funciones de pantalla táctil.

Cuando el usuario selecciona la opción de "TOP 10 CONSEJOS" se consultará los diez consejos mejor calificados por los usuarios dentro de la base de datos. Del mismo modo que con el top de usuarios, en pantalla se desplegarán a manera de lista los diez consejos consultados, los cuales adicionalmente contarán con una barra indicadora de estrellas la cuál servirá para realizar comparativas entre cada consejo. Para realizar esta prueba se han ingresado a la base de datos múltiples consejos con diferentes calificaciones. La siguiente captura de pantalla demuestra lo descrito anteriormente.

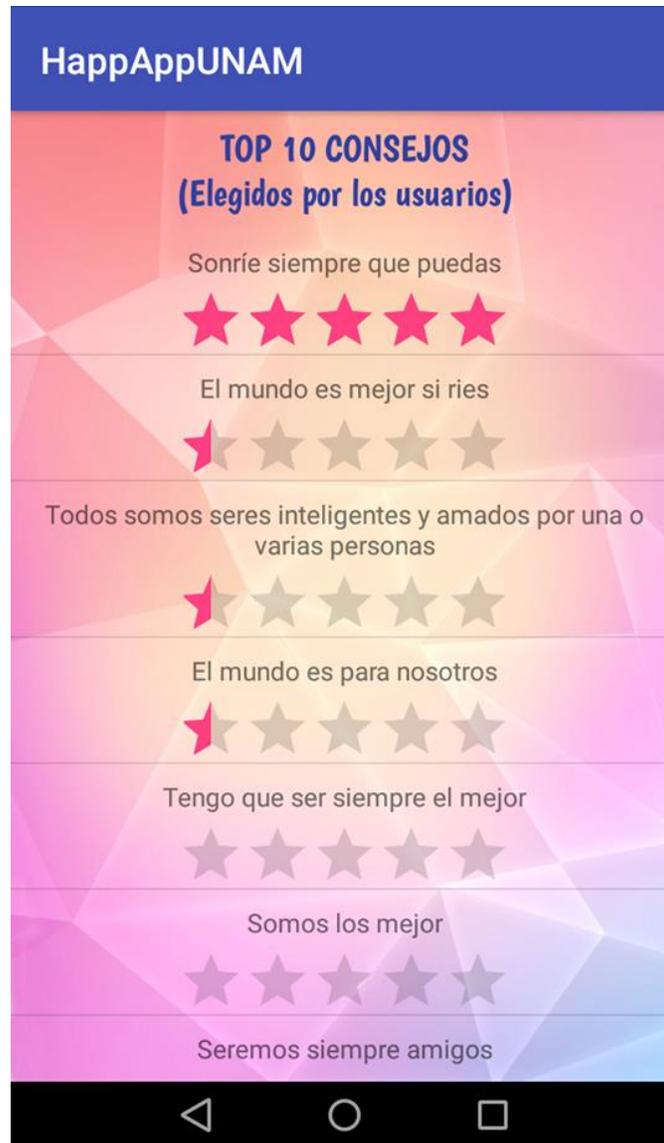


Figura 6.29 Aplicación Móvil Android.

Como se puede apreciar en la captura de pantalla anterior, las estrellas representan que tan bueno es un consejo en comparación de otros, esto se logra con base a las calificaciones proporcionadas por los usuarios a través de la interacción con el Bot.

6.3.1 VISTAS HORIZONTALES.

Cuando un usuario tiene activa la opción de rotación automática en su dispositivo móvil y coloca su dispositivo en posición horizontal las aplicaciones tienen la opción de adaptar su diseño para que puedan ser visualizadas de una forma horizontal. La aplicación desarrollada en Android Studio en este caso también permitirá la visualización de su contenido de forma horizontal es por eso que se implementan las características necesarias para que esto sea posible. Las vistas horizontales de la aplicación se implementan con la finalidad de brindarle al usuario una experiencia más satisfactoria dentro de la aplicación y que si así lo desea pueda navegar a través de todas las funciones de un modo horizontal.

A continuación, se anexan todas las vistas horizontales que se probaron dentro del dispositivo de pruebas, las cuales cumplen con su finalidad de adaptación.



Figura 6.30 Navegación horizontal.



Figura 6.31 Navegación horizontal.

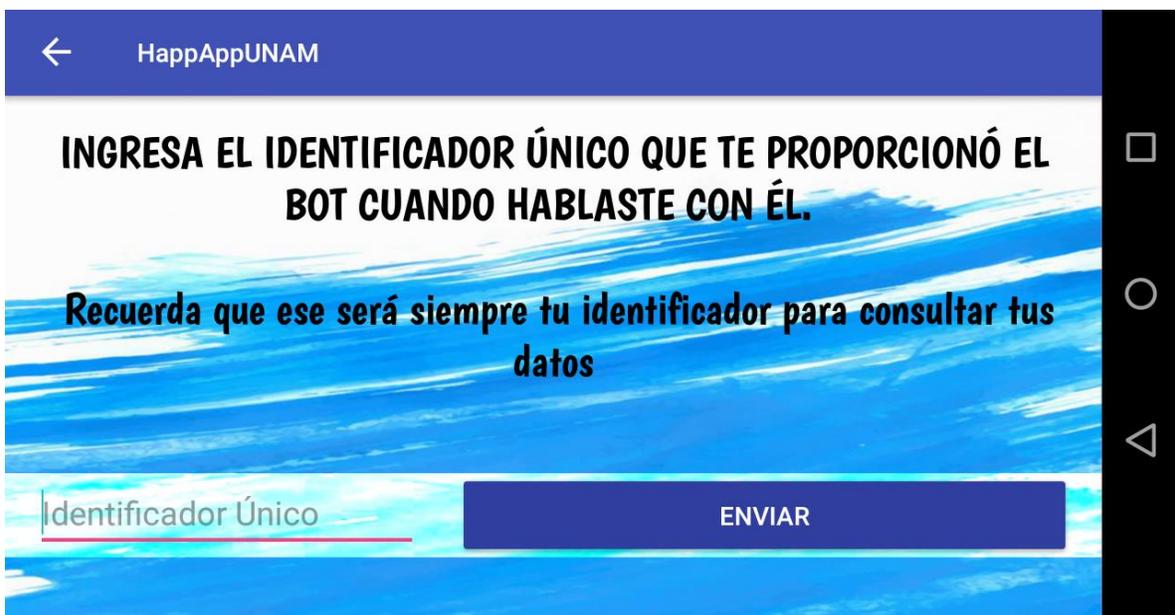


Figura 6.32 Navegación horizontal.



Figura 6.33 Navegación horizontal.



Figura 6.34 Navegación horizontal.



Figura 6.35 Navegación horizontal.

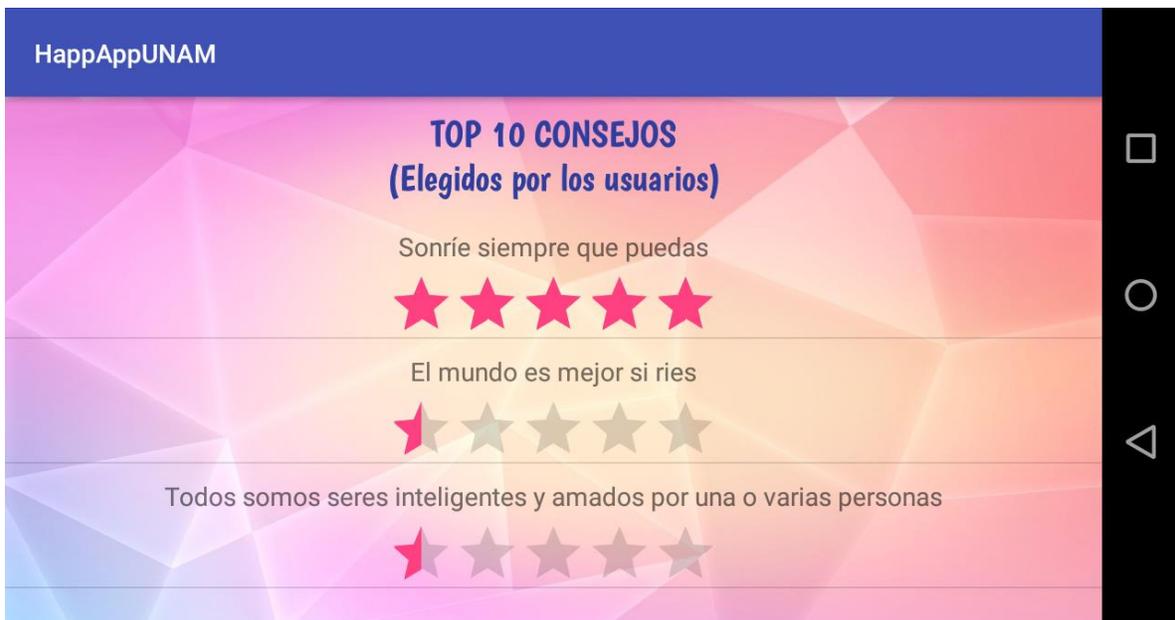


Figura 6.36 Navegación horizontal.



Figura 6.37 Navegación horizontal.

Como se puede apreciar en las capturas de pantalla, las vistas horizontales cumplen con su objetivo al proporcionar al usuario una navegación de forma horizontal, todos los elementos en cada pantalla se reestructuran para adaptarse de la manera más cómoda a la posición horizontal.

CAPÍTULO 7

RESULTADOS

CAPÍTULO 7. RESULTADOS

7.1 RESULTADOS DEL BOT.

El Bot cumplió satisfactoriamente con todos los requerimientos que se establecieron en el principio del proyecto, tras realizar las pruebas pertinentes con la interacción de diversos usuarios el Bot presenta una funcionalidad óptima dentro de los parámetros esperados, ya que responde a las interacciones con el usuario de manera inmediata, brindando de este modo una experiencia más fluida y útil para el usuario.

Con las pruebas realizadas se verificó el correcto funcionamiento del Bot lo cual permite su implementación en un medio de producción real. Para lograr esto el Bot pasó por una fase de transición para llevarlo de la etapa de desarrollo a una etapa de producción. Tras estar la fase de desarrollo en la cual se implementaba su ejecución dentro de una laptop que fungía como servidor HTTP mediante el uso del software Ngrok, el Bot fue migrado a una plataforma en la nube que soporta lenguaje Node.js y brinda la capacidad de correr una aplicación dentro de contenedores virtuales, brindando protocolos HTTP y SSL asociados a la página de Heroku. La plataforma de Heroku brinda ciertos servicios de manera gratuita, de manera que la aplicación del Bot aprovechará este beneficio para almacenar su programación dentro de la plataforma que será la encargada de correrla en sus servidores.

Para poder hacer uso de la plataforma de Heroku y todas sus características se creó una cuenta de Heroku y se instalaron las dependencias necesarias para su uso. El Bot fue implementado en la plataforma de Heroku, a la cual se subieron los archivos necesarios para que la ejecución se realizara de forma correcta en la nube. La siguiente impresión de pantalla demuestra la correcta implementación del Bot en la plataforma oficial de Heroku.

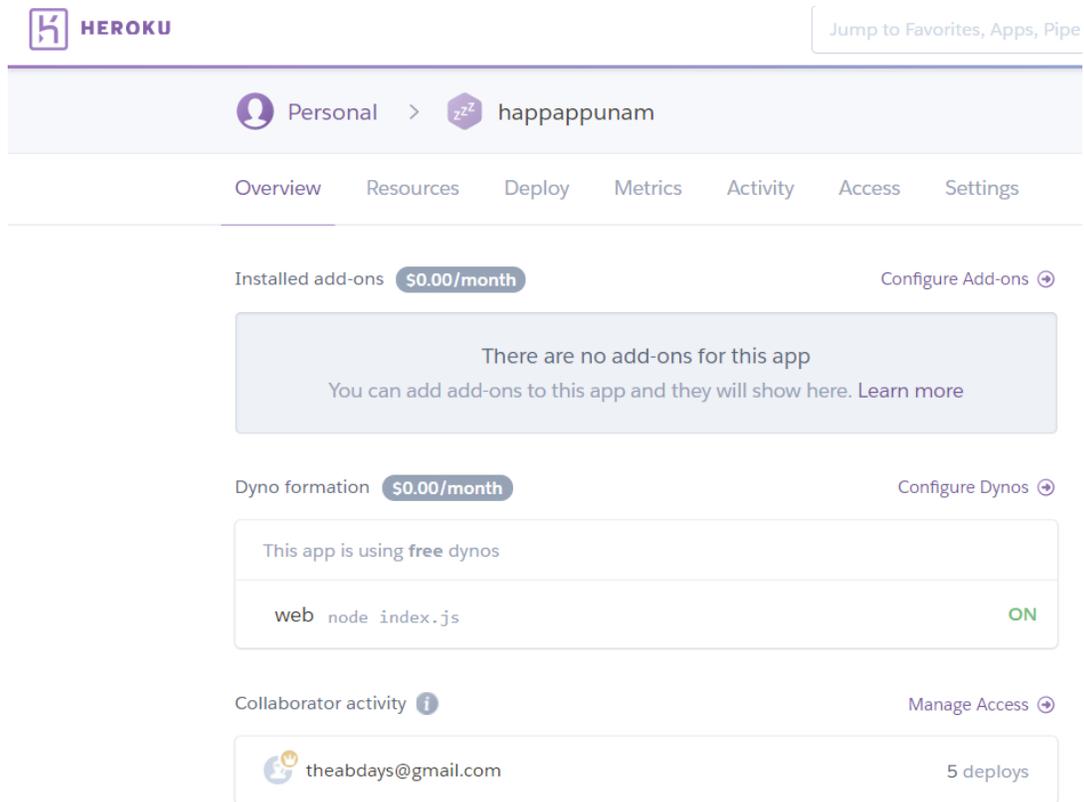


Figura 7.1 Implementación en plataforma Heroku.

Tras la implementación del Bot en la plataforma de Heroku, fue necesario cambiar la suscripción del “webhook” asociado a la aplicación de Facebook en la plataforma oficial de Facebook para desarrolladores. La siguiente imagen representa el cambio realizado correspondiente a la suscripción del “webhook”.

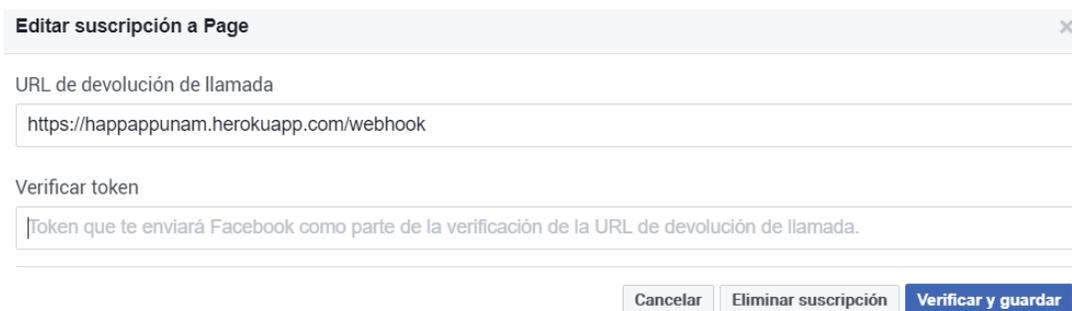


Figura 7.2 Suscripción del webhook a Heroku.

Como se puede apreciar en la imagen anterior, la URL de devolución de llamada corresponde a una proporcionada por la plataforma de Heroku que se encuentra asociada a la implementación del Bot dentro de la propia plataforma.

Después de realizar la implementación en la plataforma de Heroku se probó la funcionalidad del Bota través de las interacciones con el usuario. El Bot presentó

una funcionabilidad óptima con el añadido de que ahora no es necesario correr la programación del Bot en un servidor virtual cada que se quiera interactuar con él, sino que el Bot se encuentra en constante ejecución dentro de la plataforma de Heroku, esto permite la interacción del Bot con el usuario a cualquier hora del día, dotando así a la aplicación de una alta disponibilidad.

7.1.1 RETROALIMENTACIÓN DEL USUARIO.

Durante la fase de pruebas, usuarios de la aplicación realizaron observaciones con respecto a las interacciones que realizaron con el Bot las cuales se tomaron en cuenta para mejorar el funcionamiento del Bot. La retroalimentación proporcionada por los usuarios se ve reflejada en la siguiente tabla.

Componente del BOT.	Observaciones y comentarios.	Entidad.
Inicio de la conversación.	Se sugiere que el Bot de una bienvenida corta y sea concreto con las instrucciones a realizar.	Usuario.
Botones.	Se sugiere que los botones tengan texto corto y sean llamativos para el usuario.	Usuario.
Mensajes.	Se sugiere que los mensajes desplegados por el Bot sean cortos y en su estructura hagan el uso de “emojis” para hacer la conversación más amena.	Usuario.
Navegación.	Se sugiere que la navegación en el Bot se pueda realizar al antojo del usuario. Es decir que el usuario pueda navegar hacia atrás en la conversación para retomar una interacción pasada, así será más fácil utilizar las funciones del Bot sin la necesidad de tener que hacer el recorrido lógico completo de la programación.	Usuario.
Vinculación.	Se sugiere que el Bot proporcione la imagen actualizada del perfil del usuario dentro de	Usuario.

	Facebook para que encontrarlo en la red social sea más sencillo	
Lenguaje.	Se sugiere que el Bot utilice un lenguaje familiar o coloquial para comunicarse con el usuario, de este modo el usuario podrá sentirse con una mayor confianza al interactuar con el usuario.	Usuario.

Tabla 7.1 Retroalimentación para el Bot.

Todas las recomendaciones hechas por los usuarios fueron tomadas en cuenta para el desarrollo final del Bot. El cuál al final cuenta con la aprobación de los usuarios y cumple con su entera satisfacción.

7.2 RESULTADOS DE LA APLICACIÓN MÓVIL.

La aplicación móvil en general cumple con todos los requerimientos establecidos en el análisis del proyecto, adicionalmente presenta un acoplamiento exitoso con los otros componentes del sistema los cuales son la base de datos y el Bot.

La aplicación móvil fue instalada en diferentes dispositivos con sistema operativo Android. Se probó con versiones del sistema operativo anteriores a la versión 6.0 Marshmallow y con versiones posteriores, presentando en ambos casos un rendimiento similar sin presentar fallos o contrariedades.

Tras presentar la aplicación con algunos usuarios en la etapa de desarrollo, estos se encargaron de proporcionar ciertas observaciones y recomendaciones para mejorar la aplicación móvil.

La siguiente tabla representa la retroalimentación proporcionada por los usuarios tras haber hecho uso de la aplicación móvil.

Componente de la aplicación móvil.	Observaciones y comentarios.	Entidad.
Icono.	Se sugiere que el icono de la aplicación sea llamativo para que por medio de este se convenza al usuario de utilizar la aplicación con mayor frecuencia.	Usuario.
Menús.	Se sugiere que los botones utilizados para los menús sean llamativos para el usuario, para que este se vea motivado a acceder a todas las opciones que brinda el menú.	Usuario.
Diseño.	Se sugiere que el diseño de la aplicación en general cuente con colores e imágenes llamativas para que el usuario se sienta entretenido al navegar a través de la aplicación.	Usuario.

Gráficas.	Se sugiere que las gráficas cuenten con alguna animación o funcionalidad interactiva para que el usuario se sienta más atraído a consultar la información.	Usuario.
Tops.	Se sugiere que los tops que contiene la aplicación sean llamativos y no solo letras para que sea más atractivo para el usuario consultarlos.	Usuario.
Navegación.	Se sugiere que la navegación a través de la aplicación se pueda realizar tanto de manera vertical como de manera horizontal. Esto le brindará cierta flexibilidad de uso al usuario	Usuario.

Tabla 7.2 Retroalimentación para la aplicación móvil.

Las sugerencias realizadas por los usuarios fueron consideradas para el desarrollo definitivo de la aplicación implementando de este modo nuevas adiciones. Tras la interacción de los usuarios con la fase final de la aplicación estos expresaron su plena satisfacción y aprobación para la aplicación móvil.

7.3 RESULTADOS DEL USUARIO.

Para verificar el funcionamiento de todos los componentes finales del sistema, se solicitó a diferentes usuarios que hicieran uso de este, con la finalidad de que expresaran comentarios tras haberlo utilizado de este modo poder determinar si el sistema cumple con los objetivos generales planteados en un principio.

Después de haberlo utilizado los usuarios en general expresaron una completa aceptación para el sistema y corroboran que su funcionamiento cumple con los objetivos que se han presentado.

La siguiente tabla representa un condensado de todas las opiniones dadas por los usuarios con respecto a diferentes cuestionamientos realizados sobre los diversos aspectos de la aplicación y sus objetivos.

Aspecto.	Resumen de opiniones.
Estado de ánimo después de utilizar el sistema.	Un estado de ánimo favorable, el cual mejoró con respecto al estado de ánimo que se tenía antes de interactuar con la aplicación, el cual se le atribuye tanto a la conversación que se mantuvo con el Bot el cuál proporciona los consejos, y también a la aplicación donde se pudieron consultar aún más consejos.
Interacción con el Bot.	El Bot resulta ser amistoso con los usuarios, no es invasivo y muy explicativo conforme a las actividades que puede realizar. Gusta demasiado el hecho de que pueda interactuarse con él a través de la red social de Facebook.
Uso de la aplicación móvil.	La aplicación es muy ligera y ocupa poco espacio dentro del dispositivo móvil. Es fácil de utilizar y muy llamativa. La información que proporciona es fácil de entender y representa datos muy útiles que podrían utilizarse para ir mejorando las actividades que uno realiza a fin de mejorar sus estadísticas dentro de la aplicación. Adicionalmente sirve como un registro para ir viendo como mejora la situación sentimental del usuario.

<p>Generalidades.</p>	<ul style="list-style-type: none"> • Gusta mucho que los consejos que se pueden consultar mediante el Bot o dentro de la aplicación móvil pueden tener cualquier extensión y no están restringidos de ningún modo. • Las gráficas, los elementos como emojis y estrellas dentro de la aplicación le brindan un estilo divertido a la aplicación. • El diseño de los fondos que se encuentran en la aplicación son muy alegres. • Las gráficas interactivas sirven para el estrés y son comparables con un “fidget spinner”.
------------------------------	---

Tabla 7.3 Retroalimentación para la aplicación móvil.

CAPÍTULO 8

CONCLUSIONES

CAPÍTULO 8. CONCLUSIONES

8.1 OBJETIVO PLANTEADO.

El sistema cumplió satisfactoriamente con todos los requerimientos establecidos en un principio. En conjunto la aplicación presenta una alta disponibilidad, lo cual garantiza su funcionamiento en cualquier momento que el usuario así lo requiera. Por otro lado, la aplicación maneja correctamente todos los errores que se puedan cometer durante un estado de producción real, lo cual vuelve al sistema a prueba de fallos.

Una de las consideraciones importantes a tener en cuenta es que el sistema esta enfocado en el uso de servicios proporcionados por plataformas que se encuentran en la nube, es decir que depende directamente del correcto funcionamiento de estos servicios, ya que si alguno de los servicios llega a presentar problemas internos esto se verá reflejado directamente en el funcionamiento del sistema. Sin embargo, cabe destacar que los servicios utilizados cuentan con el respaldo de empresas internacionales especializadas en desarrollo de software y nuevas tecnologías, es por esto que se puede contar con la alta fiabilidad de los servicios, ya que cada uno de estos es administrado por entidades especializadas además de que cuentan con servicios de mantenimiento profesionales que garantizan su correcto funcionamiento. Los servicios de los cuales es dependiente el sistema desarrollado se listan a continuación:

- Facebook
- Firebase
- Heroku

Se pueden consultar cualquiera de estos servicios, así como la información de las empresas que los gestionan con la finalidad de comprobar la fiabilidad de estos.

En general el objetivo general que se planteó al principio, el cual es poder combatir los efectos causados por la depresión, resulta ser cumplido tras la prueba con múltiples usuarios ya que estos aseguran que la aplicación les parece útil para manejar su estado de ánimo además de que la consideran divertida, lo cual va en pro de los objetivos planteados. Se espera que con una comunidad de usuarios grande y activa se puedan apreciar de mejor manera los beneficios que proporciona la aplicación móvil.

8.2 DESARROLLO.

La aplicación se desarrolló utilizando un esquema de desarrollo basado en la metodología RAD (Rapid Application Development), lo cual facilitó la organización del trabajo durante el desarrollo de la aplicación, mediante la prueba de diversos prototipos que se fueron mejorando continuamente se logró llegar a una versión final la cual cumplió satisfactoriamente con todos los requerimientos y objetivos planteados.

Es de mencionar que el trabajo realizado representa un reto de desarrollo ya que implementa tecnologías relativamente nuevas como lo son el desarrollo de Bots y la implementación de servicios enfocados en la nube. Así también la implementación de una base de datos horizontal y su estructuración nos permite colocar este desarrollo dentro del tipo de “redes sociales” ya que permite interacciones indirectas entre los usuarios del sistema, así como un crecimiento indefinido de la comunidad de usuarios lo cual se facilita mediante la base de datos de tipo no relacional.

El sistema cuenta con múltiples componentes como un Bot, base de datos no relacional y aplicación móvil para dispositivos con sistema operativo Android, tomando en consideración esto hay que destacar que se requieren amplios conocimientos de programación y desarrollo web, ya que se implementaron diferentes lenguajes en los que se incluyen Java, XML, JavaScript del lado del servidor, lenguaje nativo Firebase basado en estructuración JSON. Del mismo modo se manejó programación orientada a eventos la cual resulta ser poco común ya que la mayoría de los desarrollos se enfocan únicamente en programación orientada a objetos.

Finalmente podría sugerirse que este tipo de desarrollo es conveniente realizarlo con un equipo de trabajo de varios desarrolladores para guardar un mayor enfoque en cada uno de los componentes del sistema, incrementando así la productividad y acelerando procesos para realizar pruebas con mayor eficacia y constancia.

8.3 IMPLEMENTACIÓN.

Para liberar el sistema completamente funcional y que sea compatible con la interacción masiva de múltiples usuarios, será necesario adquirir planes de pago que ofrecen las plataformas utilizadas en el sistema con el fin de adquirir más almacenamiento disponible para la información que se pueda generar además de contar con mejor desempeño de los servicios.

Por otro lado, hay que mencionar que el Bot debe cumplir con las normas de uso de la plataforma con el fin de cumplir con características como protección de información personal y usos sin fines maliciosos.

Tras la implementación del sistema se puede pensar en un plan de marketing y difusión para lograr que la aplicación sea conocida por más usuarios. Haciendo resaltar los beneficios que proporciona la aplicación se puede lograr atraer a más usuarios e ir creciendo la comunidad de usuarios, con la finalidad de generar un impacto positivo y significativo en la sociedad en la que nos desarrollamos.

Los usuarios con los que se realizaron las pruebas aportaron comentarios positivos hacia la aplicación haciendo énfasis en su utilidad y los beneficios que puede proporcionar al mundo real. Del mismo modo declararon que el sistema es de una gran calidad y presenta tecnología innovadora.

8.4 INNOVACIÓN.

Tomando como referencia los comentarios aportados por los usuarios que han interactuado con la aplicación y todos los componentes que el sistema implementa se puede afirmar que es una solución innovadora para combatir los efectos causados por la depresión. El hecho de involucrar tecnología móvil enfocada a tratar un problema tan relevante como lo es este padecimiento que afecta a la sociedad a un nivel mundial representa un avance significativo en las capacidades que pueden brindar las tecnologías de la información en el mundo actual.

Los Bots al ser una tecnología relativamente nueva que no ha sido explotada por completo y que se encuentra en constante desarrollo e innovación, representan un área de oportunidad en la cual se pueden realizar avances significativos que contribuyan al desarrollo tecnológico de nuevos sistemas. Este sistema implementa el uso de un Bot con un enfoque que no se había utilizado anteriormente, siendo así que el sistema en sí representa ya un avance tecnológico.

Se pretende como trabajo a futuro que la aplicación pueda actualizarse con nuevos bloques de programación que proporcionen funcionalidades adicionales tanto al Bot como a la aplicación móvil. Este tipo de mejoras se implementarían tras realizar un análisis de los comentarios generados por una comunidad de usuarios amplia. Al mismo tiempo se podrán considerar nuevas funcionalidades que ofrezcan API's independientes las cuales se adaptarían al sistema y que representen un beneficio sustancial para los objetivos que se persiguen.

La innovación en general trae grandes beneficios al mundo ya que siempre está enfocada a mejorar una situación actual, esto se logra mediante realizar las cosas de un modo diferente aprovechando las herramientas tecnológicas con las que se cuenta. En conclusión, se espera que este desarrollo sea un paso adelante en la innovación de las tecnologías de la información y que sirva como base para futuras generaciones que busquen incursionar en la misma área de oportunidad.

BIBLIOGRAFÍA.

FUENTES

1. American Psychiatry Association (1994). DSM-IV. Manual diagnóstico y estadístico de los trastornos mentales. 3a. ed. Washington. 11 de Abril del 2018.
2. Psicomed. (2017). Clasificación Internacional de Enfermedades. 11 de Abril del 2018, de Psicomed Sitio web: http://www.psicomed.net/cie_10/cie10_F34.html.
3. Altshuler LL, Hendrich V, Cohen LS. Course of mood and anxiety disorders during pregnancy and the postpartum period. *Journal of Clinical Psychiatry*, 1998; 59: 29. 11 de Abril del 2018.
4. Rohan KJ, Lindsey KT, Roecklein KA, Lacy TJ. Cognitive-behavioral therapy, light therapy and their combination in treating seasonal affective disorder. *Journal of Affective Disorders*, 2004; 80: 273-283. 11 de Abril del 2018.
5. Bernardo R.. (2017). F32.9 Trastorno depresivo no especificado. 11 de Abril del 2018, de PSIKIA UNED Sitio web: <https://core.psykia.com/node/693>.
6. Firebase. (2018). Documentación. 11 de Abril del 2018, de Google Firebase Sitio web: <https://firebase.google.com/docs/guides/?hl=es-419>.
7. Facebook. (2018). Plataforma de Messenger. 11 de Abril del 2018, de Facebook Sitio web: <https://developers.facebook.com/docs/messenger-platform>.
8. Android Developers. (2018). Introducción a Android. 11 de Abril del 2018, de Google Sitio web: <https://developer.android.com/guide/index.html?hl=es-419>.

9. Spamlaws. (2018). Different Types of Internet Bots and How They Are Used. 11 de Abril del 2018, de Spamlaws Sitio web: <https://www.spamlaws.com/how-internet-bots-are-used.html>.
10. Techopedia. (2018). Internet Relay Chat Bot (IRC Bot). 11 de Abril del 2018, de Techopedia Sitio web: <https://www.techopedia.com/definition/24493/internet-relay-chat-bot-irc-bot>.
11. Facebook. (2018). Messenger. 11 de Abril del 2018, de Facebook Sitio web: <https://messenger.fb.com/>.
12. Facebook NewsRoom. (2018). Noticias. 11 de Abril del 2018, de Facebook Sitio web: <https://ltam.newsroom.fb.com/>.
13. http://www.abc.es/salud/enfermedades/abci-tratamiento-mas-selectivo-para-depresion-201805071740_noticia.html

CITAS

-
1. ⁱ R.I ABC Salud. (2018). Un tratamiento más selectivo para la depresión. 7 de Mayo del 2018, de DIARIO ABC, S.L Sitio web: http://www.abc.es/salud/enfermedades/abci-tratamiento-mas-selectivo-para-depresion-201805071740_noticia.html
 2. ⁱⁱ American Psychiatry Association (1994). DSM-IV. Manual diagnóstico y estadístico de los trastornos mentales. 3a. ed. Washington. 11 de Abril del 2018.
 3. ⁱⁱⁱ Psicomed. (2017). Clasificación Internacional de Enfermedades. 11 de Abril del 2018, de Psicomed Sitio web: http://www.psicomed.net/cie_10/cie10_F34.html.
 4. ^{iv} Psicomed. (2017). DSM IV: Manual diagnóstico y estadístico de los trastornos mentales. 11 de Abril del 2018, de Psicomed Sitio web: <http://www.psicomed.net/dsmiv/dsmiv6.html#f34.1>.
 5. ^v Altshuler LL, Hendrich V, Cohen LS. Course of mood and anxiety disorders during pregnancy and the postpartum period. Journal of Clinical Psychiatry, 1998; 59: 29. 11 de Abril del 2018.
 6. ^{vi} Rohan KJ, Lindsey KT, Roecklein KA, Lacy TJ. Cognitive-behavioral therapy, light therapy and their combination in treating seasonal affective

disorder. *Journal of Affective Disorders*, 2004; 80: 273-283. 11 de Abril del 2018.

7. ^{vii} Bernardo R.. (2017). F32.9 Trastorno depresivo no especificado. 11 de Abril del 2018, de PSIKIA UNED Sitio web: <https://core.psykia.com/node/693>.
8. ^{viii} Dr. José Antonio García Higuera. (2017). Tratamientos de la depresión. 11 de Abril del 2018, de Psicoterapeutas.com Sitio web: http://www.psicoterapeutas.com/trastornos/depresion/tratamientos_depresion.html.
9. ^{ix} Muy Interesante. (2017). El ejercicio físico combate la depresión. 11 de Abril del 2018, de Muy Interesante Sitio web: <https://www.muyinteresante.es/salud/articulo/el-ejercicio-fisico-combate-la-depresion-751484040814>.
10. ^x Mario Luis Fuentes. (2017). México social: depresión, agenda de riesgo. 11 de Abril del 2018, de Excelsior Sitio web: <http://www.excelsior.com.mx/nacional/2017/04/04/1155769#imagen-1>.

FIGURAS

1. Andrea Luna. (2017). Neurociencias de la Conducta. 11 de Abril del 2018, de Blogspot Sitio web: <http://neuro-y-psicologia.blogspot.mx/2017/>.
2. Mario Luis Fuentes. (2017). México social: depresión, agenda de riesgo. 11 de Abril del 2018, de Excelsior Sitio web: <http://www.excelsior.com.mx/nacional/2017/04/04/1155769#imagen-1>.
3. StatCounter Global Stats. (2018). Mobile Operating System Market Share Worldwide. 7 de Mayo del 2018, de StatCounter Sitio web: <http://gs.statcounter.com/os-market-share/mobile/worldwide#monthly-201704-201804-bar>.
4. Móvil Celular. (2017). Huawei GW Especificaciones. 11 de Abril del 2018, de Móvil Celular Sitio web: <http://www.movilcelular.es/huawei-gw/caracteristicas/3466>.